Web Semantics: Science, Services and Agents on the World Wide Web 31 (2015) 39–58

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

WSMO-Lite and hRESTS: Lightweight semantic annotations for Web services and RESTful APIs

Dumitru Roman [a,*], Jacek Kopecký [b], Tomas Vitvar [c], John Domingue [e], Dieter Fensel [d]

[a] SINTEF, Forskningsveien 1, 0314 Oslo, Norway

[b] School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, UK

[c] Institut für Informatik, University of Innsbruck, Technikerstraße 21a, 6020 Innsbruck, Austria

[d] STI Innsbruck, University of Innsbruck, Technikerstraße 21a, 6020 Innsbruck, Austria

[e] Knowledge Media Institute, The Open University, Walton Hall, MK7 6AA Milton Keynes, UK

ARTICLE INFO

Article history:
Received 17 February 2014
Received in revised form
24 November 2014
Accepted 26 November 2014
Available online 3 December 2014

Keywords:
WSMO-Lite
SAWSDL
Web services
RESTful services

ABSTRACT

Service-oriented computing has brought special attention to service description, especially in connection with semantic technologies. The expected proliferation of publicly accessible services can benefit greatly from tool support and automation, both of which are the focus of Semantic Web Service (SWS) frameworks that especially address service discovery, composition and execution. As the first SWS standard, in 2007 the World Wide Web Consortium produced a lightweight bottom-up specification called SAWSDL for adding semantic annotations to WSDL service descriptions. Building on SAWSDL, this article presents WSMO-Lite, a lightweight ontology of Web service semantics that distinguishes four semantic aspects of services: function, behavior, information model, and nonfunctional properties, which together form a basis for semantic automation. With the WSMO-Lite ontology, SAWSDL descriptions enable semantic automation beyond simple input/output matchmaking that is supported by SAWSDL itself. Further, to broaden the reach of WSMO-Lite and SAWSDL tools to the increasingly common RESTful services, the article adds hRESTS and MicroWSMO, two HTML microformats that mirror WSDL and SAWSDL in the documentation of RESTful services, enabling combining RESTful services with WSDL-based ones in a single semantic framework. To demonstrate the feasibility and versatility of this approach, the article presents common algorithms for Web service discovery and composition adapted to WSMO-Lite.

1. Introduction

The emergence of service-oriented computing has brought special attention to the area of service modeling. Service descriptions are a fundamental element that enables such tasks as service discovery, composition and mediation. As argued by [1], semantic interoperability is crucial for Web services, and the technologies of the Semantic Web have the potential to provide the needed level of interoperability. In the recent years, research into semantic Web service (SWS) models and descriptions has received much attention and funding: the major research projects include the US-based OWL-S initiative,[1] and the European projects DIP, SUPER and SOA4All.[2] In these and other projects, researchers have proposed several frameworks for semantic Web services (SWS), especially OWL-S, WSMO, and WSDL-S.

OWL-S [2] and WSMO [3] embody a *semantics-first* approach to semantic modeling of Web services: in both frameworks, the semantic description of a service is conceptually independent of the underlying technical description (e.g. WSDL [4]), and there is a *grounding* mechanism to connect the semantics with the technical descriptions. In contrast, WSDL-S [5] attaches semantic annotations directly to the underlying technical descriptions, in effect building on the Web service model of WSDL.

Service-oriented computing, revolving around the so-called WS–∗ family of technologies (based on the SOAP protocol [6] and the WSDL service description language), is heavily driven by standardization. The approach embodied by WSDL-S turned out to

* Corresponding author.
E-mail addresses: dumitru.roman@sintef.no (D. Roman), jacek.kopecky@port.ac.uk (J. Kopecký), tomas@vitvar.com (T. Vitvar), john.domingue@open.ac.uk (J. Domingue), dieter.fensel@sti2.at (D. Fensel).

[1] http://www.daml.org/services/owl-s/.

[2] dip.semanticweb.org, ip-super.org, soa4all.eu.

http://dx.doi.org/10.1016/j.websem.2014.11.006
1570-8268/

be an acceptable common ground for both the research community and industry; based on WSDL-S, the World Wide Web Consortium (W3C) produced in 2007 its Recommendation called SAWSDL—Semantic Annotations for WSDL and XML Schema [7]. SAWSDL is not a fully-fledged SWS framework; instead it only provides hooks in WSDL where semantic annotations can be attached, leaving the specification and standardization of concrete service semantics for later. Modeling service semantics, within the framework of SAWSDL, is the focus of our work.

In addition to WS−∗, in recent years another approach to service-oriented computing has started gaining traction: *RESTful Web services* [8], often also called Web APIs. RESTful services are a common part of Web applications, as the applications' functionalities are increasingly used by programmatic clients, such as other Web applications. RESTful services are especially present in Web applications that use rich *Ajax*[3] user interfaces, where the JavaScript code running in the browser is a programmatic client, for which the Web application must provide an appropriate access interface. RESTful Web services use native Web technologies around HTTP [9], the Web architecture [10], and standard data formats, resulting in effective integration with the Web. RESTful services are proliferating in part because the Web technologies used by RESTful Web service are perceived as simpler than the WS−∗ technologies [11].

In [11] the authors analyze the architectural differences between WS−∗ and RESTful services. Among the key differences between WS−∗ and RESTful services is their client-facing structure: where a WS−∗ service exposes a single endpoint that handles all the operations of the service, a non-trivial RESTful service exposes a number of independently meaningful and addressable resources.

Irrespective of the types of services (WS−∗ vs. RESTful), the availability of service annotations is a prerequisite for service automation in discovery, composition, mediation, etc. Treating service annotations uniformly for both WS−∗ and RESTful services would not only enable better interoperability between them but would also enable technologies for service automation to be applicable across both types of services. In our work, we reconcile the two different kinds of services and we show that most semantic automation algorithms can disregard the difference, eventually enabling more scalable and robust mechanisms for WS−∗ and RESTful services tasks.

To achieve this reconciliation, we define hRESTS and MicroWSMO – two HTML microformats that mirror WSDL and SAWSDL over the HTML documentation of RESTful services – enabling us to create WSMO-Lite semantic descriptions of RESTful services. Furthermore, we develop a concrete ontology of service semantics (WSMO-Lite) for use with the standard SAWSDL, and we broaden the applicability of SAWSDL to RESTful services and Web APIs (through hRESTS and MicroWSMO). Finally, to demonstrate the viability of our SWS approach, we adapt automation algorithms for service discovery and composition to WSMO-Lite.

The overall purpose of this paper is to give a comprehensive overview of the WSMO-Lite approach for reconciling WS−∗ and RESTful services, to demonstrate the feasibility of the approach, and to serve as an entry and reference point for interested parties in using the WSMO-Lite approach.

Parts of this paper are based upon, and extend work reported in [12,13]. [12] introduced the annotation of WSDL services with WSMO-Lite, and [13] introduced the hRESTS microformat, however, no previous publication provided this comprehensive overview of the WSMO-Lite approach, or the feasibility analysis on how SWS tasks can be applied to the approach. To the extent of our knowledge, WSMO-Lite is the first principled effort focused on defining an ontology of service semantics for SAWSDL service descriptions.

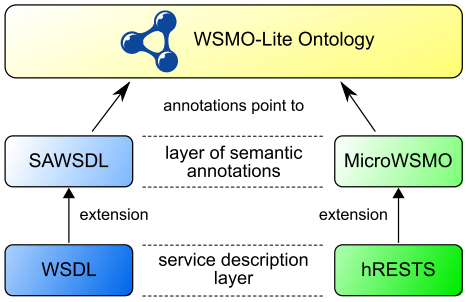

Fig. 1. Semantic Web service descriptions with WSMO-Lite.

The rest of this article is organized as follows. Section 2 provides an overview of the WSMO-Lite framework for lightweight semantic service descriptions, and highlights some design considerations important for our work. Section 3 introduces a minimal service model that captures the common structure of Web services and bridges the worlds of WS−∗ and RESTful Web services. Section 4 formally defines the WSMO-Lite ontology for service semantics, showing how it fits WSDL and SAWSDL. Section 5 shows how we apply SAWSDL and WSMO-Lite to RESTful services—it defines the two microformats, hRESTS and MicroWSMO, for annotating RESTful services' HTML documentation. Section 6 presents concrete algorithms for service discovery and composition, adapted to WSMO-Lite from earlier SWS research. Section 7 presents an analysis of feasibility of the overall approach from different angles (tooling support, viability), Section 8 discusses related work and the limitations of WSMO-Lite, and Section 9 concludes the article with a summary of the core contributions and with a discussion of future work.

2. Overall approach and design principles

Fig. 1 shows the core technologies that make up our framework. Our work stems from the standardization of SAWSDL; as illustrated on the left-hand side of the figure, SAWSDL extends the service description language WSDL with semantic annotations. For the content of the annotations, we define the *WSMO-Lite ontology* of service semantics, where we identify the kinds of service semantics that are necessary to support SWS automation.

The right-hand side of the figure shows our technologies for describing and semantically annotating RESTful services: *hRESTS and MicroWSMO*. WSMO-Lite builds on a unified minimal service model that extracts the concepts common in WS−∗ and RESTful Web services. Based on this model, hRESTS (HTML for RESTful Services) is a microformat[4] for structuring common HTML documentation of RESTful APIs to make it machine-processible, analogously to how WSDL provides machine-processible descriptions on the WS−∗ side.[5] MicroWSMO (Microformat for WSMO-Lite) is an extension of hRESTS that adds SAWSDL annotation properties, where WSMO-Lite semantics can be attached.

Our work is guided by the following design principles:

Proximity to underlying standards. In semantics-first frameworks (OWL-S, WSMO), the semantic description of a service is

[3] http://en.wikipedia.org/wiki/Ajax_(programming).

[4] Microformats are an approach for annotating mainly human-oriented Web pages so that key information is machine-readable [14].

[5] There is currently no accepted equivalent of WSDL for RESTful services; the Web Application Description Language (WADL, [15]) is a RESTful (resource-oriented) alternative to WSDL, but it is not commonly accepted.