Web Service Description Alignment[1]

Analay Baltá, Alberto Fernández

CETINIA, University Rey Juan Carlos, Móstoles, Spain
analay@ia.urjc.es, alberto.fernandez@urjc.es

Abstract. In large-scale open environments mechanisms for locating appropriate services have to deal with the additional problem of semantic mismatches among the components. The objective of this paper is to step forward towards the integration of semantic alignment mechanisms into service discovery frameworks. In particular we focus on the alignment of current service description languages.

1 Introduction

The addition of semantic information to describe Web Services, in order to enable the automatic location, combination and use of distributed components, is nowadays one of the most relevant research Service Oriented Architecture (SOA) topics due to its potential to achieve dynamic, scalable and cost-effective Enterprise Application Integration and eCommerce.

The process of discovering and interacting with a Semantic Web Service includes [3] *candidate service discovery* (match advertised service descriptions against specifications from requesters), *service engagement*, and *service enactment*.

Several description frameworks to annotate provided services on the one hand and express service requests on the other have been proposed. They range from logic-based complex and expressive semantic service descriptions (e.g. OWLS, WSMO) to syntactical ones (WSDL, keywords, tag clouds), with some approaches in between (SAWSDL). In this context, several frameworks to semantically match service advertisements and requests have been presented in the literature [8, 12, 13, 15].

In such open environments the mechanisms for locating appropriate services have to struggle with the additional problem of semantic mismatches among the *service description models or languages* as well as *domain ontologies* used to specify the concepts in the descriptions. Note that most approaches assume the use of the same language for both service advertisements and requests. Therefore, semantic alignment mechanisms need to be purposefully integrated into.

Several works exist on the use of ontology alignment techniques [5, 6] to transform information from one ontology to another. In this paper we tackle the problem of service description model alignment. We study the most widely used approaches for describing services and propose a unified description model for service discovery.

[1] Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-02 and CSD2007-0022 (CONSOLIDER-INGENIO 2010)

Some efforts have been made trying to align or compare different service description approaches. We set out from existing conceptual comparisons between OWL-S and WSMO [18], OWL-S, SAWSDL and WSDL [11, 16] to obtain a general model description of services that facilitates their discovery. Giantsiou et al. [9] propose a service meta-model for the semantic alignment although they only consider two types of services: SAREST [19] and SAWSDL [10].

The rest of the paper is organized as follows. In the next section we analyse the most popular approaches to describing services. In section 3 we first propose an integrated model which contains all the important characteristics of the existing models. Afterwards, we detail how the original models can be mapped into the integrated model. Finally, we present some conclusions and future work.

2 Service description approaches

In this section we describe different approaches to service description. We include semantic models (OWL-S, WSMO), syntactic models (WSDL), hybrid (SAWSDL), as well as other lighter approaches (*keyword*-, *cloud*-, and *text*-based service descriptions). For each approach, we concentrate on the important characteristics from a service matchmaking point of view. Throughout this paper we use an example of an airline service (the Bravo Air example[2]), a fictitious service that provides flight reservations based on the specification on a flight request.

OWL-S. Web Ontology Language for Services (OWL-S) [14] *service profile* is used to describe what the service does and it is crucial in the web service discovery process since it describes the capabilities of web services. We select the profile's relevant fields for service matchmaking so as to define a service by $S = \langle \mathcal{I}, \mathcal{O}, \mathcal{P}, \mathcal{E}, \mathcal{C}, \mathcal{T} \rangle$, where $\mathcal{I}$ represents a set of inputs (property *hasInput*), $\mathcal{O}$ is a set of outputs (property *hasOutput*), $\mathcal{P}$ is a set of preconditions (property *hasPrecondition*), $\mathcal{E}$ a set of effects (property *hasResult*), $\mathcal{C}$ is a set of categories (property *hasCategory*), and $\mathcal{T}$ is plain text description of the Web service (property *textDescription*).

Example: *S*=<{ba[3]:DepartureAirport, ba:ArrivalAirport, ba:OutboundDate, ba:InboundDate, ba:RoundTrip, ba:AcctName, ba:Password, ba:Confirm}, {ba:FlightsFound, ba:PreferredFlightItinerary, ba:ReservationID}, ϕ, {ba:HaveSeatResult}, {unspsc[4]:Travel_agents}, *"This service provides flight reservations based on the specification on a flight request. This typically involves a departure airport, an arrival airport, a departure date, and if a return trip is required, a return date. If the desired flight is available, an itinerary and reservation number will be returned."*>

WSMO. Web Service Modeling Ontology (WSMO) [2] offers four key components to model different aspects of Semantic Web Services: *ontologies*, *goals*, *services*, and *mediators*. Web Service descriptions are defined into WSMO capability

2 http://www.ai.sri.com/daml/services/owl-s/1.0/examples.html
3 ba = "http://www.daml.org/services/owl-s/1.1/BravoAirProcess.owl#"
4 unspsc = "http://www.cs.vu.nl/~mcaklein/unspsc/unspsc84-title.rdfs#"

by their *precondition, postcondition, assumption, effect*, and their *nonFunctionalProperties* (*title, subject, description*). For our service discovery approach, WSMO and OWL-S share the same formalization, although these components are obtained in WSMO from different fields: $\mathcal{I}$ (*precondition*), $\mathcal{O}$ (*postcondition*), $\mathcal{P}$ (*assumption*), $\mathcal{E}$ *(effect*), $\mathcal{C}$ (*subject*), $\mathcal{T}$(*description*).

WSDL. Web Service Description Language (WSDL) [4] is an XML-based language for syntactically describing the service, including the service name, functions, and input and output parameters of abstract definitions. WSDL descriptions are limited in their search by their inability to extend beyond the keyword-based matches. We define a WSDL service $S = <\mathcal{I}, \mathcal{O}, \mathcal{T}>$ where $\mathcal{I}$ is a set of inputs (strings extracted from the *wsdl:input* element defined in the *messages* of operations), $\mathcal{O}$ is the set of outputs (*wsdl:output*) and $\mathcal{T}$ is a text (*wsdl:documentation*).

Example: *S*=<{DepartureAirport,ArrivalAirport,OutboundDate,InboundDate, RoundTrip, AcctName, Password, Confirm},{PreferredFlightItinerary, FlightsFound, ReservationID},"*This service provides flight …*">

SAWSDL. Semantic Annotations for WSDL and XML Schema (SAWSDL) [10] introduces three new attributes: *modelReference* for specifying associations between WSDL components and semantic concepts, *liftingSchemaMapping* and *loweringSchemaMapping* for specifying mappings between semantic data and XML. Some of the elements might not be semantically annotated. For us, SAWSDL service is described by $S = <\mathcal{I}, \mathcal{O}, \mathcal{T}>$ where $\mathcal{I}=<\mathcal{I}_{syn}, \mathcal{I}_{sem}>$ is a pair of sets containing the inputs not semantically annotated ($\mathcal{I}_{syn}$) and those annotated with concepts from ontologies ($\mathcal{I}_{sem}$), analogously for $\mathcal{O}=<\mathcal{O}_{syn}, \mathcal{O}_{sem}>$, and $\mathcal{T}$ is the description.

Example: *S*=<<*ϕ*,{ba:DepartureAirport,ba:ArrivalAirport,ba:OutboundDate, ba:InboundDate, ba:RoundTrip, ba:AcctName, ba:Password, ba:Confirm}>, <*ϕ*,{ba:FlightsFound, ba:PreferredFlightItinerary, ba:ReservationID}>, "*This service provides flight …*">

Keyword-based descriptions. A typical way of describing resources is by annotating them using keywords (or tags). Service descriptions are also possible following this approach. For example, *seekda!*[5] web service search engine includes free text and tag descriptions provided by service users, as well as WSDL files. We admit two kinds of keyword-based descriptions ($\mathcal{K} = <\mathcal{K}_{syn}, \mathcal{K}_{sem}>$): syntactic (free text) and semantic (ontology-based). Example: S = <{flight,reservation,airport,date,trip},*ϕ* >

Tag Cloud based descriptions. The advantage of using tags () in the context of service discovery is that they supply a user-defined vocabulary based on a consensus of how the service is perceived or used in the world [7, 17]. A tag-cloud-based service description is a set of pairs $\mathcal{TC} = \{<t, n>\}$, where *t* is a free text tag and *n* is the frequency of the tag *t* in the cloud.

For example: S = <*{*<flight,*6*>, <reservation,*4*>, <trip,*3*>, <airport,*2*>*}*>

Textual descriptions. Services can be defined by free text human-readable information. For these kinds of descriptions service discovery is based on information retrieval techniques (IR). S = "*This*"*This service provides flight …*"

[5] http://webservices.seekda.com/

3 Service Model Alignment

In this section, we present a common model for service discovery purposes that integrates the relevant characteristics of the service description languages analysed in the previous section. Afterwards we describe the mapping from the original models.

3.1 Integrated Model for service discovery

From the analysis of the different approaches to service descriptions (section 2) we obtain an integrated *general common model (GCM)*. Let $\mathcal{N}$ be a set of concepts of domain ontologies, a *GCM* for service discovery is a tuple $<\mathcal{I}_{GCM}, \mathcal{O}_{GCM}, \mathcal{P}_{GCM}, \mathcal{E}_{GCM}, \mathcal{K}_{GCM}, \mathcal{C}_{GCM}, \mathcal{T}_{GCM}, \mathcal{TC}_{GCM}>$, where:

- $\mathcal{I}_{GCM} = <I_{syn}, I_{sem}>$ is the set of syntactic ($I_{syn} \in \{a, .., z\}^*$) and semantic ($I_{sem} \subseteq N$) inputs of the service.
- $\mathcal{O}_{GCM} = <O_{syn}, O_{sem}>$ is the set of syntactic ($O_{syn} \subseteq \{a, ..., z\}^*$) and semantic ($O_{sem} \subseteq N$) outputs.
- $\mathcal{P}_{GCM}$ is the set of preconditions. $\mathcal{P}_{GCM} \subseteq \mathcal{N}$
- $\mathcal{E}_{GCM}$ is the set of effects. $\mathcal{E}_{GCM} \subseteq \mathcal{N}$
- $\mathcal{K}_{GCM} = <\mathcal{K}_{syn}, \mathcal{K}_{sem}>$ is the set of syntactic and semantic keywords, where $\mathcal{K}_{syn} \subseteq \{a, \ldots, z\}^*$, $\mathcal{K}_{sem} \subseteq \mathcal{N}$.
- $\mathcal{C}_{GCM}$ is a set of categories of the service, described semantically ($C_{sem} \subseteq \mathcal{N}$) (e.g. NAICS or UNSPSC).
- $\mathcal{T}_{GCM}$ is a textual description of the service.
- $\mathcal{TC}_{GCM}$ is a tag cloud. $\mathcal{TC}_{GCM} = \{<t, n> | \ t \subseteq \{a, \ldots, z\}^*, n \in \mathbb{N}\}$.

We opt for using RDF as the representation language for the *GCM*. This will also allow the exploitation of SPARQL [20] to query service descriptions.

3.2 Mapping service descriptions into the *GCM*

Table 1 shows how the different elements of the *GCM* can be obtained from each source service description model. The first column specifies the element of the *GCM*, while each cell contains the value mapped from the model shown in the first row.

There are many straightforward mappings that consist of simple associations between parameters in both models. For instance, in OWLS/WSMO $\mathcal{I}_{GCM}=<\phi, \mathcal{I}>$ because they provide semantically described inputs $\mathcal{I}$. The contrary applies to WSDL, where only the syntactic values are filled ($\mathcal{I}_{GCM}=<\mathcal{I}, \phi>$). However, SAWSDL may contain both syntactic and semantic descriptions, thus $\mathcal{I}_{GCM}=\mathcal{I}$ since $\mathcal{I} = <I_{syn}, I_{sem}>$. The same is applied to the outputs. Trivial mappings apply to preconditions ($\mathcal{P}_{GCM}$), effects ($\mathcal{E}_{GCM}$), categories ($\mathcal{C}_{GCM}$) and textual descriptions ($\mathcal{T}_{GCM}$).

However, some fields (tag-clouds, keywords) may not be explicitly described by a given model but they can be obtained from the rest of the description. **Tag-clouds** can be calculated from textual descriptions by means of a function $\Delta(\mathcal{T})$, which returns the *k most relevant* words from the text $\mathcal{T}$ as well as their frequency. We adopt information retrieval (IR) techniques to obtain that information through a process of (i) word extraction, (ii) stemming, and (iii) filtering out non relevant terms (chosen heuristically, such as the words *service, input, output*, etc.). For the examples in section 2, $\mathcal{TC}_{GCM}$ = *{<flight, 3>, <airport, 2>, <reservation, 2>, <departure, 2>}*. In the case of keyword-based service descriptions (where no text is included), a plain cloud is created with frequency 1 for every keyword in the description.

Syntactic **keywords** can be easily obtained from tag clouds (either original or calculated with Δ), by simply adopting the *k* most relevant words (function τ(*TC*), being *TC* a tag-cloud), e.g. $\mathcal{K}_{GCM}$ = *<{flight, airport, reservation, departure}, ϕ>*.

Input and output concepts in semantic descriptions (OWL-S, WSMO, SAWSDL) are adopted as semantic keywords. For their examples: $\mathcal{K}_{GCM}$ = *<{flight, airport, reservation, departure}, {ba:DepartureAirport, ba:ArrivalAirport, ba:OutboundDate, ba:InboundDate, ba:RoundTrip, ba:AcctName, ba:Password, ba:Confirm, ba:FlightsFound, ba:PreferredFlightItinerary, ba:ReservationID} >*

Table 1. Service(S)-to-*GCM* mapping

GCM	**OWL-S / WSMO**	**SAWSDL**	**WSDL**	**Keyword (tag)**	**Tag Cloud**	**Text**
$\mathcal{I}_{GCM}$	$<\phi, \mathcal{I}>$	$\mathcal{I}$	$<\mathcal{I}, \phi>$	$<\phi, \phi>$	$<\phi, \phi>$	$<\phi, \phi>$
$\mathcal{O}_{GCM}$	$<\phi, \mathcal{O}>$	$\mathcal{O}$	$<\mathcal{O}, \phi>$	$<\phi, \phi>$	$<\phi, \phi>$	$<\phi, \phi>$
$\mathcal{P}_{GCM}$	$\mathcal{P}$	ϕ	ϕ	ϕ	ϕ	ϕ
$\mathcal{E}_{GCM}$	$\mathcal{E}$	ϕ	ϕ	ϕ	ϕ	ϕ
$\mathcal{K}_{GCM}$	$<\tau(\Delta(\mathcal{T})), \mathcal{I} \cup \mathcal{O}>$	$<\tau(\Delta(\mathcal{T}), \mathcal{I} \cup \mathcal{O}>$	$<\tau(\Delta(\mathcal{T}), \phi>$	$\mathcal{K}$	$<\tau(S), \phi>$	$<\tau(\Delta(S), \phi>$
$\mathcal{C}_{GCM}$	$\mathcal{C}$	$\mathcal{C}$	ϕ	ϕ	ϕ	ϕ
$\mathcal{T}_{GCM}$	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{T}$	ϕ	ϕ	S
$\mathcal{TC}_{GCM}$	$\Delta(\mathcal{T})$	$\Delta(\mathcal{T})$	$\Delta(\mathcal{T})$	$\{<t, 1> \mid t \in \mathcal{K}_{syn} \cup \mathcal{K}_{sem}\}$	S	$\Delta(S)$

4 Conclusions

In this paper we have dealt with the problem of service discovery in open systems. In particular, we discussed in detail the alignment of service description models, and the transformation of them into a RDF common model. Although we provided with an alignment mechanism for a set of service description languages, other languages can be easily integrated into. In effect, if such new model fits into the proposed *GCM* only the adequate mappings have to be specified. Otherwise, new characteristics (e.g. geographical radius, privacy aspects…) might be added to the *GCM* to account for those new languages, and considering them empty for the previous models (additionally those legacy models might be completed with the corresponding mappings to the new characteristics).

This work is part of a proposed framework able to locate adequate services in open heterogeneous environments [1]. The alignment of the service description models (advertisement and request services) is a previous step to the service matchmaking process. The evaluation of the proposed framework is part of our ongoing work.

5 References

1. Balta, A. and Fernandez, A. Towards an abstract architecture for service discovery with semantic alignment. Workshop on Agreement technologies (WAT), CAEPIA, 2009.
2. Bruijn, J. et al. Web Service Modeling Ontology (WSMO). W3C Member Submission, 2005. http://www.w3.org/Submission/WSMO/.
3. Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M. N., Paolucci, M., Sheth, A. P., and Williams, S. A Semantic Web Services Architecture. *IEEE Internet Computing* 9, 5, 72-81. 2005.
4. E. Christensen, F. Curbera, G. Meredith, and S.Weerawarana. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl, March 2001
5. Ehrig M. Ontology Alignment: Bridging the Semantic Gap. Springer. 2007.
6. Euzenat, J. and Shvaiko P. Ontology matching. Springer, Heidelberg (DE), 2007
7. Fernandez, A., Hayes, C., Loutas, N., Peristeras, V., Polleres, A., Tarabanis, K. Closing the service discovery gap by collaborative tagging and clustering techniques. Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2). 2008
8. Fernandez, A., Polleres, A., and Ossowski, S. Towards Fine-grained Service Matchmaking by Using Concept Similarity, ISWC-2007 Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2). 2007
9. Giantsiou, L., Loutas, N., Peristeras, V. and Tarabanis, K. Semantic Service Search Engine (S3E): An Approach for Finding Services on the Web. In LNCS, Volume 5736/2009, pages 316-325. Springer, 2009.
10. Joel Farrell and Holger Lausen. Semantic Annotations for WSDL and XML Schema (SAWSDL). W3C Recommendation 28 August 2007. http://www.w3.org/TR/sawsdl/
11. Kourtesis, D., Paraskakis, I. Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery. Springer Berlin / Heidelberg. Pages 614-628. 2008.
12. Klusch, M., Fries, B., and Sycara, K. Automated semantic web service discovery with owls-mx. In AAMAS ’06: fifth international joint conference on Autonomous agents and multiagent systems, pages 915–922, New York, USA, 2006. ACM Press.
13. Li, L and Horrocks, I. A software framework for matchmaking based on semantic web technology. Int. J. of Electronic Commerce, 8(4):39–60, 2004.
14. Martin, D. et al. OWL-S: Semantic Markup for Web Services. W3C Submission, 2004. http://www.w3.org/Submission/OWL-S/.
15. Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. Semantic Matching of Web Service Capabilities. In ISWC, pages 333–347. Springer Verlag, 2002.
16. Paolucci, M., Wagner M., Martin M. Grounding OWL-S in SAWSDL. Springer Berlin / Heidelberg. Pages 416-421. 2007.
17. Papathanasiou, M., Loutas, N., Peristeras, V., Tarampanis, K. Combining Service Models, Semantic and Web 2.0 Technologies to Create a Rich Citizen Experience. M.D. Lytras et al. (Eds.): WSKS 2009, LNAI 5736, pp. 296–305, 2009
18. Polleres, A., Lara, R. A Conceptual Comparison between WSMO and OWL-S, WSMO Working Group working draft, 2005. http://www.wsmo.org/2004/d4/d4.1/v0.1/.
19. Sheth, A., Gomadam, K., Lathem, J.: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. IEEE Internet Computing 11(6), 91–94 (2007).
20. W3C World Wide Web Consortium. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. http://www.w3.org/TR/rdf-sparql-query/.