Grounding OWL-S in SAWSDL

Massimo Paolucci\textsuperscript{1}, Matthias Wagner\textsuperscript{1}, and David Martin\textsuperscript{2}

\textsuperscript{1} DoCoMo Communications Laboratories Europe GmbH
\{paolucci, wagner\}@docomolab-euro.com
\textsuperscript{2} Artificial Intelligence Center, SRI International
martin@ai.sri.com

Abstract. SAWSDL and OWL-S are Semantic Web services languages that both aim at enriching WSDL with semantic annotation. In this paper, we analyze the similarities and differences between the two languages, with the objective of showing how OWL-S annotations could take advantage of SAWSDL annotations. In the process, we discover and analyze representational trade-offs between the two languages.

1 Introduction

Semantic Web services have emerged in the last few years as an attempt to enrich Web services languages with ontological annotations from the Semantic Web. Overall, the goal of such efforts is to facilitate Web services interaction by lowering interoperability barriers and by enabling greater automation of service-related tasks such as discovery and composition. A number of proposals, such as OWL-S 0, WSMO 0 and WSDL-S 0, have been on the table for some time. They provide different perspectives on what Semantic Web services ought to be, and explore different trade-offs. Each of these efforts is concerned with supporting richer descriptions of Web services, but at the same time each has made an effort to tie in with WSDL, and through it to Web service technology. In the case of OWL-S, an ontology-based WSDL Grounding is provided, which relates elements of an OWL-S service description with elements of a WSDL service description.

Recently, Semantic Web services reached the standardization level with SAWSDL 0, which is closely derived from WSDL-S. A number of important design decisions were made with SAWSDL to increase its applicability. First, rather than defining a language that spans across the different levels of the WS stack, the authors of SAWSDL have limited their scope to augmenting WSDL, which considerably simplifies the task of providing a semantic representation of services (but also limits expressiveness). Second, there is a deliberate lack of commitment to the use of OWL 0 or to any other particular semantic representation technology. Instead, SAWSDL provides a very general annotation mechanism that can be used to refer to any form of semantic markup. The annotation referents could be expressed in OWL, in UML, or in any other suitable language. Third, an attempt has been made to maximize the use of available XML technology from XML schema, to XML scripts, to XPath, in an attempt to lower the entrance barrier to early adopters.
Despite these design decisions that seem to suggest a sharp distinction from OWL-S, SAWSDL shares features with OWL-S’ WSDL grounding: in particular, both approaches provide semantic annotation attributes for WSDL, which are meant to be used in similar ways. It is therefore natural to expect that SAWSDL may facilitate the specification of the Grounding of OWL-S Web services, but the specific form of such Grounding is still unknown, and more generally a deeper analysis of the relation between SAWSDL and OWL-S is missing. To address these issues, in this paper we define a SAWSDL Grounding for OWL-S. In this process we try to identify how different aspects of OWL-S map into SAWSDL. But we also highlight the differences between the two proposals, and we show that a mapping between the two languages needs to rely on fairly strong assumptions. Our analysis also shows that despite the apparent simplicity of the approach, SAWSDL requires a solution to the two main problems of the semantic representation of Web services: namely the generation and exploitation of ontologies, and the mapping between the ontology and the XML data that is transmitted through the wire.

The result of this paper is of importance for pushing forward the field of Semantic Web services by contributing to the harmonization of two proposals for the annotation of Web services. In the paper, we will assume some familiarity with OWL-S and SAWSDL, neither of which is presented. The rest of the paper is organized as follows. In section 2 we will analyze the similarities and differences between OWL-S and SAWSDL. In section 3, we will introduce an OWL-S grounding based on SAWSDL, with analysis of its strengths and weaknesses. In section 4 we will discuss the finding and conclude.

2 Relating SAWSDL to OWL-S

The first step toward the definition of a SAWSDL Grounding for OWL-S is the precise specification of the overlap between the two languages. Since the two languages have a very similar goal: provide semantic annotation to WSDL, they have some similarities. The first one is that both OWL-S and SAWSDL express the semantics of inputs and outputs of WSDL operations. SAWSDL does it via a direct annotation of the types and elements while the OWL-S Grounding maps the content of inputs and outputs to their semantic representation in the Process Model. The second similarity is that both languages support the use of transformations, typically based on XSLT, to map WSDL messages to OWL concepts. These transformations allow a level of independence between the message formats and the semantic interpretation of the messages, allowing developers to think of the implementation of their application independently of the semantic annotation that is produced. The third similarity is that both OWL-S and SAWSDL acknowledge the importance of expressing the category of a service within a given taxonomy. SAWSDL provides category information by annotating interface definitions. OWL-S provides this information in the Profile through its type specification or through the property serviceCategory.

Despite their similarities, the two languages have also strong differences. The first one is in the use of WSDL. OWL-S uses WSDL exclusively at invocation time; therefore the WSDL description relates directly to atomic processes in the Process