HAL
open science

Yet another semantic annotation for WSDL

Yassin Chabeb, Samir Tata

To cite this version:

Yassin Chabeb, Samir Tata. Yet another semantic annotation for WSDL. WWW/Internet 2008 : IADIS International Conference, Oct 2008, Freiburg, Germany. pp.437 - 441. hal-01380984

HAL Id: hal-01380984

https://hal.science/hal-01380984v1

Submitted on 13 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

YET ANOTHER SEMANTIC ANNOTATION FOR WSDL

Yassin Chabeb and Samir Tata
Institut TELECOM, UMR CNRS SAMOVAR
9 rue Charles Fourier 91011 Evry France
{Yassin.Chabeb,Samir.Tata}@it-sudparis.eu

ABSTRACT

Web services are loosely-coupled and self descriptive applications. They are based on standards such as SOAP for message transport, WSDL for service description, and UDDI for service advertisement and discovery. Nevertheless, the lack of semantics in WSDL prevents automatic discovery and hence automatic invocation and composition. In our work, we are interested in extending existing approaches for the description of Semantic Web Services. Our proposed approach, Yet Another Semantic Annotation for WSDL (YASA4WSDL), is an extension of the W3C recommendation on semantics for Web services (SAWSDL) and use two types of ontologies: a Technical Ontology type containing concepts defining semantics of services, their QoS... and a Domain Ontology type containing the concepts defining the semantics of the business domain. We present how our approach is more expressive than the W3C recommendation and submissions to W3C on semantics of Web Services. We show via our implementation how to automatically generate from YASA4WSDL, descriptions like SAWSDL, OWL-S, or WSMO.

1. INTRODUCTION

Web services are running through the Web and based on standards such as SOAP for message transport, WSDL for service description, and UDDI for service advertisement and discovery. Nevertheless, the lack of semantics in WSDL (W3C, 2001; Chinnici et al., 2007) prevents automatic discovery and hence automatic invocation and composition. To deal with these issues several approaches were developed that use semantic models (ontologies,...) for description of semantic Web services. We can cite among others, OWL-S (W3C, 2004), SAWSDL (Farrell & Lausen, 2007) and WSMO (ESSI WSMO working group, 2004).

WSMO propose high-level objectives and approaches similar to those of OWL-S but WSMO focuses on goals mediation and choreography. OWL-S focuses on process model. In SAWSDL there is no explicit mention of precondition and effects that one can find in WSMO and OWL-S. In addition SAWSDL is not dedicated to describe Web service behaviour which is essential for service invocation and composition. Nevertheless, SAWSDL is an approach independent of the used semantic representation language thanks to the separation of semantic annotation mechanism from the representation of the semantic descriptions. This gives flexibility to developer community to select their favourite semantic representation language, to reuse semantic domain models and annotate descriptions using multiple ontologies.

In our work, we are interested in description of semantic Web services taking into account existing approaches. Our proposed description of semantic Web services is based on the de-facto standard to describe Web services, namely WSDL. It requires no other changes to existing WSDL or XML Schema (Fallside & Walmsley, 2004) documents, or the way in which they had been used previously. Specially, the developed tools for parsing WSDL documents or for service invoking. To meet this target, we propose YASA4WSDL an extension of SAWSDL that use two types of ontologies. The first one, called Technical Ontology Type, concerns ontologies that describe service concepts (*e.g.* interface, input, output...) and ontologies that describe non functional concepts of services (*e.g*. QoS, context awareness attribute types...). The second ontology type, called Domain Ontology Type, concerns ontologies that define the semantics of the service business domain (e.g. tourism, health, trade...).

This paper is organized as follows. Section 2 presents the related works on the description of semantic Web services. Section 3 presents requirements and motivations for semantic description. In Section 4, we give an overview of our proposed approach. Section 5 presents our contribution and ongoing implementations to manipulate YASA4WSDL and to generate from YASA4WSDL semantic description in SAWSDL, OWL-S and WSMO. Finally, Section 6 concludes this paper and presents our future work.

2. RELATED WORK

We present in this section the most related work to our contribution. In (Chinnici et al., 2007), SAWSDL suggests how to add semantic annotations to various parts of a WSDL document like interface and operation. This extension is inline with WSDL extensibility framework. SAWSDL defines a new namespace called "sawsdl" and adds an extension attribute called modelReference so that relationships between WSDL components and concepts in another semantic model (e.g. ontology) are handled. The modelReference attribute may contain a list of references, but actually, one can not know the nature or concept of each reference, so we can not link the semantic information to a particular item in the vocabulary of service (profile, process, ...) or to a feature of the service (effect of operation, precondition…).

OWL-S approach proposes an ontology of services motivated by the need to provide three essential types of knowledge about a Web service: what does the service provide for prospective clients, how it is used and how does one interact with it? The answer to the first question is given in the "profile" which is used to advertise the service. The service profile elements include preconditions, inputs, outputs, results and service category. The answer to the second question is given in the "process model" which includes inputs, outputs, preconditions, effects and the behaviour of the service (data and control flow). The answer to third question is given in the "grounding". A grounding provides the needed details about transport protocols.

In (ESSI WSMO working group, 2004), WSMO is a conceptual model for four top level elements as the main concepts which have to be described in order to describe Semantic Web services: ontologies, services, mediators and goals. Descriptions of a WSMO service comprise non functional properties, a provided interface and a provided capability. Descriptions comprise of a WSMO goal comprise non functional properties, a requested interface and a requested capability. A WSMO interface describes messages sent to/by a WSMO service and the visible behaviour of that service. A WSMO capability includes: non functional properties, preconditions, assumptions, post-conditions, and effects.

In (Martin et al., 2007b), the authors propose to use OWL-S constructs (service profile and process model) as references of SAWSDL annotations. The idea is not to continue employing the OWL-S grounding and to adopt a SAWSDL-based perspective. This approach starts from the assumption that atomic processes in OWL-S correspond to SAWSDL's operations. Then it proposes to add a modelReference attribute with message elements as it is not defined by SAWSDL, and finally the authors propose to refer a message's modelReference to an OWL class. However, in this paper, there are many issues needing clear decision and more precision, having to do with difficulties in mapping between Message Exchange Pattern in SAWSDL and inputs/outputs in atomic processes on one hand and the fact that OWL-S does not have constructs that provide direct correspondence to "interface" elements in SAWSDL on the other hand.

In (Klein et al., 2005; Küster & Konig-Ries, 2007), DIANE Service Description (DSD) and DIANE Elements (DE) propose object-oriented service description in order to put into practice additional requirements that are not fulfilled by semantic service description such us WSMO and OWL-S. In (Klan, 2006; Küster et al., 2007), the authors describe how DSD is enhancing semantic description by using a ontology to express concepts like precondition and effect.

3. REQUIREMENTS AND MOTIVATIONS

Semantic Web services use domain ontologies to provide their description in terms of semantic concepts. A concept is used to denote service description parts in the service domain ontology. Semantic-based description languages such us OWL-S and WSMO are closed approaches: they only handle respectively OWL and WSML as type of ontologies (Ould Ahmed M'Bareck & Tata, 2008). Moreover, they specify a definite but limited set of concepts which is not easy to extend. WSMO contains some concepts that do not appear in OWL-S ontology and vice versa. In SAWSDL there is no explicit mention of precondition and effects that one can find in WSMO and OWL-S. In addition SAWSDL is not dedicated to describe Web service behaviour which is essential for service invocation and composition.

Semantic annotation in SAWSDL use an extended attribute called modelReference so that relationships between WSDL components and concepts in another semantic model (e.g. ontology) are handled. Therefore, SAWSDL is an approach independent of the semantic representation language thanks to the separation of semantic annotation mechanism from the representation of the semantic descriptions. This gives flexibility to

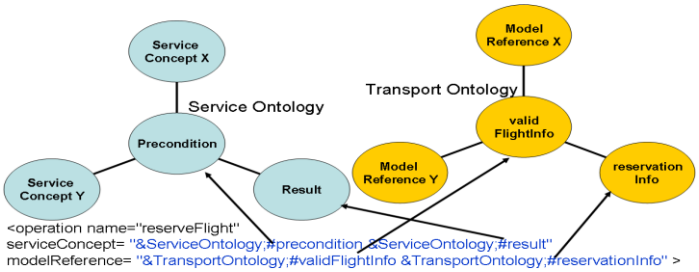
the developers' community to select their favourite semantic representation language, to reuse semantic domain models and annotate descriptions using multiple ontologies.

In order to enhance service discovery, composition and invocation, we have identified requirements for the description of semantic Web services. The SAWSDL specification states that a modelReference may be used with every element within WSDL and XML schema. However, SAWSDL defines its meaning only for interface, operation, fault, element, complexType, simpleType and attribute (Farrell & Lausen, 2007). It should be noted that the guidance given regarding the uses of modelReference for each of these elements has much more the flavour of suggestions than definitions (Martin et al., 2007a). For example, the material on usage with interfaces mentions that modelReference can be used to categorize them according to some model, specify behavioural aspects or other semantic definitions, and similarly for operations. Consequently, when an operation is annotated using several semantic concepts in a ontology, one is not able to differentiate these concepts: which one annotates category, which one annotate the behaviour, which one annotate the QoS? And so on. Similar observation can be made for any semantic annotation of any WSDL element. Therefore, we are convinced that it is necessary to be able to differentiate the semantic description of all services elements. Differentiation of semantic annotation of WSDL elements can be used to enhance discovery. Instead of using modelReference to associate one or more semantic properties to a WSDL element, we should use means to differentiate each semantic property that can be associated to a WSDL element. Indeed, one can consider discovering using one specific semantic property such as Web service effects or can consider composing/invoking services using one specific semantic property such as behaviour.

4. SEMANTIC ANNOTATION FOR WSDL

The main idea is to extend SAWSDL for enhancing expressiveness of service description. In SAWSDL for a given WSDL element one can use many references to concepts in a domain ontology but there is no specification of the semantic information nature: is it a precondition, an effect, a result? etc. That is why we propose, in our description, a new attribute called serviceConcept to give references to the technical concepts corresponding in the same order, to the domain concepts listed in the original SAWSDL "modelReference" attribute. Indeed our approach for semantic Web service description is based on the use of two types of ontologies. The first one, called Technical Ontology, contains several concepts defining semantics of services' concepts and concepts describing their non functional properties (QoS, context…). The second type of ontologies, called Domain Ontology, contains the semantics of the service domain concepts (e.g. tourism).

Figure 1. Service Ontology and Domain Ontology



A service described in YASA4WSDL can define for each WSDL element two attributes providing semantic description. The first attribute, called serviceConcept, contains a set of URI referencing the corresponding concepts in one or several Technical Ontologies. The second attribute contains a set of URI corresponding to the first list and which define the semantics in one or several Domain Ontologies. Let's consider the example presented in Figure 1. There are two ontologies: the first is Technical (Service Ontology) that describes the semantics of some service concepts: precondition and effect. The second ontology is a Domain ontology (called TransportOntology) that describes some concepts in the travel domain: validFlightInfo and reservationInfo. The example presented the semantic annotation of an operation named reserveFlight. The importance of the extended attribute serviceConcept is to distinguish the role played by: validFlightInfo and reservationInfo referenced by modelReference attribute. The serviceConcept references two service concepts: precondition and effect that correspond to the two domain concepts

validFlightInfo and reservationInfo. The order is important here. It associates the first technical concept with the first domain concept, the second technical concept with the second domain concept and so on.

Another advantage is provided by this new approach: in YASA4WSDL, we can extend the Service Ontology by new concepts (new description element, more precise concepts ...) and there is no impact on the annotation system. There will be no limited service ontology. This makes the system of annotation independent of the used semantic business domain representation language and ontology and the used semantic service representation language and ontology. This gives flexibility to the developers' community to select their favourite semantic representation language and their technical ontology, to reuse semantic domain models and annotate descriptions using multiple ontologies.

The Web Service for trip reservation, given below, provides operation for flight reservation based on the specification of a flight request. If the desired flight is available, an itinerary and reservation number will be returned. The similar things are offered by the service concerning hotel and car reservation.

```
1  <description>
2          <interface name="TripReservationInterface"
3              serviceConcept="&ServiceOntology;#interface"      modelReference="&TransportOntology;#reservation">
4                  <operation name="reserveFlight" pattern="http://www.w3.org/ns/wsdl/in-out"
5                      serviceConcept="&ServiceOntology;#precondition &ServiceOntology;#result"
6                      modelReference="&TransportOntology;#validFlightInfo &TransportOntology;#reservationInfo">
7                              <input element="reserveFlightRequest"/>
8                              <output element="reserveFlightResponse"/>
9                  </operation>
10                 <operation name="reserveHotel"...>...     <operation name="reserveCar"...>...
11         </interface>
12         <binding name="reservationSOAPBinding"              interface="TripReservation" type="http://www.w3.org/ns/wsdl/soap"...>
13                 <operation ref="reserveFlight"...>...   <operation ref="reserveHotel"...>...   <operation ref="reserveCar"...>...
14         </binding>
15         <service name="reservationService" interface="TripReservationInterface"...>
16                 <endpoint name="reservationEndpoint" binding="reservationSOAPBinding" address="http://cyl.com/reservation"/>
17         </service>    </description>
```

The semantic annotation of the interface called TripReservationInterface (line 2) is ensured by two attributes: *serviceConcept* and *modelReference* (line 3). They indicate respectively that the interface represents a set of reservation operations in the context of a transport domain ontology. In the operation called *reserveFlight* (line 4), her *serviceConcept* attribute (line 8) provides a list of service concepts to which corresponds respectively, in the same order, a list of model references from the transport domain ontology provided by the *modelReference* attribute (line 6). The operation has as *precondition* associated to the concept of *validFlightInfo* (Flight information have to be valid before reserving flight). The operation has as result the concept of *reservationInfo* (Information about reservation is returned after the effective reservation).

5. YASA4WSDL CONTRIBUTION

We explain here the contribution of YASA4WSDL according the state of the art presented in Section 2. Compared to WSMO and OWL-S, YASA4WSDL offers multiple advantages. First, users can describe, in an upwardly compatible way, both the semantics and the syntax of functional and non functional properties in WSDL, a language that the developers' community is familiar with. Second, by externalizing the semantic of business domain models and the technical concepts (service ant its QoS, context…), one can allow Web service developers to annotate their Web services with their choice of ontology language (such as UML or OWL) unlike in OWL-S or WSMO. In addition, OWL-S and WSMO define their own Service ontologies unlike in YASA4WSDL, which can integrate any technical ontology (Context, QoS or service ontology). In addition (and regarding to the work presented in (Martin et al., 2007b)), YASA4WSDL allows the developer community to specify the correspondence between service concepts (interfaces, operation, fault…) and business concepts. This explicit correspondence can be used to bridge YASA4WSDL and OWL-S, WSMO, and many other description languages for semantic Web services. According to DIANE approach (Klein et al., 2005; Küster & Konig-Ries, 2007), our approach proposes almost the same contributions but it focuses on up to date standards (SAWSDL) and common tools (WSDL-oriented tools).

Currently, we are developing a semantic service bus within the French ANR national project SemEUsE. This bus will provide means to Semantic Web Service based applications for service description, discovery, composition, invocation and monitoring. YASA4WSDL is the proposed description language we are using in the SemEUsE bus. Service discovery, composition, invocation and monitoring are out of the scope of this paper. In this context, we implemented bridges between YASA4WSDL and SAWSDL, WSMO and OWL-S. To manipulate YASA4WSDL descriptions we have used the Eclipse Modelling Framework Project (EMF, http://www.eclipse.org/modeling/emf/). EMF is a modelling framework and code generation facility for

building tools and other applications based on a structured data model. From the XML schema of YASA4WSDL, EMF provided us tools and runtime support to produce a set of Java classes for YASA4WSDL manipulation, along with a set of adapter classes that enable viewing and command-based editing of YASA4WSDL descriptions, and a basic editor (Eclipse plug-in). We have developed bridges from YASA4WSDL to SAWSDL, OWL-S and WSMO. For shake of space we do not present here those bridges.

In addition to EMF tools for YASA4WSDL, we have used also OWL-S Application Programming Interface (OWL-S API, http://www.mindswap.org/2004/owl-s/api/) to create and manipulate OWL-S components as Java objects under Eclipse (Eclipse Ganymede, http://www.eclipse.org/) in order to generate OWL-S descriptions (Service, Process, and Profile) from YASA4WSDL ones.

6. CONCLUSION

We have proposed in this paper Yet Another Semantic Annotation for WSDL (YASA4WSDL). YASA4WSDL is an extension of the W3C recommendation on semantics for Web services (SAWSDL) and use two ontologies: a Technical Ontology and Domain Ontology. The Technical Ontology references semantic concepts of services and their non functional properties whereas the Domain Ontology references concepts of the business domain of the Web service. In addition, we have presented how our approach is more expressive than the W3C recommendation and submissions to W3C on semantics of Web Services. We mentioned our implementation to automatically generate from YASA4WSDL descriptions like OWL-S, SAWSDL and WSMO but not presented here in detail for shake of space. Our ongoing work, within the French ANR national project SemEUsE aims at providing means to Semantic Web Service based applications for service description, discovery, composition, invocation and monitoring. YASA4WSDL is the proposed description language we are using in the SemEUsE service bus. Service discovery, composition, invocation and monitoring in this project will be based on YASA4WSDL descriptions and will consider three practical cases: “bike ride”, “Fire fighting” and “Emergency and crisis situation”.

REFERENCES

Chinnici, Roberto, Weerawarana, Sanjiva, Moreau, Jean-Jacques, & Ryman, Arthur. 2007 (June). Web services description language (WSDL) version 2.0 part 1: Core language. W3C recommendation. W3C. http://www.w3.org/TR/2007/REC-wsdl20-20070626.

ESSI WSMO working group. 2004. Web Service Modeling Ontology. http://www.wsmo.org/.

Fallside, David C., & Walmsley, Priscilla. 2004 (Oct.). XML schema part 0: Primer second edition. W3C recommendation. W3C. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

Farrell, Joel, & Lausen, Holger. 2007 (Aug.). Semantic annotations for WSDL and XML schema. W3C recommendation. W3C. http://www.w3.org/TR/2007/REC-sawsdl-20070828/.

Klan, Friederike. 2006. Context-aware service discovery, selection and usage. Grundlagen von datenbanken.

Klein, Michael, Ries, Birgitta Konig, & Mussig, Michael. 2005. What is needed for semantic service descriptions, a proposal for suitable language constructs. Int. J. of Web and Grid Services.

Küster, U., & Konig-Ries, B. 2007. Semantic Mediation between Business Partners - A SWS-Challenge Solution Using DIANE Service Descriptions. Web Intelligence and Intelligent Agent Technology Workshops.

Küster, Ulrich, König-Ries, Birgitta, Stern, Mirco, & Klein, Michael. 2007. DIANE: an integrated approach to automated service discovery, matchmaking and composition. Proceedings of the 16th Int. Conf. on WWW. New York, USA.

Li, Hui, & Wang, Haiyang. 2006. A Method of Service Description and Discovery in Pervasive Computing Environments. 1st Int. Symposium on Pervasive Computing and Applications, Xinjiang, China.

Martin, David, Paolucci, Massimo, & Wagner, Matthias. 2007a. Bringing Semantic Annotations toWeb Services: OWL-S from the SAWSDL Perspective. ISWC/ASWC.

Martin, David, Paolucci, Massimo, & Wagner, Matthias. 2007b. Toward Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective. In: OWL-S: Experiences and Directions Workshop, European Semantic Web Conf., Innsbruck, Austria.

Ould Ahmed M’Bareck, Nomane, & Tata, Samir. 2008. Services Web : revue des approches de description sémantique. Conférence internationale ”Systèmes d’Information et Intelligence Economique”, Hammamet, Tunisie.

W3C. 2001. Web Services Description Language (WSDL). http://www.w3.org/TR/wsdl/.

W3C. 2004. OWL-S: Semantic Markup for Web Services. http://www.w3.org/Submission/OWL-S/.