Semantically supported SLA negotiation

András Micsik[#1], Henar Muñoz Frutos [*2], Ioannis Kotsiopoulos[$3], Bastian Koller [&4]

[#] *Department of Distributed Systems, MTA SZTAKI*
Lágymányosi utca 11. Budapest, 1111 Hungary
[1] micsik@sztaki.hu

[*] *Telefónica Investigación y Desarrollo S.A*
C/ Emilio Vargas, 6, 28043 Madrid, Spain
[2] henar@tid.es

[$] *School of Computer Science, University of Manchester*
Kilburn Building, Oxford Road, Manchester, M13 9PL, United Kingdom
[3] ioannis@cs.man.ac.uk

[&] *High Performance Computing Center Stuttgart, University of Stuttgart*
Nobelstrasse 19 70550 Stuttgart, Germany
[4] koller@hlrs.de

***Abstract*—The evolution of services market raises the need for automatic support for negotiating service use criteria. In order to reach an agreement, the negotiating parties need to develop a common understanding of the Quality of Service (QoS) terms which are part of the Service Level Agreement (SLA). The use of semantic annotations together with reasoning can increase the level of flexibility and automation in SLA management. A framework is presented for SLA negotiation allowing the two parties to use their own terminologies.**

I. INTRODUCTION

An important factor of service-based business nowadays is the use of Service Level Agreement (SLA). A SLA is a formal agreement negotiated between two parties, designed to create a common understanding about services, priorities, responsibilities, etc. Based on current evolution of web service provision such as Cloud Computing we assume that SLAs will be used more frequently and will cover parts of electronic contracts defining the QoS terms, as well as the obligations on involved business parties.

Existing SLA specifications such as Web Service Agreement Specification (WS-Agreement) [1] or Web Service Level Agreements (WSLA) [2] are seen as the basis for the establishment of a common understanding between the negotiating parties, and mainly address interoperability at a syntactic level. However, the need for semantic interoperability between service providers and business users raises the need to enhance these SLA specifications, since the terminology used inside the SLA files can be very different in terms of metrics (CPU, Compute Unit), and measurement units (GHz, ECU). Ontologies and semantic technologies can address this obstacle to a large extent as we discuss this in the rest of the paper. Ontologies can capture the meaning of the SLA related terminology in an unambiguous and machine understandable way, while semantic annotations within the SLA document can link any term to its well defined equivalent in the common ontology. The objective of our approach is to adjust the negotiation protocol to take into account the semantic annotations and to improve the efficiency and the flexibility of the negotiation process.

II. SEMANTICALLY ANNOTATED SLA

Customers and service providers need a common language for SLA negotiation, in order to understand each other's offers and bids. Current SLA specifications only define the format of expressing an SLA offer, but the content can use different languages, terms and metrics. Here we propose the use of semantic annotations and a new format for semantically annotating SLA descriptions called Semantic Annotations for Service Level Agreement (SA-SLA) [5]. The annotation mechanism is independent of the ontology expression language and this specification requires and enforces no particular ontology language. SA-SLA is based on the Semantic Annotations for Web Service Description Language (SA-WSDL) [3], which has become the dominant approach in the area of Semantic Web Services. SA-SLA provides a standard description format extending current WS-Agreement [1] and WSLA [2] specifications with semantic annotations in order to provide the domain vocabulary they lack. In this matter, the elements which can be annotated by SA-SLA are: Service Description Terms related to functional properties, SLA parameters and metrics, and finally service properties involving measurable non-functional properties.

Thus, it becomes possible to link elements of WS-Agreement to ontology concepts. In order to test the approach in practice, we developed a common SLA ontology [4] providing a model for SLA offers and bids together with SLA parameters potentially used in SLAs. The ontology enables the representation of SLA template files as instances in the ontology, considering also the parameters and metrics contained and their values. It also collects QoS metrics and measurement units. In our ontology, QoSParameter represents a non-functional property of the requested service within a specific domain. Each QoSParameter is associated with a Metric characterized by ValueType (float, integer, boolean, etc.), a Value and a MeasurementUnit (e.g. euro, kB, ms).

The local part of the knowledge model, which is defined by the customer or provider can extend this common ontology in various ways. First, it adds the definition of locally used QoS parameters, metrics or measurement units. It can also add descriptions about local environment, such as available resource types and their parameters, licenses or platform dependencies. Furthermore, the mapping of received SLAs into the local environment is also implemented here, for example by using rules.

III. Practical Application of SA-SLA

The experimental implementation of our approach applied for SLA negotiation can be seen in Figure 1. As common parts we can consider the SA-SLA specification defining how to annotate SLA files and the common conceptual model expressed as an ontology.

The parties involved use SLA Negotiators to exchange SA-SLA files according to agreed protocols, such as the WS-Agreement protocol. The role of the SLA Mediator component is to import received SLAs into the locally supported SLA model. To achieve this, the mediator component combines the common model with local knowledge.

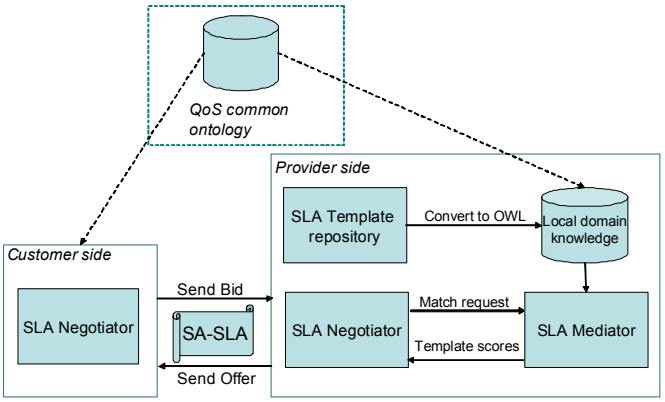

Fig. 1. Example setup for semantically annotated SLA negotiation

The bid sent by the customer is received by the SLA Negotiator component of the provider, which forwards it to the SLA Mediator. The SLA Mediator first parses the bid and creates an OWL instance representing the bid. The locally used SLA templates are already represented in the local domain knowledge as instances. In the next step, the QoS parameters of the bid are mapped to local QoS parameters using local metrics and terminology. Then, the bid and the local templates are directly compared and a matching score is calculated for each template. The SLA Negotiator receives the template scores and proceeds with negotiation, which in the current phase of implementation means to send the template with the best score as an offer to the customer.

The step of translating incoming SLAs to the local knowledge domain is explained in Figure 2. The XML excerpt shows an SLA parameter definition according to WSLA, which is named CacheSizeMin, and contains also an SA-SLA annotation: a modelReference to the MinCacheSize class of the common QoS ontology. This QoS parameter definition is later used to define a concrete guarantee term, which requires the minimum cache size to be greater than 4 gigabytes (XML is not shown for this part). The guarantee term is converted to an instance of the local ontology, and its original attributes are mapped to OWL properties, keeping the connections to the common ontology wherever possible. Then, rules are used to fill in the values of local unit and local value properties. For example, here megabytes are used locally instead of gigabytes. As a result, the QoS parameter in the example becomes comparable to other QoS parameters stored locally, as the measurement units are homogeneous now and the metrics are all mapped to the common ontology.

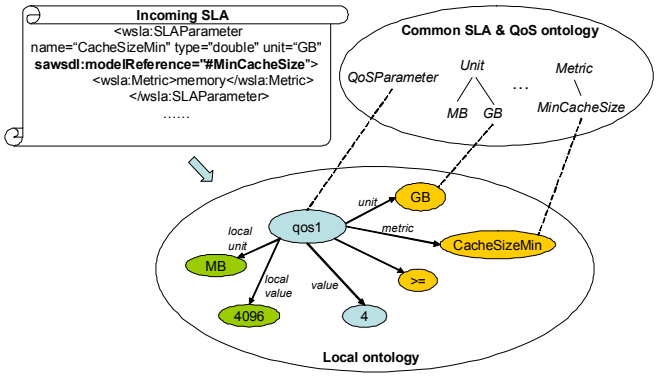

Fig. 2. Using SA-SLA for SLA parameter translation

The presented approach allows to keep the original, standard SLA formats, and to explain its elements in a semantic way, by using semantic annotations in SLA template files. The approach is backward compatible, so that components non aware of semantic annotations can continue working with SLA files. The common core ontology helps to avoid the many-to-many mapping problem, so that only a single mapping to the common ontology is required in local environments. Our experimentation with SA-SLA proved the feasibility of the approach using available semantic tools such as Jena and Pellet.

Acknowledgment

This work has been supported by the BREIN project (http://www.gridsforbusiness.eu) and has been partly funded by the European Commission under contract FP6-034556.

References

[1] GRAAP-WG, "Web Services Agreement Specification (WS-Agreement)", September 2005.

[2] H. Ludwig, A.Keller, A. Dan, R.P. King and R. Franck, "Web service level agreement (WSLA) language specification", 2003, IBM.

[3] J. Kopecky, T. Vitvar, C. Bournez and J. Farrell, "SAWSDL: Semantic Annotations for WSDL and XML Schema", IEEE Internet Computing, 11(6), 2007, pp. 60-67.

[4] "The BREIN ontologies", available at http://www.eu-brein.com/index.php?option=com_mambowiki&Itemid=115

[5] H. Muñoz, I. Kotsiopoulos, L. M. Vaquero, L. Rodero, "Enhancing Service Selection by Semantic QoS", in 6th Annual European Semantic Web Conference (ESWC), 2009, Heraklion, Greece