SmartLink: A Web-Based Editor and Search Environment for Linked Services

Stefan Dietze, Hong Qing Yu, Carlos Pedrinaci, Dong Liu, and John Domingue

Knowledge Media Institute, The Open University, MK7 6AA, Milton Keynes, UK
{s.dietze,h.q.yu,c.pedrinaci,d.liu,j.b.domingue}@open.ac.uk

Abstract. Despite considerable research dedicated to Semantic Web Services (SWS), structured semantics are still not used significantly to annotate Web services and APIs. This is due to the complexity of comprehensive SWS models and has led to the emergence of a new approach dubbed *Linked Services.* Linked Services adopt Linked Data principles to produce simplified, RDF-based service descriptions that are easier to create and interpret. However, current Linked Services editors assume the existence of services documentation in the form of HTML or WSDL files. Therefore, we introduce SmartLink, a Web-based editor and search environment for Linked Services. Based on an easy-to-use Web form and a REST-ful API, SmartLink allows both humans as well as machines to produce light-weight service descriptions from scratch.

Keywords: Semantic Web Services, Linked Services, Linked Data, SmartLink.

1 Introduction

The past decade has seen a range of research efforts in the area of Semantic Web Services (SWS), aiming at the automation of Web service-related tasks such as discovery, orchestration or mediation. Several conceptual models, such as OWL-S [6], WSMO [3], and standards like SAWSDL [7] have been proposed, usually covering aspects such as service capabilities, interfaces and non-functional properties. However, SWS research has for the most part targeted WSDL or SOAP-based Web services, which are not prevalent on the Web. Also, due to the inherent complexity required to fully capture computational functionality, creating SWS descriptions has represented an important knowledge acquisition bottleneck and required the use of rich knowledge representation languages and complex reasoners. Hence, so far there has been little take up of SWS technology within non-academic environments.

That is particularly concerning since Web services – nowadays including a range of often more light-weight technologies beyond the WSDL/SOAP approach, such as RESTful services or XML-feeds – are in widespread use throughout the Web. That has led to the emergence of more simplified SWS approaches to which we shall refer here as "lightweight", such as WSMO-Lite [9] SA-REST [7] and Micro-WSMO/hRESTs [4] which replace "heavyweight" SWS with simpler models expressed in RDF(S) which aligns them with current practices in the growing Semantic Web [1] and simplifies the creation of service descriptions. While the Semantic Web has successfully redefined

G. Antoniou et al. (Eds.): ESWC 2011, Part II, LNCS 6644, pp. 436–440, 2011.

itself as a Web of *Linked (Open) Data (LOD)* [2], the emerging *Linked Services* approach [7] exploits the established LOD principles for service description and publication, and is catering for exploiting the complementarity of the Linked Data and services to support the creation of advanced applications for the Web.

In order to support annotation of a variety of services, such as WSDL services as well as REST APIs, the EC-funded project SOA4ALL[1], has developed the Linked Services registry and discovery engine iServe[2]. iServe supports publishing service annotations as linked data expressed in terms of a simple conceptual model that is suitable for both human and machine consumption and abstracts from existing heterogeneity of service annotation formalisms: the Minimal Service Model (MSM). The MSM is a simple RDF(S) ontology able to capture (part of) the semantics of both Web services and Web APIs. While MSM [7] is extensible to benefit from the added expressivity of other formalisms, iServe allows import of service annotations following, for instance, SAWSDL, WSMO-Lite, MicroWSMO, or OWL-S. Once imported, service annotations are automatically published on the basis of the Linked Data principles. Service descriptions are thus accessible based on resolvable HTTP URIs by utilising content negotiation to return service instances in either plain HTML or RDF. In addition to a SPARQL endpoint, a REST API allows remote applications to publish annotations and to discover services through an advanced set of discovery strategies that combine semantic reasoning and information retrieval techniques. In order to support users in creating semantic annotations for services two editors have been developed: SWEET [5] (SemanticWeb sErvices Editing Tool) and SOWER (SWEET is nOt a Wsdl EditoR), which support users in annotating Web APIs and WSDL services respectively.

However, SWEET and SOWER build on the assumption that either HTML documentation of services/APIs (SWEET) or WSDL files (SOWER) are available as starting point for annotation. While that holds for a certain set of services, a number of services on the Web neither provide a WSDL nor an HTML documentation and hence, current Linked Services editors cannot be deployed in a range of cases. In addition, we would like to promote an approach were services documentation relies on structured RDF(S) and additional human-readable documentation is not provided manually but automatically generated to avoid redundancies. Therefore, we introduce and demonstrate *SmartLink*, an editing and search environment for Linked Services addressing the issues described above.

2 SmartLink: Linked Services Editor and Search Environment

In order to provide a Linked Services editor which allows the annotation of REST-ful services without any pre-existing documentation, a new services annotation and search tool was created, *SmartLink*[3] ("SeMantic Annotation enviRonmenT for Linked services"). SmartLink allows annotation of REST-ful services based on the MSM from scratch, that is, without any pre-existing services documentation such as WSDL or HTML files, as assumed by existing annotation tools (Section 1). SmartLink

[1] http://www.soa4all.eu/
[2] http://iserve.kmi.open.ac.uk
[3] http://smartlink.open.ac.uk & http://kmi.open.ac.uk/technologies/name/smartlink

operates on top of LOD stores such as iServe and is an open environment accessible to users simply via OpenID[4] authentication.

SmartLink exploits an extension of the MSM schema including a number of additional non-functional properties. These non-functional properties cover, for instance, contact person, developer name, Quality of Service (QoS), development status, service license, and WSMO goal reference. The latter property directly contributes to facilitate our approach of allowing MSM models to refer to existing WSMO goals which utilize the same service entity. MSM-schema properties are directly stored in iServe, while additional properties are captured in a complementary RDF store based on OpenRDF Sesame[5]. Due to the SmartLink-specific extensions to the MSM, we refer in the following to our Linked Services RDF store as *iServe+*. The following figure depicts the overall architecture of the SmartLink environment.

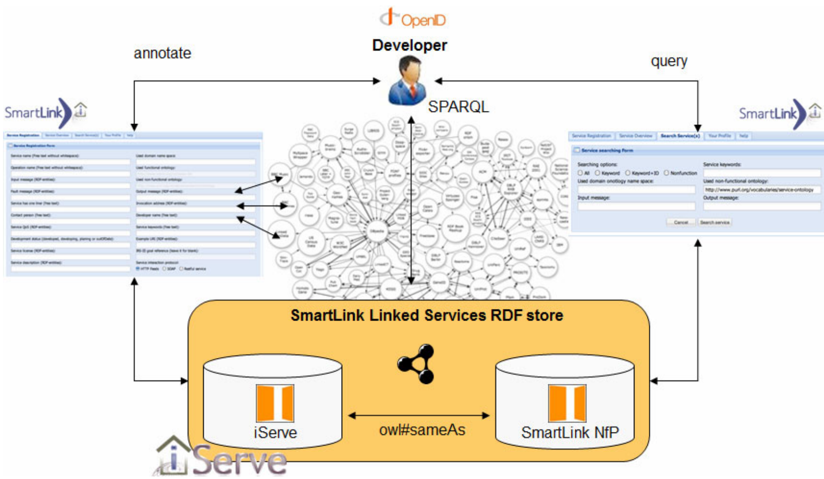

Fig. 1. SmartLink – overall architecture

SmartLink allows developers to directly annotate existing RESTful services and APIs, which potentially also includes the annotation of WSMO goal requests – which in fact are RESTful services themselves – as MSM service instances. Figure 2 depicts the SmartLink Web interface showing the service editing form and the services library.

Being an LOD-compliant environment, one of the core features of the MSM is the capability to associate service descriptions with so-called *model references* which refer to RDF descriptions in external vocabularies defining the semantics of the service or its parts. That way, for instance, a particular *service response message* can be associated with an external RDF description which details and further describes the nature of the response. However, while this feature is useful and even necessary in order to provide meaningful service models, finding appropriate model references across the entire Web of data is a challenging task. Therefore, SmartLink uses established Linked Data

[4] http://openid.net/
[5] http://www.openrdf.org/

APIs – currently the WATSON[6] API - to identify and recommend suitable model references to the user.

Dedicated APIs allow machines and third party applications to interact with iServe+, e.g., to submit service instances or to discover and execute services. In addition, the Web application provides a search form which allows to query for particular services. Service matchmaking is being achieved by matching a set of core properties (input, output, keywords), submitting SPARQL queries, and a dedicated set of APIs.

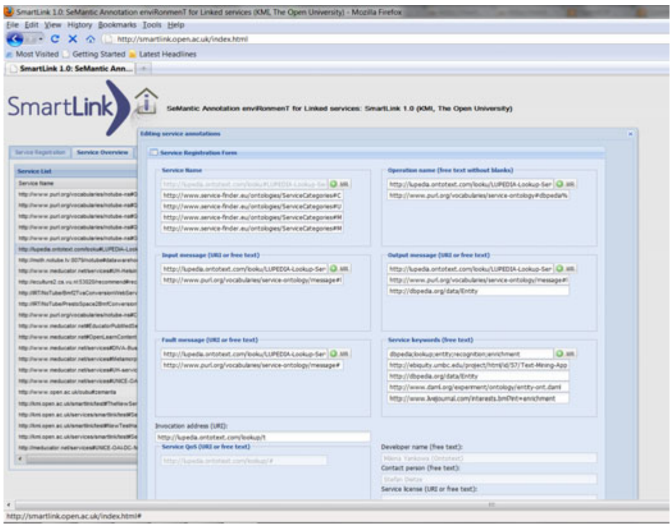

Fig. 2. SmartLink – Web interface

SmartLink currently provides mechanisms which enable the export of particular (MSM) service instances as RDF or human-readable HTML. In order to facilitate service model transformation between MSM and other SWS formalisms, current research deals with the establishment of an export mechanism of MSM services.

3 Discussion and Conclusion

Current work deals with a first exploitation of SmartLink in the context of the NoTube project[7] where the ultimate goal is to develop a network of services, connected through the use of semantics, to personalize consumption of digital (IP)TV content. NoTube adopts the Linked Services approach by utilising the iServe+ and SmartLink tools. In addition, we have devised a functional classification for services specific to the NoTube domain, stored and exposed via our iServe+ environment. From our initial use case, a few observations have been made which will shape our

[6] http://watson.kmi.open.ac.uk/WatsonWUI/
[7] http://www.notube.tv

future efforts. For instance, the recommendation of LOD model references via open APIs proved very useful to aid SmartLink users when annotating services. However, due to the increasing number of LOD datasets – strongly differing in terms of quality and usefulness – it might be necessary in the future to select recommendations only based on a controlled subset of the LOD cloud in order to reduce available choices.

While SmartLink proved beneficial when creating light-weight service annotations, the lack of service automation and execution support provided by our extended MSM models, and, more importantly, the current tool support, made it necessary to transform and augment these models to into more comprehensive service models (WSMO). Due to the lack of overlap between concurrent SWS models, transformation is a manual and costly process. Hence, our current research and development deals with the extension of the MSM by taking into account execution and composition oriented aspects and the development of additional APIs, which allow the discovery, execution and semi-automated composition of Linked Services, and make the exploitation of additional SWS approaches obsolete.

Acknowledgments

This work is partly funded by the European projects NoTube and mEducator. The authors would like to thank the European Commission for their support.

References

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American Magazine (2001) (retrieved March 29, 2009)

[2] Bizer, C., Heath, T., et al.: Linked data - The Story So Far. Special Issue on Linked data. International Journal on Semantic Web and Information Systems, IJSWIS (2009)

[3] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: Enabling Semantic Web Services: The Web Service Modeling Ontology. Springer, Heidelberg (2007)

[4] Kopecky, J.; Vitvar, T.; and Gomadam, K. MicroWSMO. Deliverable, Conceptual Models for Services Working Group (2008), http://cms-wg.sti2.org/TR/d12/v0.1/20090310/d12v01_20090310.pdf

[5] Maleshkova, M., Pedrinaci, C., Domingue, J.: Supporting the creation of semantic restful service descriptions. In: 8th International Semantic Web Conference on Workshop: Service Matchmaking and Resource Retrieval in the Semantic Web, SMR2 (2009)

[6] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services. Member submission, W3C. W3C Member Submission, November 22 (2004)

[7] Pedrinaci, C., Domingue, J.: Toward the Next Wave of Services: Linked Services for the Web of Data. Journal of Universal Computer Science 16(13), 1694–1719 (2010)

[8] Sheth, A.P., Gomadam, K., Ranabahu, A.: Semantics enhanced services: Meteor-s, SAWSDL and SA-REST. IEEE Data Eng. Bull. 31(3), 8–12 (2008)

[9] Vitvar, T., Kopecky, J., Viskova, J., Fensel, D.: Wsmo-lite annotations for web services. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, Springer, Heidelberg (2008)