

Fostering a Relationship between Linked Data and the Internet of Services

John Domingue¹, Carlos Pedrinaci¹, Maria Maleshkova¹, Barry Norton², and Reto Krummenacher³

¹ Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK6 7AA UK

{j.b.domingue, c.pedrinaci, m.maleshkova}@open.ac.uk

² Karlsruhe Institute of Technology, Karlsruhe, Germany

barry.norton@aifb.uni-karlsruhe.de

³ Semantic Technology Institute, University of Innsbruck, 6020 Innsbruck, Austria
reto.krummenacher@sti2.at

Abstract. We outline a relationship between Linked Data and the Internet of Services which we have been exploring recently. The Internet of Services provides a mechanism for combining elements of a Future Internet through standardized service interfaces at multiple levels of granularity. Linked Data is a lightweight mechanism for sharing data at web-scale which we believe can facilitate the management and use of service-based components within global networks.

Keywords: Linked Data, Internet of Services, Linked Services

1 Introduction

The Future Internet is a fairly recent EU initiative which aims to investigate scientific and technical areas related to the design and creation of a new global infrastructure. An overarching goal of the Future Internet is that the new platform should meet Europe's economic and societal needs. The Internet of Services is seen as a core component of the Future Internet:

“The Future Internet is polymorphic infrastructure, where the boundaries between silo systems are changing and blending and where the emphasis is on the integration, interrelationships and interworking of the architectural elements through new service-based interfaces”. [Frederic Gittler, FIA Stockholm]

The Web of Data is a relatively recent effort derived from research on the Semantic Web [1], whose main objective is to generate a Web exposing and interlinking data previously enclosed within silos. Like the Semantic Web the Web of Data aims to extend the current human-readable Web with data formally represented so that software agents are able to process and reason with the information in an automatic and

flexible way. This effort, however, is based on the simplest form of semantics, RDF(S) [2], and has thus far focused on promoting the publication, sharing and linking of data on the Web.

From a Future Internet perspective a combination of service-orientation and Linked Data provides possibilities for supporting the integration, interrelationship and interworking of Future Internet components in a partially automated fashion through the extensive use of machine-processable descriptions. From an Internet of Services perspective, Linked Data with its relatively simple formal representations and in-built support for easy access and connectivity provides a set of mechanisms supporting interoperability between services. In fact, the integration between services and Linked Data is increasingly gaining interest within industry and academia. Examples include, for instance, research on linking data from RESTful services by Alarcon et al. [3], work on exposing datasets behind Web APIs as Linked Data by Speiser et al. [4], and Web APIs providing results from the Web of Data like Zemanta¹.

We see that there are possibilities for Linked Data to provide a common ‘glue’ as services descriptions are shared amongst the different roles involved in the provision, aggregation, hosting and brokering of services. In some sense service descriptions as, and interlinked with, Linked Data is complementary to SAP’s Unified Service Description Language² [5], within their proposed Internet of Services framework³, as it provides appropriate means for exposing services and their relationships with providers, products and customers in a rich, yet simple manner which is tailored to its use at Web scale.

In this paper we discuss the relationship between Linked Data and services based on our experiences in a number of projects. Using what we have learnt thus far, at the end of the paper we propose a generalization of Linked Data and service principles for the Future Internet.

2 Linked Data

The Web of Data is based upon four simple principles, known as the Linked Data principles [6], which are:

1. Use URIs (Uniform Resource Identifiers) as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using standards (RDF*, SPARQL).
4. Include links to other URIs, so that they can discover more things.

¹ <http://developer.zemanta.com/>

² <http://www.internet-of-services.com/index.php?id=288&L=0>

³ <http://www.internet-of-services.com/index.php?id=260&L=0>

On the corporate side, the BBC has been making use of RDF descriptions for some time. BBC Backstage⁸ allows developers to make use of BBC programme data available as RDF. The BBC also made use of scalable RDF repositories for the back-end of the BBC world cup website⁹ to facilitate “agile modeling”¹⁰. This site was very popular during the event receiving over 2 million queries per day.

Other examples of commercial interest include: the acquisition of Metaweb¹¹ by Google to enhance search, and the release of the OpenGraph¹² API by Facebook. Mark Zuckerberg, Facebook’s CEO claimed recently that Open Graph was the “the most transformative thing we’ve ever done for the Web”¹³.

3 Services on the Web

Currently the world of services on the Web is marked by the formation of two main groups of services. On the one hand, “classical” Web services, based on WSDL and SOAP, play a major role in the interoperability within and among enterprises. Web services provide means for the development of open distributed systems, based on decoupled components, by overcoming heterogeneity and enabling the publishing and consuming of functionalities of existing pieces of software. In particular, WSDL is used to provide structured descriptions for services, operations and endpoints, while SOAP is used to wrap the XML messages exchanged between the service consumer and provider. A large number of additional specifications such as WS-Addressing, WS-Messaging and WS-Security complement the stack of technologies.

On the other hand, an increasing number of popular Web and Web 2.0 applications as offered by Facebook, Google, Flickr and Twitter offer easy-to-use, publicly available Web APIs, also referred to as RESTful services (properly when conforming to the REST architectural principles [7]). RESTful services are centred around resources, which are interconnected by hyperlinks and grouped into collections, whose retrieval and manipulation is enabled through a fixed set of operations commonly implemented by using HTTP. In contrast to WSDL-based services, Web APIs build upon a light technology stack relying almost entirely on the use of URIs, for both resource identification and interaction, and HTTP for message transmission.

The take up of both kinds of services is, however, hampered by the amount of manual effort required when manipulating them. Research on semantic Web services [8] has focused on providing semantic descriptions of services so that tasks such as the discovery, negotiation, composition and invocation of Web services can have a higher level of automation. These techniques, originally targeted at WSDL services, have highlighted a number of advantages and are currently being adapted towards lighter and more scalable solutions covering Web APIs as well.

⁸ <http://backstage.bbc.co.uk/>

⁹ http://news.bbc.co.uk/sport1/hi/football/world_cup_2010/default.stm

¹⁰ http://www.bbc.co.uk/blogs/bbcinternet/2010/07/bbc_world_cup_2010_dynamic_sem.html

¹¹ <http://www.freebase.com/>

¹² <http://developers.facebook.com/docs/opengraph>

¹³ http://news.cnet.com/8301-13577_3-20003053-36.html

4 Linked Services

The advent of the Web of Data together with the rise of Web 2.0 technologies and social principles constitute, in our opinion, the final necessary ingredients that will ultimately lead to a widespread adoption of services on the Web. The vision toward the next wave of services, first introduced in [9] and depicted in Figure 1, is based on two simple notions:

1. Publishing service annotations within the Web of Data, and
2. Creating services for the Web of Data, i.e., services that process Linked Data and/or generate Linked Data.

We have since then devoted significant effort to refining the vision [10] and implementing diverse aspects of it such as the annotation of services and the publication of services annotations as Linked Data [11, 12], as well as on wrapping, and openly exposing, existing RESTful services as native Linked Data producers dubbed Linked Open Services [13, 14]. It is worth noting in this respect that these approaches and techniques are different means contributing to the same vision and are not to be considered by any means the only possible approaches. What is essential though is exploiting the complementarity of services and the Web of Data through their integration based on the two notions highlighted above.

As can be seen in Figure 2 there are three main layers that we consider. At the bottom are Legacy Services which are services which may be WSDL-based or Web APIs, for which we provide in essence a Linked Data-oriented view over existing functionality exposed as services. Legacy services could in this way be invoked, either



Fig. 2. Services and the Web of Data

by interpreting their semantic annotations (see Section 4.1) or by invoking dedicated wrappers (see Section 4.2) and RDF information could be obtained on demand. In this way, data from legacy systems, state of the art Web 2.0 sites, or sensors, which do not directly conform to Linked Data principles can easily be made available as Linked Data.

In the second layer are Linked Service descriptions. These are annotations describing various aspects of the service which may include: the inputs and outputs, the functionality, and the non-functional properties. Following Linked Data principles these are given HTTP URIs, are described in terms of lightweight RDFS vocabularies, and are interlinked with existing Web vocabularies. Note that we have already made our descriptions available in the Linked Data Cloud through iServe these are described in more detail in Section 4.1.

The final layer in Figure 2 concerns services which are able to consume RDF data (either natively or via lowering mechanisms), carry out the concrete activity they are responsible for, and return the result, if any, in RDF as well. The invoking system could then store the result obtained or continue with the activity it is carrying out using these newly obtained RDF triples combined with additional sources of data. Such an approach, based on the ideas of semantic spaces, has been sketched for the notion of Linked Open Processes [13]. In a sense, this is similar to the notion of service mashups [15] and RDF mash-ups [16] with the important difference that services are, in this case, RDF-aware and their functionality may range from RDF-specific manipulation functionality up to highly complex processing beyond data fusion that might even have real-life side-effects. The use of services as the core abstraction for constructing Linked Data applications is therefore more generally applicable than that of current data integration oriented mashup solutions.

We expand on the second and third layers in Figure 2 in more detail below.

4.1 Implementing Linked Services with Linked Data-Based Annotations

One thread of our work on Linked Services is based on the use of Linked Data-based descriptions of Linked Services allowing them to be published on the Web of Data and using these annotations for better supporting the discovery, composition and invocation of Linked Services.

Our research there is based on the Minimal Service Model (MSM) [17], originally introduced together with hRESTS [18] and WSMO-Lite [19], and slightly modified for the purposes of this work [12]. In a nutshell, MSM is a simple RDF(S) integration ontology which captures the maximum common denominator between existing conceptual models for services. The best-known approaches to annotating services semantically are OWL-S [20], WSMO [21], SAWSDL [22], and WSMO-Lite for WSDL services, and MicroWSMO [23], and SA-REST for Web APIs. To cater for interoperability, MSM represents essentially the intersection of the structural parts of these formalisms. Additionally, as opposed to most semantic Web services research to date, MSM supports both “classical” WSDL Web services, as well as a procedural view on the increasing number of Web APIs and RESTful services, which appear to be preferred on the Web.

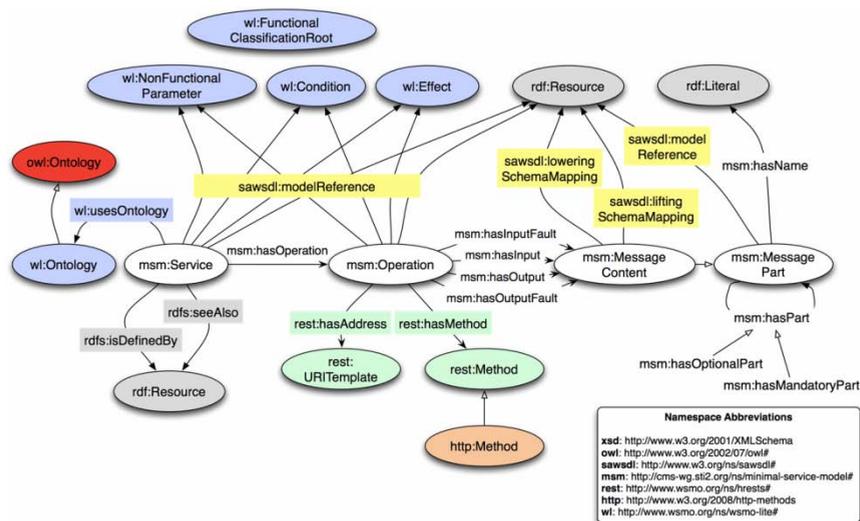


Fig. 3. Conceptual model for services used by iService

As it can be seen in Figure 3, MSM defines Services, which have a number of *Operations*. Operations in turn have input, output and fault *MessageContent* descriptions. *MessageContent* may be composed of mandatory or optional *MessageParts*. The addition of message parts extends the earlier definition of the MSM as described in [18]. The SAWSDL, WSMO-Lite and hRESTS vocabularies, depicted in Figure 3 with the *sawSDL*, *wl*, and *rest* namespaces respectively, complete MSM. SAWSDL supports the annotation of WSDL and XML Schema syntactic service descriptions with semantic concepts, but does not specify a particular representation language nor does it provide any specific vocabulary that users should adopt. WSMO-Lite builds upon SAWSDL by extending it with a model specifying the semantics of the particular service annotations. It provides classes for describing non-functional semantics through the concept of *Nonfunctional Parameter*, and functional semantics via the concepts *Condition*, *Effect*, and *Functional Classification Root*. Finally, hRESTS extends the MSM with specific attributes for operations to model information particular to Web APIs, such as a method to indicate the HTTP method used for the invocation.

The practical use of the MSM for service annotation is supported by two tools, namely SWEET [11] and SOWER. The former is a web-based tool that assists users in the creation of semantic annotations of Web APIs, which are typically described solely through an unstructured HTML Web page. SWEET¹⁴ can open any web page and directly insert annotations following the hRESTS/MicroWSMO microformat. It enables the completion of the following key tasks:

¹⁴ <http://sweet.kmi.open.ac.uk/>

- Identification of service properties within the HTML documentation with the help of hRESTS.
- Integrated ontology search for linking semantic information to service properties.
- Adding of semantic annotations and including lifting and lowering mechanisms that handle format transformations.
- Saving of semantically annotated HTML service description, which can be republished on the Web.
- Extraction of RDF service descriptions based on the annotated HTML.

Similarly, the second tool, SOWER, assists users in the annotation of WSDL services and is based in this case on SAWSDL for adding links to semantic descriptions as well as lifting and lowering mechanisms. During the annotation both tools make use of the Web of Data as background knowledge so as to identify and reuse existing vocabularies. Doing so simplifies the annotation and additionally it also leads to service annotations that are potentially more reusable since they are adapted to existing sources of Linked Data.

The annotation tools are both connected to iServe for one click publication. iServe¹⁵, previously introduced in [12], builds upon lessons learnt from research and development on the Web and on service discovery algorithms to provide a generic semantic service registry able to support advanced discovery over both Web APIs and WSDL services described using heterogeneous formalisms. iServe is, to the best of our knowledge, the first system to publish web service descriptions on the Web of Data, as well as the first to provide advanced discovery over Web APIs comparable to that available for WSDL-based services. Thanks to its simplicity, the MSM captures the essence of services in a way that can support service matchmaking and invocation and still remains largely compatible with the RDF mapping of WSDL, with WSMO-based descriptions of Web services, with OWL-S services, and with services annotated according to WSMO-Lite and MicroWSMO.

The essence of the approach followed by iServe is the use of import mechanisms for a wide range of existing service description formalisms to automatically transform them into the MSM. Once the services are transformed, service descriptions are exposed following the Linked Data principles and a range of advanced service analysis and discovery techniques are provided on top. It is worth noting that as service publication is based on Linked Data principles, application developers can easily discover services able to process or provide certain types of data, and other Web systems can seamlessly provide additional data about service descriptions in an incremental and distributed manner through the use of Linked Data principles. One such example is for instance LUF (Linked User Feedback)¹⁶, which links service descriptions with users ratings, tags and comments about services in a separate server. On the basis of these ratings and comments, service recommendation facilities have also been implemented¹⁷.

¹⁵ <http://iserve.kmi.open.ac.uk/>

¹⁶ <http://soa4all.isoco.net/luf/about/>

¹⁷ <http://technologies.kmi.open.ac.uk/soa4all-studio/consumption-platform/rs4all/>

In summary, the fundamental objective pursued by iServe is to provide a platform able to publish service annotations, support their analysis, and provide advanced functionality on top like service discovery in a way that would allow people and machines to find and exploit service descriptions easily and conveniently. The simple conceptual model explained earlier is a principal building block to support this as a general model able to abstract away the existing conceptual heterogeneity among service description approaches without introducing considerable complexity from a knowledge acquisition and computational perspectives.

SPICES¹⁸ [24] (Semantic Platform for the Interaction and Consumption of Enriched Services) is a platform for the easy consumption of services based on their semantic descriptions. In particular, SPICES supports both the end-user interaction with services and the invocation process itself, via the generation of appropriate user interfaces. Based on the annotations the user is presented with a set of fields, which must be completed to allow the service to execute, and these fields cover input parameters as well as authentication credentials. By using the provided input and the semantic service description stored in iServe, the service can be automatically invoked through SPICES.

Further tooling covering the composition of services as well as analysis of the execution are also being developed as part of an integrated tool suite called SOA4All Studio¹⁹. The SOA4All studio is a fully-fledged system that provides extensive support for completing different tasks along the lifecycle of services, enabling the creation of semantic service description, their discovery, composition, invocation and monitoring.

4.2 Services Which Produce and Consume Linked Data

In this section we consider the relationship between service *interactions* and Linked Data; that is, how Linked Data can facilitate the interaction with a service and how the result can contribute to Linked Data. In other words, this section is not about *annotating* service descriptions by means of ontologies and Linked Data, but about how services should be implemented on top of Linked Data in order to become first class citizens of the quickly growing Linking Open Data Cloud. Note that we take a purist view of the type of services which we consider. These services should take RDF as input and the results should be available as RDF; i.e., service consume Linked Data and service produce Linked Data. Although this could be considered restrictive, one main benefit is that everything is instantaneously available in a machine-readable form. Within existing work on Semantic Web Services, considerable effort is often expended in *lifting* from a syntactic description to a semantic representation and lowering from a semantic entity to a syntactic form. Whereas including this information as annotations requires a particular toolset and platform to interpret them, following Linked Data and

¹⁸ <http://soa4all.isoco.net/spices/about/>

¹⁹ <http://technologies.kmi.open.ac.uk/soa4all-studio/>

REST principles allows for re-exposing the wrappers as RESTful services so that the only required platform to interact with them is the Web (HTTP) itself.

As a general motivation for our case, we consider the status quo of the services offered over the geonames data set, a notable and ‘lifelong’ member of the Linking Open Data Cloud, which are primarily offered using JSON- and XML-encoded messaging. A simple example is given in Table 1, which depicts an excerpt of a weather report gathered from the station at the Airport in Innsbruck, Austria.

Table 1. Geonames JSON Weather Results Example

```
{ "weatherObservation": {
  "stationName": "Innsbruck-Flughafen", "ICAO": "LOWI",
  "countryCode": "AT",
  "lat": 47.266666, "lng": 11.333333, "elevation": 581,
  "clouds": "few clouds", "temperature": "3", ... } }
```

While the JSON format is very convenient for consumption in a browser-based client, it conveys neither the result’s internal semantics nor its interlinkage with existing data sets. The keys, before each colon, are ambiguous strings that must be understood per API; in Linked Data, on the other hand, geonames itself provides a predicate and values for country codes and the WGS84 vocabulary is widely used for latitude and longitude information. Similarly the value “LOWI” corresponds to a feature found within the geonames dataset (and indeed also within the OurAirports and DBpedia Linked Data sets)²⁰ but the string value does not convey this interlinkage.

A solution more in keeping with the Linked Data principles, as seen in our version of these services,²¹ uses the same languages and technologies in the implementation and description of services, communicated as the Linked Open Service (LOS) principles [14] encouraging the following:

- allowing RDF-encoded messages for input/output;
- reusing URIs from Linked Data source for representing features in input and output messages;
- making explicit the semantic relationship between input and output.

In particular with regard to the last point, we can use predicates from existing vocabularies, such as FOAF’s `basedNear`, to represent the relationship between an input point and the nearest weather station. In order to make the statement of this relationship more useful as Linked Data, the approach of Linked Data Services (LIDS) [25] is to URL-encode the input. For instance, the latitude and longitude and used as query parameters so that the point is represented in a URI forming a new

²⁰ The three identifiers for the Innsbruck Airport resource are <http://sws.geonames.org/6299669/>, <http://airports.dataincubator.org/airports/LOWI>, and http://dbpedia.org/resource/Innsbruck_Airport, respectively.

²¹ <http://www.linkedopenservices.org/services/geo/geonames/weather/>

resource identifier. This URI is then used as the subject of such a triple, encoding the relationship to the output.

In aligning LOS and LIDS principles, pursued via a Linked Services Wiki²² and a Linked Data and Services mailing list²³, a URI representing the input is returned using the standard Content-Location HTTP header field. Even in the case of a URL-encoded, LIDS-style input this can be sensible as such a URI will be canonical, whereas a user-encoded input may use variable decimal places for latitude and longitude. LOS has the further advantage that where an input cannot sensibly be URL-encoded, it can first be POSTed as a new resource (Linked Data and Linked Data Services so far concentrate on resource retrieval and therefore primarily the HTTP GET verb), in the standard REST style, and then a resource-oriented service can be offered with respect to it. This can be seen in ontology and query services offered at <http://www.linkedopenservices.org/services>.

LOS and LIDS also coincide on the idea of refining the general principles of Linked Services communicated in Section 4, of describing accepted/expected messages using SPARQL graph patterns. While this is a design decision, it aims at the greatest familiarity and ease for Linked Data developers. It is not without precedent in semantic service description [26]. The authors of [26] use the SPARQL query language to formulate user goals, and to define the pre- and post-conditions of SAWSDL-based service descriptions, which to some degree, at least conceptually, matches the ideas of our approach of using graph patterns for describing inputs (a precondition on the knowledge state prior to service invocation) and outputs (the post-condition of how the knowledge state changes after execution of the service). Although, the use of SPARQL is similar across different proposals, how the patterns are exploited again offers alternative, but complementary views due to LIDS and LOS respectively. On the one hand, atomic user desires can be encoded as a CONSTRUCT query and, under certain restrictions²⁴, query processing techniques can be used to assemble a set of services whose results can be combined to satisfy the initial user request. On the other hand, where more sophisticated control flow is needed, a process (which we call a Linked Open Process [13]) can be manually created and the graph patterns are used for both the discovery of services, and then also reused in defining the dataflow between services within a process, defined again as SPARQL CONSTRUCT queries. Work is on-going on graph pattern-based discovery and process definition and execution.

²² <http://linkedservices.org>

²³ <http://groups.google.com/group/linkedataandservices/>

²⁴ Currently that the graph patterns contained in this request, and in the service descriptions, are conjunctive – meaning do not use OPTIONAL or UNION, etc. – and free of FILTERs. etc. [4]

5 Conclusions

In this paper we have outlined how Linked Data provides a mechanism for describing services in a machine readable fashion and enables service descriptions to be seamlessly connected to other Linked Data. We have also described a set of principles for how services should consume and produce Linked Data in order to become first-class Linked Data citizens.

From our work thus far, we see that integrating services with the Web of Data, as depicted before, will give birth to a services ecosystem on top of Linked Data, whereby developers will be able to collaboratively and incrementally construct complex systems exploiting the Web of Data by reusing the results of others. The systematic development of complex applications over Linked Data in a sustainable, efficient, and robust manner shall only be achieved through reuse. We believe that our approach is a particularly suitable abstraction to carry this out at Web scale.

We also believe that Linked Data principles and our extensions can be generalized to the Internet of Services. That is, to scenarios where services sit within a generic Internet platform rather than on the Web. These principles are:

Global unique naming and addressing scheme - services and resources consumed and produced by services should be subject to a global unique naming and addressing scheme. This addressing scheme should be easily resolvable such that software clients are able to access easily underlying descriptions.

Linking – linking between descriptions should be supported to facilitate the reuse of descriptions and to be able to specify relationships.

Service abstraction – building from SOA principles functionality should be encapsulated within services which should have a distinct endpoint available on the Internet, through which they can be invoked using standard protocols.

Machine processability – the descriptions of the services and resources should be machine-processable. RDF(S) achieves this by having an underlying semantics and also with the ability to point to an ontology based description of the schema used. Ideally, the inputs and outputs for services should be machine-processable as well.

Following from the above we believe that the Future Internet will benefit greatly from a coherent approach which integrates service orientation with the principles underlying Linked Data. We are also hopeful that our approach provides a viable starting point for this. More generally, we expect to see lightweight semantics appearing throughout the new global communications platform which is emerging through the Future Internet work and also note that proposals already exist for integrating Linked Data at the network level²⁵.

Acknowledgements. This work was partly funded by the EU project SOA4All (FP7-215219)²⁶. The authors would like to thank the members of the SOA4All project and the members of the STI Conceptual Models for Services Working Group for their interesting feedback on this work.

²⁵ <http://socialmedia.net/node/175>

²⁶ <http://www.soa4all.eu/>

Open Access. This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284(5), 34–43 (2001)
2. Brickley, D., Guha, D., R.V. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation (February 2004), <http://www.w3.org/TR/rdf-schema/>
3. A.R., Wilde, E.: Linking Data from RESTful Services. In: Workshop on Linked Data on the Web at WWW 2010 (2010)
4. Speiser, S., Harth, A.: Taking the LIDS off Data Silos. In: 6th International Conference on Semantic Systems (I-SEMANTICS) (October 2010)
5. Cardoso, J., Barros, A., May, N., Kylau, U.: Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments. In: IEEE Int'l Conference on Services Computing, July 2010, pp. 602–609 (2010)
6. Berners-Lee, T.: Linked Data - Design Issues (July 2006), <http://www.w3.org/DesignIssues/LinkedData.html>
7. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD Thesis, University of California (2000)
8. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic Web Services. *IEEE Intelligent Systems* 16(2), 46–53 (2001)
9. Pedrinaci, C., Domingue, J., Krummenacher, R.: Services and the Web of Data: An Unexploited Symbiosis. In: AAAI Spring Symposium “Linked Data Meets Artificial Intelligence”, March 2010, AAAI Press, Menlo Park (2010)
10. 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)
11. Maleshkova, M., Pedrinaci, C., Domingue, J.: Supporting the creation of semantic RESTful service descriptions. In: Workshop: Service Matchmaking and Resource Retrieval in the Semantic Web at ISWC (November 2009)
12. Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J., Domingue, J.: iServe: a Linked Services Publishing Platform. In: Workshop: Ontology Repositories and Editors for the Semantic Web at ESWC (June 2010)
13. Krummenacher, R., Norton, B., Marte, A.: Towards Linked Open Services and Processes. In: Future Internet Symposium, October 2010, pp. 68–77.
14. Norton, B., Krummenacher, R.: Consuming Dynamic Linked Data. In: 1st International Workshop on Consuming Linked Data (November 2010)
15. Benslimane, D., Dustdar, S., Sheth, A.: Services Mashups: The New Generation of Web Applications. *IEEE Internet Computing* 12(5), 13–15 (2008)
16. Phuoc, D.L., Polleres, A., Hauswirth, M., Tummarello, G., Morbidoni, C.: Rapid Prototyping of Semantic Mash-ups Through Semantic Web Pipes. In: 18th Int'l Conference on World Wide Web, April 2009, pp. 581–590 (2009)
17. Maleshkova, M., Kopecky, J., Pedrinaci, C.: Adapting SAWSDL for Semantic Annotations of RESTful Services. In: Workshop: Beyond SAWSDL at OTM, November 2009, pp. 917–926 (2009)

18. Kopecky, J., Gomadam, K., Vitvar, T.: hRESTS: An HTML Microformat for Describing RESTful Web Services. In: IEEE/WIC/ACM Int'l Conference on Web Intelligence and Intelligent Agent Technology, December 2008, pp. 619–625 (2008)
19. Vitvar, T., Kopecký, 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, pp. 674–689. Springer, Heidelberg (2008)
20. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T.R., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services. Technical Report, Member Submission, W3C (2004)
21. 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 (2006)
22. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema. W3C Recommendation (August 2007), <http://www.w3.org/TR/sawSDL/>
23. Maleshkova, M., Kopecky, J., Pedrinaci, C.: Adapting SAWSDL for Semantic Annotations of RESTful Services. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 917–926. Springer, Heidelberg (2009)
24. Alvaro, G., Martinez, I., Gomez, J.M., Lecue, F., Pedrinaci, C., Villa, M., Di Matteo, G.: Using SPICES for a Better Service Consumption. In: Extended Semantic Web Conference (Posters (June 2010)
25. Speiser, S., Harth, A.: Towards Linked Data Services. In: Int'l Semantic Web Conference (Posters and Demonstrations (November 2010)
26. Iqbal, K., Sbodio, M.L., Peristeras, V., Giuliani, G.: Semantic Service Discovery using SAWSDL and SPARQL. In: 4th Int'l Conference on Semantics, Knowledge and Grid, December 2008, pp. 205–212 (2008)