Cost-efficient web service composition for processes with distributed retrieval queries: Position paper

Irma Sofia Espinosa Peraldi and Ralf Möller
Hamburg University of Technology
Software Systems Institute (STS)
Hamburg, Germany
sofia.espinosa@tuhh.de, r.f.moeller@tuhh.de

1 Problem description

In the near future, description logic reasoning services will be offered as web services and initial versions of, for instance, OWL QL web services have already been developed (e.g., [1]). Due to the high acceptance of web service technologies to support processes across heterogeneous architectures, the amount of web service offering similar functionality is proliferating. Thus, the problem of selecting a particular service binding in order to minimize resource consumption in a web service composition has emerged. In particular we investigate optimization strategies for web service compositions that define *large-scale inference processes* with *query correlation*. Due to the business logic behind the process, there is an implicit knowledge about the enquired domain, the processing order of the queries and if a query subsumes another. Furthermore, the web service partners of such a composition provide reasoning services over (possibly heterogeneous) ontologies. Thus, the optimization criteria we are interested in is very much service-dependent, where reasoning tasks with query subsumption, optimization techniques with Abox indexing and KB availability are being considered.

2 Approach

The optimization problem is expressed in terms of a *configuration problem* in which the web service partners (DL reasoners) become objects to be combined such that they satisfy the given criteria. The optimization algorithm was designed such that it can handle multiple criteria expressed as constraints, which

are hard criteria that can not be relaxed, and objectives with no total order, of which a minimum or maximum number should not be violated. Furthermore, the considered criteria have dynamic values, due to the exposure of the web services partners to other (competing) calls for instance retrieval. Thus, it is not possible to a priori determine an execution plan. Therefore, we pursued an approach proposed by [3] using the notion of Pareto dominance, for local optimization, where for each invocation step in the composition, the set of given web service partners are being compared against the multiple objectives along the dimensions. A relaxation process is applied in case of over constraining. For details on the implementation see [4].

3 Contribution

Until now no solution has been presented to optimize large-scale inference processes (represented as web service compositions), where on the one side, the correlation of the process's queries is being considered, and on the other side, reasoning tasks and optimization techniques offered by several DL reasoning services (representing the web service partners of the composition), is being used as criteria for optimizing resource consumption. In the current implementation only RacerPro[2] reasoners can be used to obtain required information about loaded Tboxes and Aboxes, about Abox index structures being computed, about whether a new query is subsumed by another query already answered by some server etc. However, since web service architecture are more and more accepted, current proposals for DL reasoner communication languages such as DIG might be extended with corresponding facilities in the future.

References

[1] J. Galinski, A. Kaya, and R. Möller. Development of a server to support the formal semantic web query language OWL-QL. In I. Horrocks, U. Sattler, and F. Wolter, editors, *Proc. International Workshop on Description Logics*, 2005.

[2] Racer Systems GmbH & Co. KG. Racerpro 1.9. WWW page, March 2005. http://www.racer-systems.com (10.03.06).

[3] D. Navinchandra. *Exploration and innovation in Design. Towards a Computational Model.* Springer-Verlag, 1 edition, 1991.

[4] I. S. Espinosa Peraldi. Cost-efficient web service compositions. Master thesis, TU Hamburg-Harburg, June 2005.