Toward the Next Wave of Services: Linked Services for the Web of Data

Carlos Pedrinaci
(Knowledge Media Institute, The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
c.pedrinaci@open.ac.uk)

John Domingue
(Knowledge Media Institute, The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
j.b.domingue@open.ac.uk)

Abstract: It has often been argued that Web services would have a tremendous impact on the Web, as a core enabling technology supporting a highly efficient service-based economy at a global scale. However, despite the outstanding progress in the area we are still to witness the application of Web services in any significant numbers on the Web. In this paper, we analyse the state of the art highlighting the main reasons we believe have hampered their uptake. Based on this analysis, we further discuss about current trends and development within other fields such as the Semantic Web and Web 2.0 and argue that the recent evolution provides the missing ingredients that will lead to a new wave of services – Linked Services – that will ultimately witness a significant uptake on a Web scale. Throughout the presentation of this vision we outline the main principles that shall be underpinning the development of Linked Services and we illustrate how they can be implemented using a number of technologies and tools we have developed and are in the process of extending.

Key Words: Linked Services, Web Services, Semantic Web, Semantic Web Services, Web of Data

Category: D2.11, D2.13, H.3.5, I2.1, I2.5, I2.11, K4.4, K6.3

1 Introduction

Web Services and the Service-Oriented Architecture are commonly lauded as a silver bullet for Enterprise Application Integration, implementation of inter-organizational business processes, and even as a general solution for the development of all complex distributed applications. Despite the appealing characteristics of service-orientation principles and technologies, their uptake on a Web-scale has been significantly less prominent than initially anticipated [14]. First and foremost Web services, despite their name, are hardly a Web-oriented technology [58] but rather one suited for enterprises which so far have been reluctant to publish functionality on the Web. Secondly, from a technical perspective, current technologies are such that software developers need to devote a significant effort to discovering sets of suitable services, interpreting them, developing software that overcomes their inherent data and process mismatches, and finally combining them into a complex composite process.

Semantic Web Services (SWS) [35] have long tried to overcome Web services limitations by enriching them with semantic annotations in order to better support their discovery, composition, and execution. Up until now, however, the impact of SWS on the Web has been minimal. In the Web, semantics are used to mark up a wide variety of data-centric resources but are not used to annotate online functionality in any form in significant numbers. In fact, although SWS technologies have already shown their benefits, e.g., in discovery [45], research in the area has failed to take into account the socio-economic aspects devoted to the creation and annotation of services. First, research has mostly focused on devising highly expressive conceptual models and has given birth to a number of diverging and largely incompatible solutions. These efforts have essentially glossed over the complexity they introduce, the additional effort demanded of users, and they have brought additional heterogeneity to an already overwhelming stack of specifications. Second, SWS research has for the most part targeted WSDL/SOAP based Web services which are not prevalent on the Web [14]. As a consequence, SWS is instead a niche technology only accessible to highly trained experts and the benefits obtained are most often not considered worth the additional investment.

In parallel, the Web is currently witnessing a dramatic change with the advent of Web 2.0 [38] and Linked Data technologies [8]. The former is "socialising" the Web, putting individuals at the core of the Web as both data producers and consumers. Web 2.0 technologies have shown that collaboration over the Web can produce outstanding results with a low cost, and it is also encouraging enterprises and institutions to offer their data and services publicly at a previously unprecedented scale and pace [25, 12, 14]. Second, Linked Data technologies, which derive from research on the Semantic Web, have given birth to the Web of Data, "a Web of things in the world, described by data on the Web" [8]. The Web of Data, impelled by the current trend towards an open Web, has recently experimented an outstanding growth and currently provides publicly large amounts of interconnected data concerning a wide range of domains and described in terms of light weight ontologies for supporting automated processing [8].

In this paper we explore the relationship between services and the Web of Data. We identify the potential benefits that can be obtained by adequately integrating these so far rather disconnected worlds. We anticipate that this integration will mitigate the existing limitations of both services and the Web of Data, giving birth to a new wave of services dubbed Linked Services, that will ultimately lead to an explosion in the publication and use of services on the Web. We outline how this integration could take place by using simpler vocabularies for describing services and through the adoption of Linked Data principles for publishing services on the Web. Finally, we outline how Linked Services will be able to provide the additional necessary building blocks for appropriately exploiting the wealth of information exposed in the Web of Data.

The remainder of this paper is organised as follows. First, we present the technological background around services and the Web. We then discuss why, in our opinion, the current situation can give birth to a new wave of services. We then present how the use of light weight semantics can allow us to bring services into the Web enabling their discovery through state of the art Linked Data technologies. Next, we focus on how services can contribute to the Web of Data both generating new data and processing existing one. Finally, we conclude the paper and outline key topics for further research.

2 Background and Related Work

The current technological landscape is characterised by a number of highly complementary technologies that have so far remained disconnected. In this section we review existing work in the area of Web Services, Web 2.0, Semantic Web, the Web of Data, and Semantic Web Services presenting the main results achieved so far and highlighting the main trends, challenges, and opportunities.

2.1 Web Services

Traditionally the idea of deploying and providing services on the Web has been tightly bound to Web service technologies. Web services are software systems offered over the Internet via platform and programming-language independent interfaces defined on the basis of a set of open standards such as XML, SOAP, and WSDL [16]. The fundamental advantage of this technology lies in the support it brings to developing highly complex distributed systems maximising the reuse of loosely coupled components. Several languages for Web service composition have been proposed over the years in order to combine services in a process-oriented way, among which the most prominent is BPEL4WS [2]. Additionally, the stack of technologies is completed by a large and rather overwhelming number of specifications dubbed WS-*, which deal with aspects such as security, transactions, messaging, and notification [16]. This stack has brought a considerable level of complexity and yet suffers from the fact that descriptions are purely syntactic. As a consequence discovering, composing, and mediating Web services remains a predominantly manual task.

A fundamental tenet of Service-Oriented Architectures is the notion of service registries for programmatic access and discovery of suitable services. Service publication has therefore been at the core of research and development in this area since the very beginning. The Universal Business Registry part of Universal

Description Discovery and Integration (UDDI) [23] is perhaps the most well-known effort towards supporting the publication of services on the Web. On the basis of UDDI, large companies like SAP, IBM and Microsoft created a universal registry for enterprise services that could be accessed publicly but it did not gain enough adoption and it was discontinued in 2006 after five years of use.

One of the main reasons for the lack of success of UDDI was the fact that, although these registries are relatively complex, they do not support expressive queries [45]. Another fundamental reason is the fact that, as we saw earlier, the work around services has essentially focussed on enterprises which have thus far been reluctant to publish their services on the Web. Today, `Seekda.com` provides one of the largest indexes of publicly available Web services which currently accounts for 28,500 Web services with their corresponding documentation. The number of services publicly available contrasts significantly with the billions of Web pages available, and interestingly is not significantly bigger than the 4,000 services estimated to be deployed internally within Verizon [56]. Other academic efforts in crawling and indexing Web services on the Web have found far lower numbers of services [1].

2.2 Web 2.0

The term Web 2.0, commonly attributed to OReilly [38], was first defined on the basis of the technologies used, e.g., AJAX. More recently, however, it is increasingly used to account for the central role users play within these applications [25, 12]. Most successful Web 2.0 web sites are largely based on exploiting user-provided content and on the elicitation and use of the social networks created among them. For instance, Wikipedia and Flickr are largely based on content provided by their users in a rather altruistic manner. This new way of providing content is based on dropping the unnecessarily limiting distinction between providers and consumers, giving birth instead to what is often referred to as *prosumers*. Additionally, and thanks to the close integration of prosumers in the provisioning process, networks among users are elicited and exploited by sites such as Last.fm or Amazon to provide highly accurate recommendations.

Impelled by the Web 2.0 phenomenon, the world around services on the Web, thus far limited to “classical” Web services based on SOAP and WSDL, has significantly evolved with the proliferation of Web APIs, also called RESTful services [47] when they conform to the REST architectural style [20]. This newer kind of services is characterised by the simplicity of the technology stack they build upon, i.e., URIs, HTTP, XML and JSON, and their natural suitability for the Web. Nowadays, an increasingly large quantity of Web sites offer (controlled) access to part of the data they hold through simple Web APIs, see for instance Flickr[1], Last.fm[2], and Facebook[3]. This trend towards opening access to previously closed data silos has generated a new wave of Web applications, called *mashups*, which obtain data from diverse Web sites and combine it to create novel solutions [5].

`ProgrammableWeb.com`, the most popular directory of Web APIs lists at the time of this writing lists 2,000 APIs and 4,800 mashups. This directory is based on the manual submission of APIs by users and currently provides simple search mechanisms based on keywords, tags, or a simple classification, none of which are particularly expressive. In fact, Web APIs are generally described using plain, unstructured HTML, except for a few that use the XML-based format WADL [22]. As a consequence, and despite their popularity, discovering Web APIs or developing mashups that integrate disparate services in this manner suffers from a number of limitations similar to those we previously outlined for "classical" Web services, with an increased complexity since most often no machine-processable description is available. Discovering services, handling heterogeneous data, and creating service compositions are largely manual, tedious tasks which result in the development of custom tailored solutions on a case by case basis.

2.3 The Semantic Web and the Web of Data

The Semantic Web [7] is an extension of the current human-readable Web, adding formal knowledge representation so that intelligent software can reason with the information in an automatic and flexible way. Semantic Web research has therefore largely focussed on defining languages and tools for representing knowledge in a way that can be shared, reused, combined, and processed over the Web. This research has led to a plethora of standards such as RDF(S) [9], OWL [41], as well as corresponding tools such as ontology editors [37], RDF(S) storage and querying systems [11] and reasoners [21], to name a few.

The Web of Data is a relatively recent effort derived from research on the Semantic Web, whose main objective is to generate a Web exposing and interlinking data previously enclosed within silos. The Web of Data is based upon four simple principles, known as the Linked Data principles, which essentially dictate that every piece of data should be given an HTTP URI which, when looked up, should offer useful information using standards like RDF and SPARQL [8]. Additionally, data should be linked to other relevant resources therefore allowing humans and computers to discover additional information.

Since the Linked Data principles were outlined in 2006, there has been a large uptake most notably by the Linking Open Data project[4] through DBpedia [3]

[1] See `http://www.flickr.com/services/api/`
[2] See `http://www.last.fm/api`
[3] See `http://developers.facebook.com/docs/`
[4] See http://linkeddata.org/

and ulterior additions of data about reviews [24], scientific information and geographical information, to name a few. Large companies like the BBC and governments from countries like the United Kingdom or the United States of America have also joined this initiative and are gradually releasing large amounts of data they have.

This outstanding growth of the Web of Data is urging researchers to devise means to exploit the valuable information it exposes. Among the main applications produced so far there are a number of data browsers that help people navigate through the data like Disco and Tabulator [6]. There are systems that crawl, index and provide intelligent search support over the Web of data like Sindice [39] and Watson [13]. And finally, there are a few domain-specific applications such as `Revyu.com` [24] or DBPedia Mobile [4] that provide domain-specific functionality by gathering and mashing up data. Although useful these applications hardly go beyond presenting together data gathered from different sources leaving the great potential of this massive data space unexploited. It is therefore becoming of crucial importance to devise ways in which smart applications that exploit the Web of Data could be systematically developed.

2.4 Semantic Web Services

Semantic Web services were initially proposed in order to pursue the vision of the semantic Web presented in [7] whereby intelligent agents would be able to exploit semantic descriptions in order to carry out complex tasks on behalf of humans. Early on, however, the research efforts focussed on combining Web services and semantic Web technologies so that tasks such as the discovery, negotiation, composition and invocation of services could have a higher level of automation.

The landscape of semantic Web services is characterized by a number of conceptual models that, despite a few common characteristics, remain essentially incompatible due to the different representation languages and expressivity utilized as well as because of conceptual differences. Major frameworks include WSMO [19], OWL-S [34], SAWSDL [18], and WSMO-Lite [59]. WSMO and OWL-S adopt a top-down view over services, covering the data models, behavioural aspects, nonfunctional properties, and supporting the definition of processes. The means for describing these are significantly different, though. In contrast, SAWSDL adopts a bottom-up approach and simply provides hooks for linking to particular ontologies and transformation definitions. In practice, the heterogeneity of the existing approaches has prevented their integration, leading to a significant fragmentation in the field and thus harming the adoption of SWS.

On the basis of the aforementioned conceptual models, many researchers have worked on enhancing service registries using semantic technologies, see for instance [29, 55], many of which have built upon UDDI. Despite demonstrating the advantages of semantic annotations in discovering services, particularly in terms of accuracy and in dealing with heterogeneous data models, SWS work has downplayed the additional complexity involved in creating semantic annotations for services. Consequently, the Web does not contain a significant body of service annotations: the largest public repository today is probably OPOSSum [31] which includes a test collection with approximately 2500 service annotations and provides programmatic access to its content solely through direct access to the database management system [31].

Regardless of the differences at the semantic level, the vast majority of the SWS initiatives are predicated upon the semantic enrichment of WSDL Web services and, as we saw earlier, these have turned out not to be prevalent on the Web. The Web services ecology has recently seen a major evolution with the advent and proliferation of Web APIs and RESTful services [47], and there has not been much progress on, or even concern with, means for providing structured descriptions and discovering these newer kinds of services. Only recently have researchers started focusing on Web APIs and RESTful services, the main examples being SA-REST [53] and hRESTS/MicroWSMO [30, 32].

3 Services and the Web of Data: An Unexploited Symbiosis

The advent of Web services and related technologies was quickly followed by considerable hype and grandiose expectations with respect to the impact Web services would have for enterprises and the economy in general. It was often assumed that Web services would ultimately lead to the creation of a service-based economy over the Web. However, Web services are nowadays mostly used within controlled environments such as large enterprises rather than on the Web. One could argue that a reason for this lack of take up is the fact that Web services, despite their name, were not really thought for the Web [58]. In fact, the considerable complexity of the WS-* stack did hamper their adoption on the Web as recent practice, based instead on the use of simpler approaches such as Web APIs, shows. Another reason is however the fact that Web services have essentially targeted enterprises, which tend not to publicly publish Web services in any significant numbers.

Research on SWS has managed to alleviate some of the technical drawbacks of existing Web services technologies. Despite the advanced results obtained, none of the approaches devised thus far have gained widespread adoption for three main reasons. First and foremost, all SWS approaches have built upon Web services technologies that are not prevalent on the Web. Secondly, SWS add complex logics to an already complex WS-* stack. SWS require complex architectures, highly advanced reasoning machinery, and rich semantic annotations that, up until now, had to be provided mostly from scratch by highly trained IT staff. Finally, the existing dichotomy between the syntactic level and the semantic level requires devoting significant effort to providing transformation mechanisms between semantic and syntactic representations of information which add further need for manual labour and are highly sensitive to minor variations on data representation.

We believe that the advent of the Web of Data together with the rise of Web 2.0 technologies and social principles constitute the final necessary ingredients that will ultimately lead to a widespread adoption of services on the Web. In the remainder of this paper we shall refer to this new kind of services as Linked Services. The main reasons for this are the existing technical symbiosis between services, semantics, and the Web of Data [42], as well as the rise of the prosumer and the global movement towards an open Web driven by the current unprecedented sharing of data and functionality openly on the Web.

On the one hand, from a technological perspective, the evolution of the Web of Data is highlighting the fact that light weight semantics yield significant benefits that justify the investment in annotating data and deploying the necessary machinery. This initiative is contributing to generate an outstanding body of knowledge (light weight ontologies and data expressed in their terms) that can help to significantly reduce the effort for creating semantic annotations for services. Furthermore, it also represents a significant use case for the application of services technologies on a Web scale in order to process this wealth of data which remains nowadays largely unexploited. On the other hand, from a socio-economic perspective, the recent evolution around Web 2.0 has shown that collaboration over the Web can lead to large quantities of very useful data with a low cost. Similarly, both Web 2.0 and more recently Linked Data technologies are encouraging enterprises and institutions to offer their data and services publicly at a previously unprecedented scale and pace. This new scenario provides in our view suitable technologies and data, as well as the necessary economic and social interest for the wide application of services technologies on a Web scale.

3.1 Linked Services

The vision toward the next wave of services – Linked Services – presented herein is based on two simple ideas: publishing service annotations in the Web of Data, and creating services for the Web of Data, i.e., services that process Linked Data and generate Linked Data. In a nutshell, Linked Services are services described as Linked Data. Therefore, these are service descriptions whereby their inputs and outputs, their functionality, and their non-functional properties are described in terms of (reused) light weight RDFS vocabularies and exposed following Linked Data principles. In fact, as such, Linked Services descriptions represent highly valuable information which is still to be provided in the Web of Data: data about reusable functionality on the Web. Secondly, by virtue of these descriptions, Linked Services are therefore services that, with appropriate infrastructure support, can consume RDF from the Web of Data, and, if necessary, can also generate additional RDF to be fed back to the Web of Data. In other words, Linked Services constitute a processing layer on top of the wealth of information currently available in the Web of Data which remains unexploited.

In the remainder of this paper we shall describe in more detail how this new wave can be supported and promoted technically, we explain which are the essential principles one needs to build upon and, where appropriate, we shall illustrate how our current research is taking us in this direction. Although in this section we present concrete technologies, the reader should note that the vision presented herein could perfectly be achieved by other means. The essential aspects are, however, the publication of service descriptions in the Web of Data for their discovery and reuse, and the provisioning of processing functionality on top Linked Data.

4 Services on the Web of Data

We previously called attention to the scarcity of publicly available Web services. We highlighted the lack of success of prior service registries on the Web as one of the reasons behind this, and highlighted several aspects that have hampered the adoption of UDDI as a suitable standard for service registries. We also pointed out the fragmentation currently affecting SWS research as well as the proliferation of Web APIs as a simpler and increasingly more popular alternative over "traditional" Web services.

Before any significant uptake of services can take place on the Web, proper mechanisms for creating, publishing and discovering services must be in place. In this respect, our previous review of the state of the art shows that:

- Semantics are essential to reach a sufficient level of automation during the life-cycle of services,
- finding an adequate trade-off between the expressivity of the service model used and the scalability from a computational and knowledge acquisition perspective is key for a wide adoption of service technologies,
- the annotation of services should be simplified as much as possible, and "crowdsourcing" appears to be a particularly effective and cheap solution to this end,
- on the Web, light weight ontologies together with the possibility to provide custom extensions prevail against more complex models,

- any solution to deploying services that aspires to be widely adopted should build upon the various approaches and standards used on the Web, including Web APIs, RDF, and SPARQL,
- Linked Data principles [8] represent nowadays the best practice for publishing data on the Web both for human and machine consumption,
- links between publicly available datasets are essential for the scalability and the value of the data exposed.

The principles we have just highlighted have an impact in a wide range of activities during the life-cycle of services. Notably, in the remainder of this section we shall tackle how Web services and Web APIs can be annotated, we shall describe how we can better support the annotation of services and finally we described how we are currently supporting the homogeneous publication and discovery of Web services and Web APIs on the Web using light weight semantics.

4.1 Annotation of WSDL Services with WSMO-Lite

W3C produced in 2007 the Semantic Annotations for WSDL and XML Schema specification [18], a minimal bottom-up approach to annotating services semantically which has gained further uptake than more ambitious solutions like OWL-S and WSMO. SAWSDL provides simple hooks for pointing to semantic descriptions from WSDL and XML elements. In particular, it supports three kinds of annotations, namely *model reference*, *lifting schema mapping* and *lowering schema mapping* which allow pointing to semantic elements described elsewhere on the Web, or to specifications of data transformations from a syntactic representation to the semantic counterpart and back respectively. SAWSDL does not advocate for a particular representation language for these documents nor does it provide any specific vocabulary that users should adopt.

WSMO-Lite [59] builds upon SAWSDL overcoming some of its limitations while remaining light weight. In a nutshell, WSMO-Lite provides a very simple RDFS ontology together with a methodology for expressing functional and nonfunctional semantics, and an information model for WSDL services based on SAWSDLs model reference hooks. WSMO-Lite makes explicit the intended meaning for model reference annotations without modifying SAWSDL but rather informing users on how they should structure the models their annotations point to.

The WSMO-Lite ontology includes means for specifying the functionality of a service with respect to a hierarchy of functional categories (e.g., eCl@ss [26]) through the notion of *Functional Classification Root*. Additionally, it provides hooks for more advanced definition of non-functional properties as well as *Conditions* and *Effects*. The ontology is entirely expressed in RDF(S) and where the expressivity of RDFS is not sufficient (notably for expressing conditions and effects) other languages such as WSML [19] and those produced by the W3C Rule Interchange Format Working Group[5] can be used.

4.2 Annotation of Web APIs with MicroWSMO

As we previously introduced, Web APIs and RESTful services are increasingly used on the Web. Therefore any approach to using services on the Web that would disregard them would be unnecessarily limiting. Annotating this kind of service does, however, bring additional complexity given that in most of the cases services are solely described through unstructured HTML pages aimed at humans.

MicroWSMO [30, 32] is a microformat-like[6] notation that forms the basis for our work on semantically describing Web APIs. MicroWSMO builds upon the hRESTS (HTML for RESTful services) microformat. hRESTS enables the creation of machine-processable Web API descriptions based on available HTML documentation [30]. As a microformat hRESTS provides a number of HTML classes that allow one to structure APIs descriptions by identifying services, operations, methods, inputs, outputs, and addresses. It therefore supports, by simple injections of HTML code within Web pages, to turn unstructured HTML-based descriptions of Web APIs into structured services descriptions similar to those provided by WSDL.

With the hRESTS structure in place, HTML service descriptions can be annotated further by including pointers to the semantics of the service, operations, and data manipulated. To this end MicroWSMO extends hRESTS with three additional properties, namely *model*, *lifting* and *lowering* that are taken from SAWSDL and have the same semantics. MicroWSMO also adopts WSMO-Lite as the reference ontology for annotating RESTful services semantically.

4.3 Supporting Services Annotation

Arguably, one of the main limitations of previous approaches to integrating services in the Semantic Web, has been the difficulty from an annotation perspective. SWS approaches like WSMO and OWL-S, mostly focussed on devising highly expressive frameworks able to capture formally the semantics of services in a considerable detail, overlooked the bottleneck they were introducing with respect to the annotation of services. Indeed, the creation of SWS based on these frameworks requires a significant manual labour devoted to devising domain models, taxonomies, orchestrations, and other rules that can only be created at a slow pace by highly trained IT personnel.

[5] See http://www.w3.org/2005/rules/wiki/RIF_Working_Group
[6] See http://www.microformats.org

Some effort has been devoted by previous research toward the automation of service annotation, notably [27] and [48]. However, although useful, the support provided still needs to be complemented with substantial manual editing, the creation of ontologies and rules. The use of light weight ontologies as opposed to highly expressive conceptual models reduces considerably the effort involved and the amount of annotations to be provided. Additionally, and more importantly, the Web of Data is significantly changing the environment from an annotation and usage point of view.

On the one hand, the wide range of ontologies and semantic data publicly available on the Web is an increasingly valuable source of knowledge. The Web of Data can be used as background knowledge [13] in order to provide suitable ontologies that can be used, extended, and combined to create domain ontologies for annotating services in an easier manner as highlighted in [33]. Furthermore, the existence of increasingly large quantities of information expressed in terms of ontologies can effectively be exploited to support the identification of the domain of a service based for instance on its documentation as well as it can, for instance, support the matching of ontologies when creating new domain models or when integrating different services [49].

On the other hand, generating service annotations by reusing existing ontologies directly contributes to increasing services usability and presumably their uptake. For instance services may be classified with respect to well-known service classifications such as the previously mentioned eCl@ss ontology, better supporting their discovery by software and humans aware of that particular ontology. Furthermore, annotating services inputs and outputs with respect to existing vocabularies ensures the direct applicability of services over data already available as well as it allows Linked Data application developers to carry out data driven discovery of services by simply checking the input and output types of services. From a more abstract perspective, this process ensures that services modeled in this way are linked to the Web of Data as encouraged by Linked Data principles.

Finally, Web 2.0 applications have highlighted the advantages that the social side of the Web can bring when a significant body of users and data has been gathered [25]. The same way we can exploit the growing body of knowledge generated by the Semantic Web, we expect that as the number of service annotations grow, we would also be able to exploit them in order to contribute to the overall annotation process by i) ranking the domain models with respect to their popularity thus indirectly contributing to increasing services compatibility; and ii) by refining the identification of the domain of a service based on prior decisions by other users.

We are currently devoting significant efforts to creating tools that support users in the annotation of services based on the principles introduced above. One such application is SWEET [33] which is, to the best of our knowledge, the first tool that enables the creation of semantic annotations for Web APIs and RESTful services. SWEET provides user support for creating hRESTS/MicroWSMO annotations over any HTML page describing Web APIs, therefore supporting a non-intrusive incremental annotation of existing resources. The tool, assisted by Watson [13], supports users in browsing the Semantic Web while annotating services so that they can identify suitable vocabularies such as FOAF [10], and use them for the annotation. A tool called SOWER, based on the same principles but focussing on the annotation of WSDL services, has also been developed. Currently, the social aspects are not exploited by these tools since it is first necessary to gather a significant body of service annotations.

4.4 Homogeneous Publication and Discovery of Services on the Web of Data

Syntactic and semantic descriptions of Web services aim at providing information about services in a way that can automatically be processed by machines. However, at present, these descriptions can only be retrieved through the Web of documents, which is essentially designed for human beings, or through specific interfaces to registries such as UDDI that have failed to gain significant uptake.

A fundamental step for bringing services closer to the Web is their publication based on current best practices. We view service annotations as a particular kind of highly valuable data: data that informs us about existing reusable functionality exposed somewhere on the Web that processes and/or generates data. As such, services should therefore be published on the Web according to current best practices for publishing data – the Linked Data principles – so that applications can easily discover and process their descriptions on the basis of the very same technologies they use for retrieving data.

In order to explore and validate these principles we have developed iServe [43], a public platform that unifies service publication and discovery on the Web through the use of light weight semantics. iServe 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 different kinds of services described using heterogeneous formalisms. The registry is, to the best of our knowledge, the first system able to homogeneously publish and provide advanced discovery support for SWS expressed in several formalisms. It is also the first one to provide advanced discovery over Web APIs and Web services homogeneously.

In the remainder of this section we first outline the conceptual model iServe builds upon and we then present the overall approach implemented by the platform in order to support the homogeneous publication and discovery of services.

4.4.1 Minimal Service Model

In order to publish services on the Web of Data it is necessary to provide a common vocabulary based on existing Web standards able to describe services in a way that allows machines to automatically locate and filter services according to their functionality or the data they handle, and to appropriately support their automated invocation. Additionally, as opposed to most SWS research to date, it is of utmost importance to support the annotation of both "classical" WSDL Web services, as well as the increasing number of Web APIs and RESTful services which appear to be preferred on the Web.

To this end our research is based on the Minimal Service Model (MSM), originally introduced together with hRESTS [30] and WSMO-Lite [59], and slightly modified for the purposes of this work. The MSM, driven by Semantic Web best practices, builds upon existing vocabularies, namely SAWSDL, WSMO-Lite and hRESTS, depicted in Figure 1 with the `sawsdl`, `wl`, and `rest` namespaces respectively. In a nutshell, the MSM is a simple RDF(S) integration ontology based on the principle of minimal ontological commitment; it captures the maximum common denominator between existing conceptual models for services. Thus, the MSM does not aim to be yet another service model to bring further heterogeneity to the SWS landscape; it is instead an integration model at the intersection of existing formalisms, able to capture the core semantics of both Web services and Web APIs in a common model, homogeneously supporting publication and discovery. Still, the MSM is devised in a way such that framework-specific extensions can remain attached, to the benefit of clients able to comprehend and exploit those formalisms.

The MSM, denoted by the `msm` namespace in Figure 1, defines `Service`s which have a number of `Operation`s. `Operation`s in turn have input, output and fault `MessageContent` descriptions. `MessageContent` may be composed of mandatory or optional `MessageParts`[7]. The intent of the message part mechanism is to support finer-grained input/output discovery, as available in SAWSDL, OWL-S and WSMO, especially including support for optional parts.

SWS frameworks thus far have provided support for semantically describing different subsets of the following aspects of services [52, 59]:

- *Functional semantics* defines service functionality, that is, the function a service offers to its clients when it is invoked. This information is of particular relevance when finding services and when composing them.

- *Nonfunctional semantics* defines any specific details concerning the implementation or running environment of a service, such as its price or quality of service. Nonfunctional semantics provide additional information about services that can help rank and select the most appropriate one.

[7] The addition of message parts is a small extension to the original MSM.

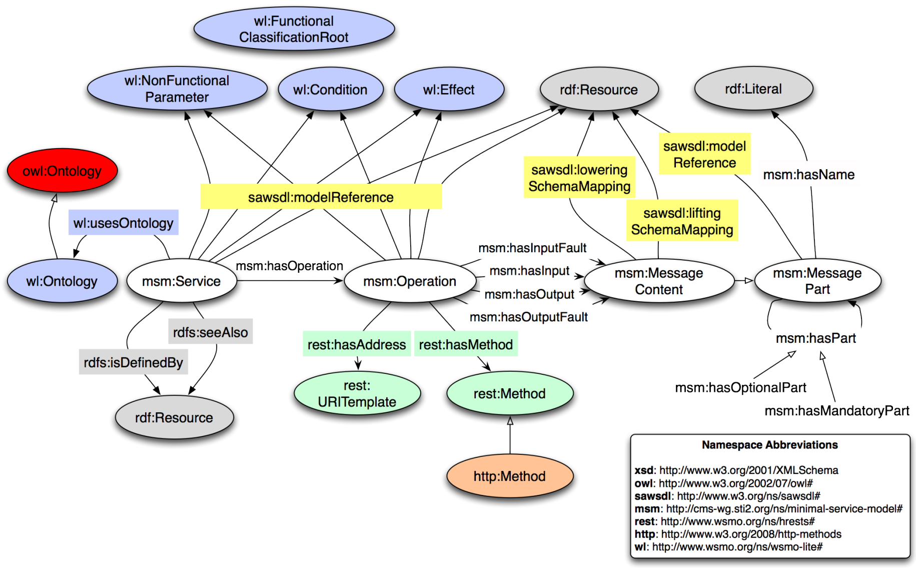

Figure 1: Minimal Service Model.

– *Behavioural semantics* specifies the protocol (i.e., ordering of operations) that a client needs to follow when invoking a service.

– *Information model* defines the semantics of input, output, and fault messages.

To attach these semantics to the service model, we adopt the RDF mapping of SAWSDL introduces earlier, which defines three kinds of annotations over WSDL and XML Schema, namely *model reference*, *lifting schema mapping*, and *lowering schema mapping*. The schema mapping annotations provide grounding from the service's Information Model to the concrete on-the-wire messages, whereas the model references can be used for pointing to ontologies covering functional semantics, nonfunctional semantics , behavioural semantics and the information model.

The WSMO-Lite vocabulary [59] completes the MSM by providing classes for describing the above four aspects of service semantics and by supplying type information to the generic model references. In particular, WSMO-Lite captures nonfunctional semantics through the concept of `Nonfunctional Parameter`, and functional semantics via the concepts `Condition`, `Effect`, and `Functional Classification Root`. The reader may note that WSMO does not have direct support for functional classifications; still, the majority of discovery engines for

WSMO have indirectly applied the notion of classifications through hierarchies of `Web Services` or `Goals` (e.g. in [56, 15]).

Behavioural semantics are likely the biggest source of heterogeneity between SWS frameworks; SAWSDL even omits this aspect altogether. We therefore do not prescribe any particular approach to describing behavioural semantics of services and defer this instead to specific applications and frameworks. Thanks to its simplicity, the MSM captures the essence of services in a way that can support service matchmaking and invocation, while remaining largely compatible with WSMO-based descriptions of Web services, with OWL-S services, and with services annotated according to SAWSDL, WSMO-Lite, and MicroWSMO.

4.4.2 iServe: a Linked Services Publishing and Discovery Platform

iServe uses as its core conceptual model the MSM and it currently includes a number of import mechanisms able to deal with WSDL files including SAWSDL annotations, with descriptions adopting the WSMO-Lite specific extensions, with MicroWSMO annotations of Web APIs as well as with OWL-S service descriptions. These import mechanisms transform the service descriptions into the appropriate terms according to the MSM. Additionally, iServe automatically generates *rdfs:definedBy* links – pointing to the definition file in case additional information is required – and *rdfs:seeAlso* links – pointing to documentation.

Once imported, iServe publishes the semantic annotations of services as Linked Data. Thus every service is assigned a resolvable HTTP URI, through which, humans and machines can access the service descriptions in HTML or in RDF using content negotiation. The registry additionally provides a SPARQL endpoint allowing advanced querying over the services annotations, as well as a read and write RESTful API so that services can easily be retrieved and published from remote applications. The RESTful API is completed with a number of semantic discovery methods that provide more refined discovery than that supported directly via SPARQL, by exploiting the semantic descriptions of services, RDFS inferencing, and similarity measures for more accurate results.

On top of iServe's RESTful API, the registry is complemented by a crawler. Currently it has only been used for targeted import for there are not many SWS descriptions available on the Web. At the time of this writing, iServe registers about 2000 SWS coming from the OWL-S test collection[8] and the SAWSDL test collection[9], 50 services coming from the Jena Geography Dataset[10] annotated manually for evaluation purposes, a test import of around 30 services indexed by `Seekda.com`, and around 20 real services annotated in the context of the use cases of the EU projects SOA4All and NoTube. The current implementation

[8] See `http://projects.semwebcentral.org/projects/owls-tc/`
[9] See `http://projects.semwebcentral.org/projects/sawsdl-tc/`
[10] See `http://fusion.cs.uni-jena.de/professur/jgd/`

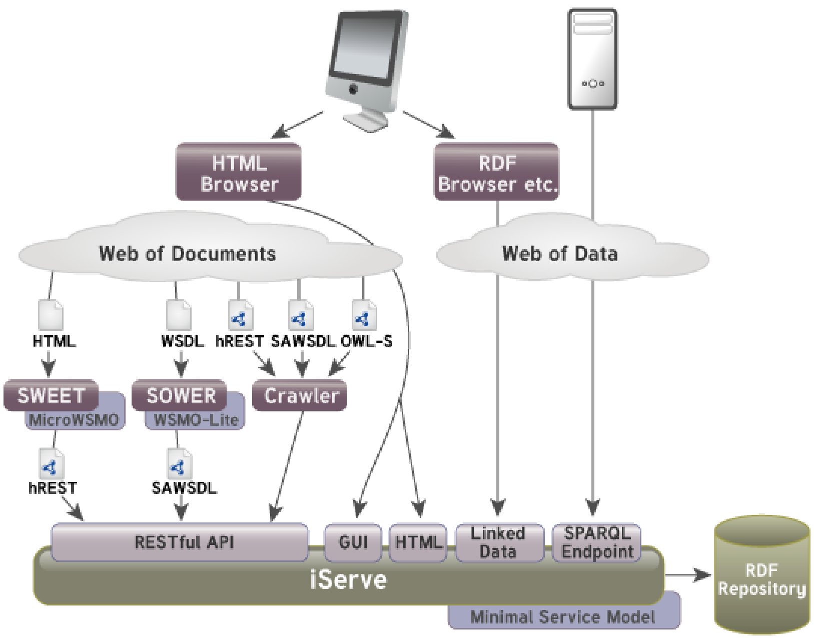

Figure 2: High-level architecture of iServe.

already shows how Web services and Web APIs can be described by means of an homogeneous conceptual model – the Minimal Service Model – and how they can be published as Linked Data, therefore better promoting their discovery based on the use of the well established and adopted Linked Data principles.

5 Services for the Web of Data

The notion of services as well-defined, independent, invokable and distributed pieces of functionality is indeed a very powerful architectural notion for developing distributed systems. Providing functionality in this way independently from the underlying technology provides the capacity for maintaining a loose coupling between integrated components which, when it comes to an environment like the Web, appears as a highly beneficial (if not necessary) feature. Services, may they be traditional Web services or RESTful services, provide therefore a suitable architectural abstraction for the integration of processing capabilities over the Web of Data in a loosely coupled manner. In the remainder of this section we shall cover what services can provide to the Web of Data both as a means for providing new sources of data as well as for processing existing assertions.

5.1 Integrating Legacy Systems

Currently a good part of the Web of Data is generated from existing databases by using tools such as D2R [8]. Indeed, this allows exposing large amounts of data which would otherwise remain private or, in the best case, offered through means that are not that convenient for automated processing. In other cases data is already stored in RDF and can be exposed easily[11]. There is, however, a large body of information owned by companies which are either not interested in offering the information publicly on the Web given its commercial value and/or its sensitivity, or because they do not have the technical skills or interest in exposing the information as Linked Data. Similarly, there is a growing number of streams of data provided by sensors through highly heterogeneous formats and interfaces, which exhibits considerable integration and processing limitations [54].

Web 2.0 developers have long realised the value of Web APIs for accessing highly valuable data on demand. Additionally, Semantic Web researchers have acknowledged the benefit that could be brought by adapting or wrapping these additional sources of information like Web APIs and sensors, so that they can be turned into Linked Data producers, see for instance [54, 51] and the Flickr Wrappr[12]. To a certain extent, the work on sensors is more advanced since there already exists proposals for exposing sensors observations as Linked Data [40]. The work around exposing Web APIs as Linked Data is, however, more an art than a science due to the lack of standard description languages and the extreme heterogeneity characterising Web APIs.

We previously highlighted that Linked Services are such that their inputs and outputs are RDF. As a consequence, they represent a natural means for exposing as Linked Data valuable information previously enclosed within silos, through the annotation of existing Web APIs and WSDL services. Web APIs could in this way be invoked by interpreting their semantic annotations (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 embrace Linked Data principles could be made available as Linked Data easily.

This approach is currently being explored in the context of a number of use cases from European projects such as SOA4All [14] and NoTube [46]. Our current experience, although preliminary at this stage, shows already the applicability and potential of bringing legacy systems to the Web of Data in this manner. Indeed proper care should be taken in order to ensure that Linked Data principles are followed in these cases (see Section 2.3). We anticipate, however,

[11] See `http://backstage.bbc.co.uk/`

[12] See `http://www4.wiwiss.fu-berlin.de/flickrwrappr/`

that at least for services strictly adhering to REST principles this should be relatively straight-forward since they should already define URIs for the resources and offer convenient means for exposing and exploring them.

5.2 Processing Linked Data

Integration and fusion of disparate data coming from the Web of Data hardly takes place nowadays and therefore applications do not perform any ulterior processing of this data other than for presenting it to the user [8]. Generating new data based on what has been found or the provisioning of added-value services that exploit this data thus remains a pending issue. For instance, something as simple and useful as a unit transformation service is still to be provided for the Web of Data. To a certain extent this is natural since the Web of Data is precisely about data; and storing an RDF triple per possible transformation result would simply be absurd since there are infinite possibilities. There is, however, a clear need for enabling the processing of Linked Data in ways such that application developers could conveniently apply them over data gathered at runtime to carry out computations as simple as unit transformations, more complex as deriving similarities between products or services based on the reviews published by users on Revyu.com, or even more advanced as envisioned for the Semantic Web [7].

The Web of Data provides large amounts of machine-processable data ready to be exploited and, as we saw earlier, services provide a suitable abstraction for encapsulating functionality as platform and language independent reusable software. It therefore seems natural to approach the development of systems that process Linked Data by composing Linked Services. These services should be able to consume RDF data (either natively or via lowering mechanisms), carry out the concrete activity they are responsible for (e.g., unit conversion), 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. In a sense this is quite similar to the notion of service mashups [5] and RDF mashups [44] 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. 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.

It is worth noting in this respect the benefit brought by having services annotations available on the cloud as we saw earlier. When developing applications that process Linked Data, discovering useful services could be driven by the data that needs to be manipulated. For instance, developers could easily discover services that manipulate a concrete kind of data or those that produce a certain type by sending SPARQL queries to service registries like iServe [43], or using advanced semantic discovery mechanisms. And, as opposed to traditional Web services repositories like UDDI-based ones, developers would benefit from the existence of semantic annotations in order to filter them based on the semantics of inputs, outputs, their classification with respect to well-known taxonomies, etc. The reuse of ontologies and vocabularies would in turn contribute towards increasing the compatibility of services. In this way, Linked Data application developers would have access to an ever growing body of reusable components ready to be combined and exploited.

5.3 The Services Ecosystem

Integrating services with the Web of Data as depicted before would give birth to a services ecosystem on top of Linked Data, whereby people would be able to collaboratively and incrementally construct complex systems by reusing the results of others, gradually taking us closer to the ambitious vision initially presented for the Semantic Web. In this process, we anticipate that two main families of services will emerge depending on whether they are domain-independent or not.

On the one hand, task-specific yet domain-independent services will allow developers to perform some of the typical tasks involved when processing Linked Data. These activities would range from relatively basic activities such as transforming data between different schemas to more complex actions such as determining how trust-worthy a piece of data is or even, eventually, to carry out knowledge intensive tasks such as Parametric Design or Diagnosis [50]. These domain-independent services which are already starting to appear, e.g., [17], can in fact be seen from a Knowledge Engineering perspective as a new generation of Problem-Solving Methods (PSM) adapted to the Web as some researchers already start considering [57].

This new family of PSMs for the Web of Data will, however, require adapting prior techniques to the new environment, notably with respect to the location, size, and quality of the data to be manipulated. In fact, traditional PSMs were applied within closed environments often with small amounts of manually curated data, whereas in this new scenario data would be obtained automatically from the Web for automated processing, and it would therefore have to be validated, fused, cleaned, and filtered prior to any execution since this would otherwise yield execution errors or incorrect results. We expect that a good deal of domain-independent services will precisely be devoted to performing these tasks. For instance entity resolution, ontology alignment, data cleansing, data fusion, provenance analysis, and trust analysis are some of the domain-independent services that we anticipate would be necessary to develop for the Web of Data. As a side effect, though, it is likely that data quality in the Web of Data will increase as software matures, and especially as it starts being processed by applications which would indirectly detect inconsistencies and incorrect data.

On the other hand, we refer as domain-dependent services to those abstracted away from the technicalities and specificities of Linked Data and generic tasks. This kind of services will be for example those directly providing access to traditional systems in order to obtain some data and carry out actions like sending an SMS or booking a hotel. These services will only be relevant for a particular domain, e.g., hotel services, and will mostly be populated by services directly addressing end-users and therefore better showcasing the potential of the Semantic Web from an end-user perspective. It is worth noting, however, that a wide proliferation of advanced domain-specific solutions for end-users will only occur when a sufficient set of stable domain-independent services able to solve complex tasks will be available. For instance, cross organisational business integration would most likely have to build upon on advanced ontology alignment support for transforming data between different schemas [28]. The systematic development of these applications in a sustainable, efficient, and robust manner shall only be achieved through reuse, and services are a particularly suitable abstraction to carry this out on a Web scale.

6 Conclusions and Outlook

Despite the appealing characteristics of service-orientation principles and technologies, their uptake on a Web-scale has been significantly less prominent than initially anticipated. This limited adoption is due to a number of issues of both socio-economic and technical nature. From a socio-economic perspective service-orientation has for the most part targeted enterprises which, thus far, have been reluctant to publish functionality of the Web. From a technical perspective, service technologies have exhibited a limited level of support for automating activities such as service discovery and composition. SWS have managed to overcome some of the technical limitations of Web services but have in turn introduced additional complexity and overheads. Consequently, SWS have not gained any significant adoption either.

In parallel, the Web is witnessing a dramatic evolution with the advent of Web 2.0 and Linked Data technologies. Web 2.0 has triggered a socialisation of the Web which has placed individuals at the centre of the Web and is widely based on somewhat altruistic contributions of free data and manual labour from users. The Linked Data initiative is in turn devoted to creating what is referred to as the Web of Data, which already provides publicly large amounts of interconnected data concerning a wide range of domains described in terms of light weight ontologies for supporting automated processing.

We have argued that the advent of the Web of Data together with the rise of Web 2.0 technologies and social principles constitute the final necessary ingredients that will give birth to a new wave of services on the Web, which we refer to as Linked Services. We have explored the relationship between services and the Web of Data. In particular we have highlighted that Linked Data represent appropriate principles for publishing services on the Web. We have illustrated how Web services and RESTful services can be brought into the Web of Data by means of simple RDF vocabularies and supporting tools. We have highlighted the fact that the current evolution of the Web of Data is gathering the necessary motivation for the development of advanced applications that process Linked Data. We have outlined that Linked Services are particularly well-suited for supporting developers in creating applications that process Linked Data. We have discussed how the evolution towards more complex Linked Data applications could be supported and we have identified the need for making publicly available domain-independent services that carry out common tasks such Data Cleansing or Trust Analysis.

The overall vision outlined herein represents the roadmap for the research we are currently carrying out trying to expand the capabilities of the Linked Data applications as well as trying to promote and support the use of services on the Web through light weight semantic annotations. This research, like the principles it builds upon, will strive to provide data, resources, tools and engines publicly on the Web in order to eventually lead to the wider uptake of services on a Web scale.

7 Acknowledgements

This work is partly funded by the EU SOA4All project. The authors would like to thank all the members of the SOA4All project, and the Conceptual Models for Services Working Group for their insightful comments in the development of this work. The authors are most grateful to Dong Liu, Maria Maleshkova, Jacek Kopecký, and Alex Simov for their contribution to creating some of the tools mentioned herein. Additionally, we thank Reto Krummenacher for his insightful comments on earlier versions of this work.

References

1. Eyhab Al-Masri and Qusay Mahmoud. Investigating Web Services on the World Wide Web. In *17th International World Wide Web Conference*, pages 795–804, 2008.
2. Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, and Sanjiva Weerawarana. Business Process Execution Language for Web Services Version 1.1. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/, May 2003.
3. Sören Auer, Christian Bizer Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pages 722–735. November 2008.

4. Christian Becker and Chris Bizer. DBpedia Mobile: A Location-Enabled Linked Data Browser. In *Linked Data on the Web (LDOW2008)*, 2008.
5. Djamal Benslimane, Schahram Dustdar, and Amit Sheth. Services Mashups: The New Generation of Web Applications. *IEEE Internet Computing*, 12(5):13–15, 2008.
6. T Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Pru d'ommeaux, and m.c. schraefel. Tabulator redux: Writing into the semantic web. 2007.
7. Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, (5):34–43, May 2001.
8. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.
9. Dan Brickley and Ramanathan V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema, 2002. http://www.w3.org/TR/rdf-schema.
10. Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.91. http://xmlns.com/foaf/spec/, November 2007.
11. Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. International Semantic Web Conference (ISWC 2002), 2002.
12. Ed Chi. The Social Web: Research and Opportunities. *Computer*, 41(9):88 – 91, Sep 2008.
13. Mathieu d'Aquin, Enrico Motta, Marta Sabou, Sofia Angeletou, Laurian Gridinoc, Vanessa Lopez, and Davide Guidi. Toward a New Generation of Semantic Web Applications. *IEEE Intelligent Systems*, 23(3):20–28, 2008.
14. John Davies, John Domingue, Carlos Pedrinaci, Dieter Fensel, Rafael Gonzalez-Cabero, Morgan Potter, Marc Richardson, and Sandra Stincic. Towards the Open Service Web. *BT Technology Journal*, 26(2), 2009.
15. John Domingue, Liliana Cabral, Stefania Galizia, Vlad Tanasescu, Alessio Gugliotta, Barry Norton, and Carlos Pedrinaci. IRS-III: A Broker-based Approach to Semantic Web Services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(2):109–132, 2008.
16. Thomas Erl. *SOA Principles of Service Design*. The Prentice Hall Service-Oriented Computing Series. Prentice Hall, July 2007.
17. Jérôme Euzenat. An API for ontology alignment. In *3rd conference on International Semantic Web Conference (ISWC)*, 2004.
18. Joel Farrell and Holger Lausen. Semantic Annotations for WSDL and XML Schema (SAWSDL). Recommendation, W3C, August 2007.
19. Dieter Fensel, Holger Lausen, Axel Polleres, Jos de Bruijn, Michael Stollberg, Dumitru Roman, and John Domingue. *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2007.
20. Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
21. V. Haarslev and R. Möller. Racer: A Core Inference Engine for the Semantic Web. In *Second International Semantic Web Conference ISWC 2003*, Florida, October 2003.
22. Marc Hadley. Web Application Description Language. Member submission, W3C, August 2009.
23. Luc Clement Andrew Hately, Claus von Riegen, and Tony Rogers. UDDI Specification Version 3.0.2. Technical report, OASIS, 2004.
24. Tom Heath and Enrico Motta. Revyu: Linking reviews and ratings into the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):266–273, 2008.
25. James Hendler and Jennifer Golbeck. Metcalfe's law, Web 2.0, and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, Jan 2008.

26. Martin Hepp. Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2(1):72–99, 2006.
27. Andreas Heß, Eddie Johnston, and Nicholas Kushmerick. ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services. In McIlraith et al. [36], pages 320–334.
28. Jason J. Jung. Semantic Business Process Integration based on Ontology Alignment. *Expert Syst. Appl.*, 36(8):11013–11020, 2009.
29. Takahiro Kawamura, Jacques-Albert De Blasio, Tetsuo Hasegawa, Massimo Paolucci, and Katia P. Sycara. Public Deployment of Semantic Service Matchmaker with UDDI Business Registry. In McIlraith et al. [36], pages 752–766.
30. Jacek Kopecký, Karthik Gomadam, and Tomas Vitvar. hRESTS: an HTML Microformat for Describing RESTful Web Services. In *The 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI2008)*, Sydney, Australia, November 2008. IEEE CS Press.
31. Ulrich Küster and Birgitta König-Ries. Towards Standard Test Collections for the Empirical Evaluation of Semantic Web Service Approaches. *Int. J. Semantic Computing*, 2(3):381–402, 2008.
32. Maria Maleshkova, Jacek Kopecký, and Carlos Pedrinaci. Adapting SAWSDL for Semantic Annotations of RESTful Services. In *Workshop: Beyond SAWSDL at OnTheMove Federated Conferences & Workshops*, 2009.
33. Maria Maleshkova, Carlos Pedrinaci, and John Domingue. Supporting the Creation of Semantic RESTful Service Descriptions. In *Workshop: Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2) at 8th International Semantic Web Conference*, 2009.
34. David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic Markup for Web Services. Member submission, W3C, 2004. W3C Member Submission 22 November 2004.
35. Sheila McIlraith, Tran Son, and Honglei Zeng. Semantic web services. *Intelligent Systems, IEEE*, 16(2):46 – 53, Jan 2001.
36. Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors. *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*. Springer, 2004.
37. N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, 2(16):60–71, March/April 2001.
38. Tim O'Reilly. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. http://oreilly.com/web2/archive/what-is-web-20.html, 2005.
39. Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, Holger Stenzhorn, and Giovanni Tummarello. Sindice.com: a Document-oriented Lookup Index for Open Linked Data. *IJMSO*, 3(1):37–52, 2008.
40. Kevin Page, David De Roure, Kirk Martinez, Jason Sadler, and Oles Kit. Linked Sensor Data: RESTfully serving RDF and GML. In *2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th International Semantic Web Conference (ISWC-2009)*, 2009.
41. Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/, February 2004. Last Visited: March 2005.
42. Carlos Pedrinaci, John Domingue, and Reto Krummenacher. Services and the Web of Data: An Unexploited Symbiosis. In *Linked AI: AAAI Spring Symposium "Linked Data Meets Artificial Intelligence"*, 2010. Accepted for publication.

43. Carlos Pedrinaci, Dong Liu, Maria Maleshkova, Dave Lambert, Jacek Kopecký, and John Domingue. iServe: a Linked Services Publishing Platform. In *Proceedings of Ontology Repositories and Editors for the Semantic Web at 7th ESWC*, 2010.
44. Danh Le Phuoc, Axel Polleres, Manfred Hauswirth, Giovanni Tummarello, and Christian Morbidoni. Rapid Prototyping of Semantic Mash-ups Through Semantic Web Pipes. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *WWW*, pages 581–590. ACM, 2009.
45. Thomi Pilioura and Aphrodite Tsalgatidou. Unified Publication and Discovery of Semantic Web Services. *ACM Trans. Web*, 3(3):1–44, 2009.
46. Hong Qing, Neil Benn, Stefan Dietze, Ronald Siebes, Carlos Pedrinaci, Dong Liu, Dave Lambert, and John Domingue. Two-staged Approach for Semantically Annotating and Brokering TV-related Services. In *The IEEE International Conference on Web Services (ICWS)*, 2010.
47. Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly Media, Inc., May 2007.
48. Marta Sabou. *Building Web Service Ontologies*. PhD thesis, Vrije Universiteit Amsterdam, 2006.
49. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Exploring the Semantic Web as Background Knowledge for Ontology Matching. *Journal of Data Semantics*, (Accepted for publication) 2008.
50. Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, 1999.
51. Juan Sequeda and Oscar Corcho. Linked Stream Data: A Position Paper. In *2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th International Semantic Web Conference (ISWC-2009)*, 2009.
52. Amit Sheth. Semantic Web Process Lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration. Invited Talk at WWW 2003 Workshop on E-Services and the Semantic Web, May 2003.
53. Amit Sheth. Beyond SAWSDL: A Game Plan for Broader Adoption of Semantic Web Services. *IEEE Intelligent Systems Trends & Controversies*, 22(6), November–December 2007.
54. Amit Sheth, Cory Henson, and Satya Sahoo. Semantic Sensor Web. *Internet Computing, IEEE*, 12(4):78–83, 2008.
55. Naveen Srinivasan, Massimo Paolucci, and Katia Sycara. Adding OWL-S to UDDI: Implementation and throughput. In *Proceedings of 1st International Conference on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, 2004.
56. Michael Stollberg, Martin Hepp, and Jörg Hoffmann. A Caching Mechanism for Semantic Web Service Discovery. In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, pages 477–490, November 2007.
57. Frank van Harmelen, Annette ten Teije, and Holger Wache. Knowledge Engineering Rediscovered: Towards Reasoning Patterns for the Semantic Web. In *K-CAP '09: Proceedings of the fifth international conference on Knowledge capture*, pages 81–88, New York, NY, USA, 2009. ACM.
58. Steve Vinoski. Putting the "Web" into Web Services: Interaction Models, Part 2. *IEEE Internet Computing*, 6(4):90–92, 2002.
59. Tomas Vitvar, Jacek Kopecky, Jana Viskova, and Dieter Fensel. WSMO-Lite Annotations for Web Services. In Manfred Hauswirth, Manolis Koubarakis, and Sean Bechhofer, editors, *Proceedings of the 5th European Semantic Web Conference*, LNCS, Berlin, Heidelberg, June 2008. Springer Verlag.