Applying Semantics (WSDL, WSDL-S, OWL) in Service Oriented Architectures (SOA)

Michael Herrmann[1], Muhammad Ahtisham Aslam[2], Oliver Dalferth[1]

[1]DaimlerChrysler AG, Sindelfingen, Germany
{michael.hm.herrmann,oliver.dalferth}@daimlerchrysler.com

[2]Betriebliche Informationssysteme, Universität Leipzig, Germany
alsam@informatik.uni-leipzig.de

Abstract.

WSDL-S offers a solution for combining services (described in WSDL) with ontologies (e.g. described in OWL). By modeling inputs, outputs, preconditions and effects (IOPE) based on ontologies WSDL-S enables the enriching of services with semantics. We are going to describe in this paper our lab based study.

1. Introduction

In a Service Oriented Architecture (SOA), interaction between service providers and service consumers takes place in a dynamic model where the service provider can also act as a service consumer. This dynamic is a key difference between Service Oriented Architecture and Client-Server Architecture [1]. Web service standards like WSDL [2], SOAP [3], UDDI [4] and BPEL [5] provide syntax based interaction and composition of Web services in a loosely coupled way. But a dynamic Web services composition needs more than syntactical information – it needs semantic descriptions of services. Approaches like OWL-S [6], WSMO [7] and WSDL-S [8] are being discussed in semantic Web service community. WSDL-S extends WSDL by defining new elements and annotations for already existing elements. The W3C standard OWL [9] is based on RDF(S) and supports the modeling of knowledge/ontologies [16]. Some work has already been done for mapping BPEL to a process model [10]. In the late 1980's 2-tier architectures in the sense of client/server platforms were created in order to improve flexibility [11]. Since the mid-1990's, 3-tier architectures have remained the most common software architectures for enterprise level computing. However, as the technology within the computing industry evolved, the original middleware champions CORBA [12] and DCOM [13] have been replaced with Web services. As the Web becomes more semantic [14] and applications become more agile [15] the need for additional architectural layers becomes more prevalent. Creating a 4-tier architecture by replacing the Business Layer by two different layers (CO Layer and Services Layer) could loosen the tight coupling of services in a 3-tier architecture.

2. WSDL-S — connecting WSDL and OWL

WSDL solves the need of interoperability in a technical manner, but lacks in adding semantic information. OWL supports developing of ontologies in a powerful way, but lacks in describing the technical details of services. WSDL-S extends WDSL in order to use semantic capabilities of OWL to provide semantically enriched meanings of WSDL services – WSDL-S connects WSDL and OWL in a practical way. Approaches like OWL-S and WSMO are well discussed in the Web Service Community, but we realized the following advantages of WSDL-S in earlier stages of our study:

+ Compatible to WSDL
+ Free choice in the use of modeling technology (OWL, UML)
+ Simplicity in appliance
+ Based on a stabile standard, which is important for practical use

We describe the elements and the extending attributes of WSDL-S in this paragraph. The extending attribute **<wssem:modelReference>** supports the connections between a WSDL document and anontology (e.g. modeled in OWL). You can actually use this extending attribute in the elements `<xsd:simpleType>`, `<xsd:complexType>`, `<wsdl:operation>`, `<wssem:precondition>` and `<wssem:effect>`. The second extending attribute **<wssem:schemaMapping>** solves differences in schemas by XSL transformation. Both of the new elements **<wssem:precondition>** and **<wssem:effect>** are sub elements of `<wsdl:operation>` and are actually used for service finding. The third new element **<wssem:category>** gets place in the classification of the service interface. This element is needed for a semantic lookup in a service registry. The following figure illustrates the usage of WSDL, OWL and WSDL-S in combination.

3. The Lab based study

We are realizing a lab based applying of semantics for a well selected business process called "planning and execution". This process is executed each day in the morning in order to plan the assembly of the engines. The orders of assembling itself, assembling jobs and best batch sizes are assigned. Therefore we need information about the availability of components, the need of engines and assembly restrictions like the number of employees, capacity and availability of the equipment. The process flow is coordinating production, assembly, organizer of assembly and scheduler/controller. The flow detects the release order, determinates the need of engines, passes the plausibility check, calculates aims, passes a restriction check, creates a production order, proves the production order, and fixes the amount and the order of production.

The goal is to model IOPEs (inputs, outputs, preconditions and effects) of each process step at the process "planning and execution", semantically. Additionally we are defining an automotive ontology based on the general concept of automobiles. We are adding specific constraints, instances of cars like an SLK (which is a Roadster) and the IT-systems needed to support the well selected process described in the last preceding paragraph. We are considering the maturity of OWL in regard to modeling the requirements of specific departments within MCG. We expect to gain significant experiences in modeling real-life requirements, in proving the maturity of OWL and in defining the restrictions of reusable enterprise ontologies.

Assumption: The process "planning and execution" exists of 11 IT supported functions. Each of those 11 functions is represented as a service. One service called "check plausibility" is selected in order to enrich it with semantic information. This service exists of one or more service operations. The goal is to improve reuse in development time (not in runtime). The inputs of the service operation "doCheck" are "SNR", "amount" and "day of delivery". The outputs are the same as the inputs plus the Boolean attribute "plausible".

```
...

xmlns:wssem="http://www.ibm.com/xmlns/WebServices/WSSemantics"
xmlns:dcx="http://www.dcx.com/xmlns/XMLSchema/DCX"
xmlns:AutomotiveDCX="http://www.dcx.com/ontologies/dcx.owl#">

...

   <complexType name="CheckPlausiResponse">
    <sequence>
     <element name="SNR" nillable="false" type="dcx:SNR" />
    </sequence>
    <sequence><element name="amount" nillable="false" type="xsd:integer"
        wssem:modelReference="AutomotiveDCX#Amount"/>
    </sequence>
    <sequence><element name="date_of_delivery" nillable="false" type="xsd:date"
        wssem:modelReference="AutomotiveDCX#Date_of_Delivery"/>
    </sequence>
    <sequence><element name="plausible" nillable="false" type="xsd:boolean"
        wssem:modelReference="AutomotiveDCX#Plausible"/>
    </sequence>
   </complexType>

...
```

The code extract above illustrates the combination of WSDL and WSDL-S. It illustrates the return type (`<complexType>`) of the service operation “doCheck” as well. Rows `01–03` define `Namespaces` for the standard WSDL-S (`01`), the DaimlerChrysler schema (`02`) and the DaimlerChrysler specific ontology (`03`). Row `06`, `08`, `11` and `14` are defining the required return data types. The functional data type “SNR” in row `06` is extended with not null and defined as technical data type `dcx:SNR`. The functional data type “amount” in row `08` is extended as not null as well and defined as primitive data type `xsd:integer`. Additionally references row `09` to the concept “amount” stored in our ontology in order to define the concept unique. Similar to “amount” in row `08` and `09` is “date of delivery” in row `11` and `12` as well as “plausible” in row `14` and `15`.

4. Conclusion and Outlook

The proposal of a 4-tier architecture by splitting the Business Layer into two different layers (CO Layer and Services Layer) aims at loosening the tight coupling between services in a 3-tier architecture. Also the enabling of a more independent interaction (orchestration) between the services is an important goal of the split. Both layers are hosted within an ESB which is positioned between the Presentation and the Persistence Layer. The ESB can be found in a 4-tier architecture at the same position where the Middleware is located in a 3-tier architecture.

The interoperable usage of services can be implemented by using WSDL but there still exists a lack of a semantic understanding of services. OWL on the other hand supports ontologies but cannot satisfy all needs for describing the services technically.

WSDL-S offers a solution for combining WSDL documents with ontologies and proved to be simple in appliance in first tests for the process “planning and execution” at MCG. It is intended to model the IOPEs for all the 11 process steps of this process semantically and to define an automotive ontology. But modeling IOPE’s based on ontologies is not enough to realize a semantic SOA. There exists a lack of common understandings of services between the participants of a SOA but adding semantics within the SOA aims at providing a shared meaning of business services within an organization and probably across the organizational boundary.

5. References

1. F. Buschmann, R. Meunier, H. Rohnert, P. Somerlad and M. Stal, *Pattern-Oriented Software Architecture – A System of Patterns*. Wiley (October 1996) 99 – 122.
2. Web Services Description Language (WSDL) [online] Available http://www.w3.org/TR/wsdl.
3. SOAP Version 1.2 Part 1: Messaging Framework [online] Available http://www.w3.org/TR/soap12-part1/.
4. UDDI Version 3.0 [online] http://uddi.org/pubs/uddi-v3.00-published-20020719.htm
5. Business Process Execution Language for Web Services Version 1.1. [online] available ftp://www6.software.ibm.com/ software/developer/library/ws-bpel.pdf
6. OWL-S: Semantic Markup for Web Services. [online] Available http://www.daml.org/services/owl-s/1.2 /overview/
7. Web Services Modeling Ontology [online] Available http://www.wsmo.org/
8. R. Akkiraju, J. Farell, J.A. Miller, M. Nagarajan, A. Sheth and K. Verma : "Web Service Semantics – WSDL-S" [online] Available http://www.w3.org/2005/ 04/FSWS/Submissions/17/WSDL-S.htm.
9. OWL (Ontology Web Langugae); W3C Recommendation http://www.w3.org/TR/2004/REC-owl-guide-20040210/
10. M. A. Aslam, S. Auer, J. Shen and M. Herrmann: Expressing Business Process Models as OWL-S Ontologies. Proceedings of BPM 2006 Workshops, LNCS 4103, pp. 395-410.
11. http://www.sei.cmu.edu/str/descriptions/clientserver.htm
12. Object Management Group home page [online]. http://www.omg.org (1996).
13. Microsoft Corporation. Distributed Component Object Model Protocol-DCOM/1.0, draft, November 1996 [online]. http://www.microsoft.com/Com/ resources /comdocs.asp (1996).
14. Tim Berners-Lee *The Sematnic Web;* http://www. consortiuminfo.org/bulletins/semanticweb.php
15. Martin Fowler *The New Methodology*, http://www. martinfowler.com/articles/newMethodology.html
16. M. Herrmann, R. Golden, JBoss World 2006, *Why SOA lacks reuse* http://jbossworld.com/jbwv_2006/soa_for_ the_-real_world/ HERRMANN_CO-Layer_final.pdf