Web Services Are Dead. Long Live Internet Services[1]

Carlos Pedrinaci and John Domingue
Knowledge Media Institute - The Open University,
Walton Hall, Milton Keynes, MK7 6AA, UK
{c.pedrinaci, j.b.domingue}@open.ac.uk

Introduction

Since its inception in the 1960s, the Internet has followed an outstanding and unprecedented growth and evolution. What was initially thought as a military system of interconnected computer networks, quickly evolved into a global platform providing world-wide communication through a variety of protocols. In the 1990s, the Internet reached a major milestone with the advent of the World-Wide Web. Since then, the Web has widely been regarded as the main front-end for the Internet and in fact both terms have even often been wrongly used interchangeably.

Impelled by the hype and interest surrounding the Web, Microsoft and IBM among others seized the opportunity to propose the use of Web Services as a means to support the creation of complex distributed applications over the Internet. Web Services were supposed to be a new technology better supporting traditional Remote Procedure Calls over the Web, which would ultimately lead to the creation of a largely automated Web-based eServices economy, i.e., an economy based on services largely delivered electronically. In reality, however, *Web Services have hardly been adopted beyond the boundaries of enterprises* [1].

A number of technical limitations have been argued to be at the core of this lack of uptake [2], some of which have been addressed by additional specifications, like the WS-* stack, as well as by the Semantic Web Services community [3, 4]. However, recent trends driven by the Web 2.0 phenomenon, have highlighted that socio-economic aspects have been as much an impediment as technological drawbacks. In fact, *the major revolution behind Web 2.0 is not on the use of particular technologies such as AJAX as initially believed, but rather on realising that, on the Web, value largely resides on the data about and the communication between people and this value is subject to the network effect [5]*. On the Web, Web Services never reached the critical mass that would justify the additional efforts and investment.

Recently, companies and institutions have realised the value of the data they hold and that allowing people to access, process and recombine this data could untap additional value with almost no, or at least very little investment.

[1] This document builds upon an earlier publication: Pedrinaci, C., and Domingue, J. (2010) **Toward the Next Wave of Services: Linked Services for the Web of Data**, Journal of Universal Computer Science.

Furthermore, *with the advent and rapid proliferation of mobile devices, the boundaries of the Internet, so far largely and artificially limited to that of the Web, have attained a new dimension in terms of availability, pervasiveness, and applicability* [6]. As a consequence, companies are gradually providing controlled access over parts of their data through Web APIs, allowing third-parties to build added-value solutions that increase the overall value of the data they hold while they retain control over it. These solutions are no longer limited to browser-based access, which enables a pervasive, customised, and continuous capture and delivery of information thus *paving the way for new business models* (e.g., media distribution, mobile assistants[2]). Similarly, a number of governments, like that of the UK, the USA and many others, have embarked in programmes for publicly releasing large amounts of data they control and can hardly exploit.

The evolution we are witnessing seems to indicate that the up to now the adolescent Web is gradually moving from an industrial era, predominated by a manufacturing based economy, on to a post-industrial era based on the provision of information, innovation, and services. In this new context we expect that services but also technologies from the Semantic Web better supporting the exploitation of information and capturing knowledge will gain importance. These services, which we refer to as Internet Services will however be significantly different from Web Services, at the technology level as well as from the perspective of the nature of the service provided which will have a clear business focus rather than a predominantly functional nature.

Internet Services will largely be driven by a number of trends and technologies and will undoubtedly face a number of challenges but will be at the origin of many opportunities. In the remainder of this paper we first analyse the state of art and current evolution around areas that we believe will have a crucial impact on Internet Services including Web 2.0, mobile devices, the Semantic Web and Cloud technologies. On the basis of this analysis we provide an outlook on what we believe will drive the future of Internet Services briefly highlighting where appropriate the role and contribution that we expect SOA4All technologies to have.

On the Evolution of the Internet Ecosystem

From Web to Web 2.0

With over 1 trillion pages and billions of users the Web is one of the most successful engineering artifacts ever created. At the end of 2009 there were 234 million websites of which 47 million were added in the year. The Web is now a rich media repository, the current upload to Flickr is equivalent to 30 billion new photos per year and YouTube now serves over 1 billion videos per day. Although, there are many factors that have driven this growth, Web 2.0 principles and technologies are widely agreed to play a crucial role.

---

[2] See for instance the iTunes store or Siri

The term Web 2.0, commonly attributed to O'Reilly [7], 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 [5, 8]. It is now largely agreed that the major revolution behind Web 2.0 is not on the use of particular technologies as initially believed, but rather on realising that, on the Web, value largely resides on the data about and the communication between people.

Metcalfe's Law anticipated that the value of a network is proportional to the square of the number of users of the system. Although this formula has not been empirically validated, existing social networks like Facebook have shown that this network effect is indeed real. The network effect essentially implies that there is a point where the number of actors involved is sufficient to justify the costs and beyond this point value increases considerably as more users get involved. Interestingly this network effect is still to reach the world of Web Services which hit a wall in 2006 with the closing of the global services repository.

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. In fact, we could even say that a good part of most successful Web companies are largely based on the exploitation of users and the network effect, including Google whose success was initially based on PageRank.

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.

One essential contribution that we may in a sense attribute to the Web 2.0 phenomenon is realising what the real value of data and information is. Especially the fact that large quantities of data and simple algorithms can outperform the most advanced algorithms [9]. A large part or even most of the value of companies like Facebook, Last.fm, and Twitter lie on the massive amounts of data they hold about people. This data, as opposed to mere software, can hardly be reproduced since it requires the interaction of large numbers of users over large periods of time, and therefore establishes a stable and significant competitive advantage over competitors.

This value needs however to be monetised and this is what Web 2.0 companies are achieving through targeted advertisement services, but also by allowing third parties to access part of this data in a controlled manner. Data is thus often exposed via Web APIs, the preferred form for exposing Internet Services, which is one of the core technologies driving innovation on the Internet. We are currently witnessing a proliferation of added-value services that exploit existing Web APIs for previously unanticipated purposes and scenarios.

The Rise of new Front-Ends

Up until now, the terms Internet and Web have (wrongly) often been used interchangeably for referring to the same thing. In some circles this has been blamed on the fact that Microsoft's Web browser is called "Internet Explorer". Only a few technically savvy and rather purist individuals have been actively distinguishing them. In fact, the predominance of the Web browser as the main means for interacting with the Internet with the exception of email clients (often replaced by browsers in any case) quickly blurred the distinction between the two.

The technical evolution and proliferation of new devices such as mobile phones, netbooks, tablets, eBook readers, but also radios, TVs and other multimedia devices, has however brought a high level of diversity with respect to the means used by people to use the Internet. Currently, all sorts of devices are used on a daily basis to access or provide information as well as to exploit functionality exposed on the Internet (including the Web). The pervasiveness of these devices provides an unprecedented level of accessibility to the Internet, allowing users to use their favourite resources and services at anytime anywhere.

In the light of this evolution, this new level of multimodality has also accelerated the process initiated by Web 2.0 sites towards providing controlled access to data silos through Web APIs. By enabling programmatic access to the sources of information other than through Web browsers and HTML, developers can provide customised solutions for exploiting Web resources and services in a way that is better suited for devices that are to be used on-the-go or that present limitations from the user interface point of view.

A large number of companies have embarked into new business ventures to exploit the opportunities brought by this new scenario. One remarkable example is Apple, which has set itself as a leading company in this area through mobile devices such as the iPod, the iPhone and more recently the iPad, but also through multimedia solutions such as the AppleTV. This whole new set of devices exploit the Internet both for continuously accessing Web information, but also as a service and content delivery platform. Other companies like Google, Sony, or even Amazon have created similar solutions that combine novel devices with an Internet-based delivery platform that provides added-value to its users, and represents new sources of revenue and valuable information. Gradually, an Internet-based eServices constellation is starting to appear.

The Semantic Web and the Web of Data

The Semantic Web [10] 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) [11], OWL [12], as well as corresponding tools such as ontology editors [13], RDF(S) storage and querying systems [14] and reasoners [15], 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 [16]. 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 impelled most notably by the Linking Open Data project[3] through DBpedia [17] and followed subsequently by ulterior additions of data about reviews, scientific information and geographical information, to name a few. The latest analysis carried out in September 2010 indicates that there are now over 200 datasets hosting nearly 25 billion RDF statements.

The Web of Data, initially mostly an academic endeavour, is gradually capturing the attention of companies and institutions some of which have already taken steps towards making use of these technologies. Examples of this latest trend are for instance, the acquisition of Metaweb[4] by Google, the release of the OpenGraph[5] API by Facebook, the use of Semantic Web technologies in public services provided by the BBC[6], and the release of governmental data following Linked Data principles by countries like the United Kingdom or the United States of America. This recent activity seems to indicate that the Web of Data is about to reach the critical mass required for main stream adoption.

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 [18]. There are systems that crawl, index and provide intelligent search support over the Web of data like Sindice [19] and Watson [20]. And finally, there are a few domain-specific applications such as DBPedia Mobile [21] 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.

Services on the Web

[3] See http://linkeddata.org/
[4] http://www.freebase.com/
[5] http://developers.facebook.com/docs/opengraph
[6] http://bbc.in/codPLr

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 [3]. 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 [22]. 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 [3]. 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. Web 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 Web 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 [2]. Another fundamental reason is the fact that, as we saw earlier, the work around Web Services has essentially focussed on enterprises which have thus far been reluctant to publish their Web Services on the Web. Today, Seekda.com provides one of the largest indexes of publicly available Web Services which currently accounts for 28,500 with their corresponding documentation. The number of Web Services publicly available contrasts significantly with the billions of Web pages available, and interestingly is not significantly bigger than the 4,000 Web Services estimated to be deployed internally within Verizon [24]. Other academic efforts in crawling and indexing Web Services on the Web have found far lower numbers of services [25].

Stirred 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 [26] when they conform to the REST architectural style [27]. 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[7], Last.fm[8], and Facebook[9]. 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 [28].

Web APIs on the World Wide Web

Services on the Web are increasingly marked by the domination of Web APIs. Still, despite their popularity currently the development of Web APIs is not guided by standards, patterns, or guidelines. It is more an art than a science whereby Web APIs are more often than not solely described in HTML as part of a webpage rather than using an interface description language that could better support software development.

Lately, there is significant effort devoted to this "new kind" of services. A good deal of this work has focused on better understanding the architectural implications of the REST Representational State Transfer (REST) principles [27] as opposed to traditional Web services [29], or to better support their composition and invocation [30, 31]. Most of this work, however, takes REST principles as a starting point despite the fact that on the Web many of the existing APIs do not strictly adhere to these constraints. Before we can make any significant improvement to current practices and technologies, it is thus necessary to reach a deeper understanding of existing Web APIs regarding for instance how they are actually described, what principles they follow and what drives their reuse. It is only in the light of these findings that we may then try and figure out how existing solutions, as opposed to ideal scenarios, could be enhanced and better supported.

To this end we carried out a thorough analysis of large body of Web APIs [32]. The analysis was carried out in February 2010 over more than 200 Web APIs randomly selected from Programmable Web[10]. 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 [33]. For each of these APIs we tracked general information (number of operations, number of mashups, etc), the type of Web APIs (RESTful, RPC oriented, or hybrid), information about the inputs and outputs (formats, parameter types, etc), invocation details and some further details such as the presence of example requests. The analysis highlighted some issues that came as no surprise such as the heterogeneity of Web APIs descriptions and the lack of detail in some of them. But most importantly, the analysis helped us identify some results which we believe are remarkable:

---

[7] http://www.flickr.com/services/api/

[8] http://www.last.fm/api

[9] http://developers.facebook.com/docs/

[10] http://www.programmableweb.com

- The quality of the descriptions leaves a lot to be desired. For instance, only 28% of the APIs clearly state the data-type of their parameters, and only 60% state the HTTP method to be used (fortunately 83% of the APIs provide examples of requests).
- Only about 32% of the Web APIs appear to be truly RESTful (48% are RPC-style and about 20% are hybrid).
- The fact that a service follows REST principles does not seem to have any impact (positive or negative) in its reuse in mashups.
- The description of inputs is particularly heterogeneous including default values, alternative values, coded values, and optional parameters.
- The output format still clearly favours XML (85% of the APIs can provide the results in XML) as opposed to JSON (supported by only 42%).
- More than 80% of the Web APIs require some form of authentication (in some cases only data modification operations require it).

Although certain aspects require further investigation, this analysis has already highlighted many issues that are indeed of particular concern for further automating or better supporting common service manipulation tasks such as their discovery, their composition or their invocation.

Despite the arguable limitations behind Web APIs, they are capturing the attention of large companies, e.g., Facebook, Yahoo, Amazon, and are increasingly used as the main entry point for custom tailored mobile applications. For instance, recently Apple bought Siri, a company that provides a mobile application able to automatically use Web APIs in order to arrange bookings at restaurants and events, or obtain information concerning the weather on the current location on behalf of the user. Similarly, driven by the growing importance of Web APIs, Alcatel bought Programmable Web - the most complete registry of Web APIs.

Last but certainly not least, initiatives like Apple's App Store, or Android's Marketplace have given birth to a relatively new business model. This new kind of business model merges traditional product-oriented software distribution with service delivery including free updates but also access to remote services on the Internet like content delivery. Together with cloud services, the world of mobile applications, supported to a great extent by Web APIs, is currently coping with a significant number of the eServices traded over the Internet.

Cloud Technologies

Hand in hand with the trends introduced so far is the rise of cloud computing as one of the main technology shifts in the Internet. Cloud computing comes as a paradigm shift similar to that from mainframes to client-server in the early 1980s. In essence, cloud computing provides highly advanced, typically customizable and scalable infrastructure to users in a transparent way over the Internet, allowing anybody to use IT infrastructure without the initial investment and the corresponding maintenance. Instead cloud providers maintain the physical resources and offer them to customers on a pay-per-use or subscription basis. Cloud provisioning services include scalable storage, computation, application hosting, or even the provisioning of entire applications accessed remotely via Web interfaces. Some examples of commercial cloud solutions are for instance Google's App Engine, Gmail, and Docs, Amazon's Elastic Computing, or Salesforce.

The popularity of cloud computing has largely been supported by two main trends that we introduced earlier. On the one hand, the increasing importance and the need for rapidly processing ever growing quantities of data has put particularly demanding requirements over companies whose core competencies are not on building and maintaining large data processing centres. On the other hand, the popularisation of smart phones, netbooks, and the like have generated an additional demand for cloud-based solutions that could compensate the reduced resources provided by these resources.

Although considerably different from the initially envisioned Web Services solutions, nowadays, cloud-based solutions are among the most popular eServices traded on a daily basis over the Internet. In fact, a plethora of names have been coined for each of the different kinds of cloud-based solutions that have been devised thus far, e.g., Infrastructure as a Service, Application as a Service, which are globally referred to as XaaS, i.e., Everything as a Service. Cloud-based services because of their core role in IT-based solutions are expected to become one of the main market niches in IT, and as a consequence most large companies provide or are working on the creation of their own solution to retain a competitive position [34].

Outlook

The Fall of Web Services and the Rise of Internet Services

Despite the initial hype behind Web Services, their uptake on a Web scale, as a means to support the creation of inter-organisational business processes that would give birth to a Web-based eServices economy, has largely fallen below the initial estimates. This is not to say, however, that Web Services technologies are not used. On the contrary, Web Services and in general Service-Oriented Architectures are widely used within enterprises. The 4,000 services estimated to be used within Verizon is an unequivocal indicator. It is their public deployment which has not met the initial expectations.

The reasons behind this appear to be partly technical, see for instance the limitations behind UDDI previously mentioned, as well as managerial or cultural in a sense. Traditional companies have been reluctant to publish their services online and the lack of an established market did not create any additional interest. On the contrary, Web companies, generally more innovative than the average IT company, are widely offering their functionality through Web APIs. They use them as a basis for supporting new devices and unanticipated uses and applications that could provide to their users some added-value, and also further revenues.

What is perhaps more important is the fact that the recent proliferation of Web APIs, seem to suggest that the importance of Web Services will decrease at the Internet level. A large part of the major Web sites provide some level of access to their functionality via Web APIs despite the fact that Web Service technologies are undoubtedly more mature and widely supported by software vendors.

Interestingly, and as opposed to what is commonly believed, the previously mentioned survey that we carried out shows that this is not due to the technical advantages brought by REST principles [32]. It is most likely to be a matter of flexibility and simplicity that attracts most to developers.

The interest on providing eServices over the Internet is, however, more accentuated than ever. In fact, we are witnessing an evolution that suggests that the up to now adolescent Internet, is gradually moving from an industrial era, predominated by a software manufacturing based economy, on to a post-industrial era based on the provision of information, innovation, and services. In addition to the traditional shop and advertisement business models, we can now also find on the Internet a growing number of service-based business models like cloud services, e.g., Amazon Elastic Computing, application marketplaces, e.g., Android Marketplace, or content distribution channels, e.g., iTunes store.

Most of these services, which we refer to as Internet Services, are based on technologies other than Web Services and many of them embrace the Internet as the delivery platform rather than tying themselves uniquely to the Web. Indeed access via Web is most often provided but it is not the only communication channel any longer. In fact, for many of these services, e.g., movie rental, the Web can hardly be the delivery platform as of today, nor is it necessary as argued in [35].

The Service Cloud

From a technical perspective most of the characteristics of what is nowadays understood as the cloud find their roots in the early days of mainframes. However, the notion of cloud is rather recent from an economic perspective, and presents certainly new technical challenges given the scale of the Internet in terms of users, devices, and data. The need for highly scalable and flexible resources notably with respect to the storage, processing, and bandwith is such that more and more frequently even small sized companies, Web sites and applications need to rely on the cloud in some way or the other.

This new scenario places cloud providers and technologies as key players for the future of the Internet, which explains the existing fierce competition in this area.
As the market grows and matures, though, experience in previous industries dictates that these solutions will, relatively quickly given the current growth, become comodities. Providers will then have to endeavour to reduce the maintenance costs and providing added-value services on top, which can give them the competitive advantage they need to exist.

We are already seeing certain activity along these lines, which has taken us from mere infrastructure providers, e.g., Amazon Elastic Computing, to higher-level services, e.g., SalesForce.com. And there are indeed a number of emerging acronyms to account for many of the intermediate layers across the whole spectrum. The closer we get to the lower level of cloud service provisioning the closer these solutions will be to being commoditized and the fiercer the competition is likely to be given the number of potential customers. In these cases, we shall witness a race towards reducing costs and providing high quality and highly customizable solutions. At the highest levels of abstraction the competition is likely to be less fierce but also the market size is likely to be smaller.

Research and development around cloud solutions is therefore expected to focus on the one hand on Service Engineering, notably product lines and variability modelling, as well as on semantic technologies for reaching further genericity and improving integration and interoperability. On the other hand we shall continue to see development towards better exploiting the infrastructure through improved backend machinery, enhanced monitoring and management support.

Linked Services

In the previous section we highlighted a number of trends but also limitations around the work on Web services. One clear limitation to the proliferation of Web services on a Web scale has been the limited expressivity and level of automation that can be achieved despite the considerable complexity of the technology stack. Semantic Web Services have brought solutions able to reach a higher-level of automation throughout the life-cycle of Web service-based applications, but they have additionally increased the level of complexity both for humans and for automated processing.

In SOA4All we have developed an approach which we term "Linked Services" [36] which is a re-conceptualisation of Semantic Web Services building on the success of the Linked Open Data initiative. 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. 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.

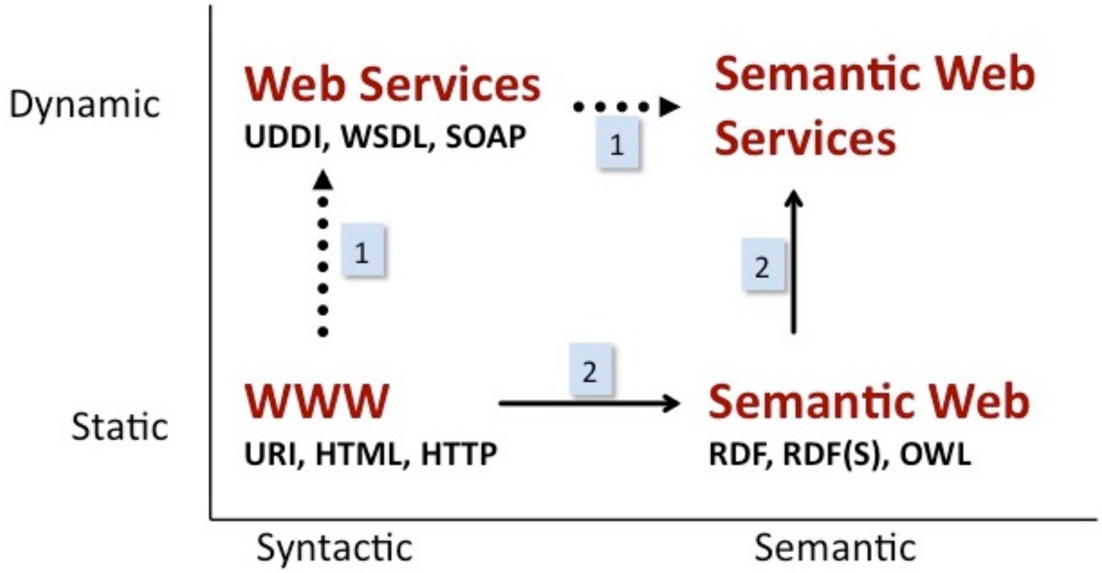

Figure 1. Changing approach to Semantic Web Services

Linked Services alter the way we reach the Semantic Web Services vision as shown in figure 1. Within the figure we differentiate along the dimensions static vs dynamic and syntactic vs semantic. Previous work on Semantic Web Services has taken an approach based on the dashed line (1). Frameworks such as OWL-S [37], WSMO [38] and SAWSDL [39] (formerly WSDL-S) created upper level service ontologies and annotation mechanisms layered on top of WSDL services. In short these frameworks build upon syntactic descriptions of services, that currently remain scarce, to add a semantic layer (re)using ontologies, vocabularies and additional data which only recently can be abundantly retrieved on the Web.

In SOA4All our Linked Services approach takes route 2. Thus, the wealth of semantic data that can nowadays be found on the Web in the form of Linked Data is the starting point and we use our light weight service models to support the production of service-based applications over this data.

Exposing Services as Linked 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.

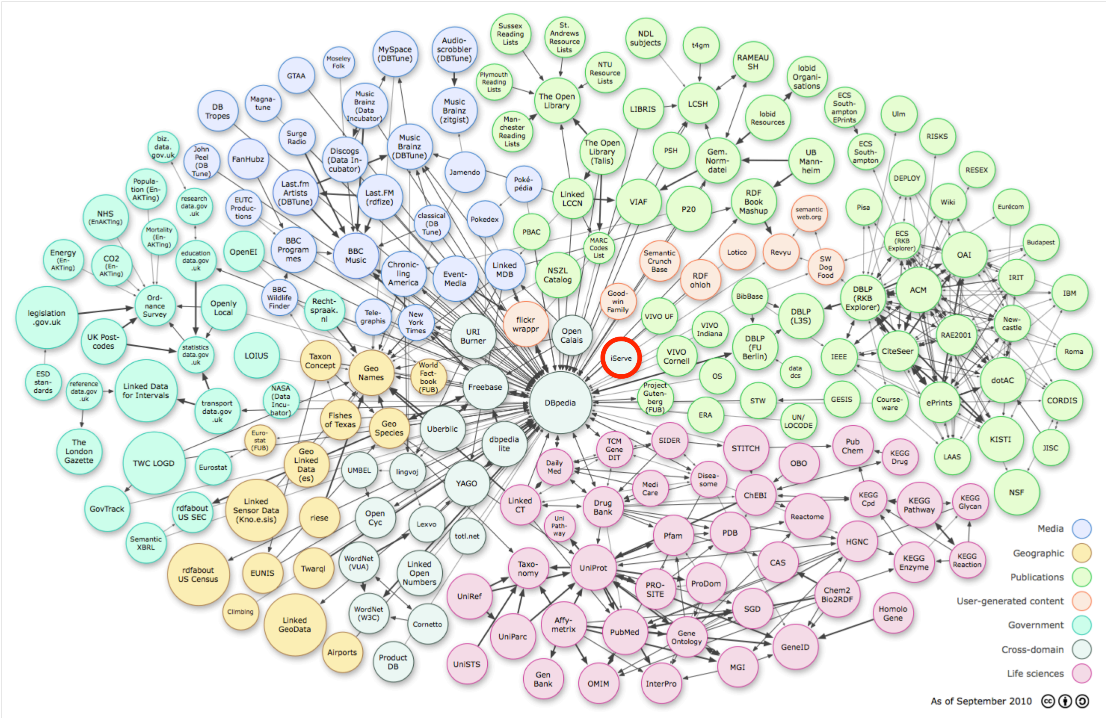

Figure 2. A Snapshot of the Linked Data Cloud from September 2010 incorporating iServe circled in red[11].

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 analysis 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 the wide adoption of service technologies,
- the annotation of services should be simplified as far as possible, and “crowdsourcing” appears to be a particularly effective 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 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.

---

[11] Based on the Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

Syntactic and semantic descriptions of Web services aim at providing information about services in a way that can automatically be processed by machines. However, up until now, these descriptions could 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.

SOA4All has adopted precisely this very approach to publishing annotations of services, may they be WSDL-based or Web APIs. In particular iServe [40], a publicly available registry 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. At the time of this writing iServe contains around 2,000 service annotations coming from test collections, and real-world service annotations which it publishes as Linked Data. In fact, iServe was recently acknowledged by the Linking Open Data initiative as the very dataset on the Web of Data providing Web service descriptions, see figure 2. The registry is, to the best of our knowledge, the first system able to homogeneously publish and provide advanced discovery support for semantic services expressed in several formalisms. It is also the first one to provide advanced discovery over Web APIs and Web services homogeneously.

Supporting Linked Data Applications

Within a number of settings communities are forming to create applications over Linked Data. One of the most well known cases is that associated with the UK open Government Data portal at data.gov.uk. Based on the ongoing release of data from governmental databases over 100 applications now exist in an iTunes-like App store.

Figure 3 below shows one such application "Where does my money go?". Within the display in figure 3 we can see how coloured ovals are used to indicate where the £620B of total annual spending within the UK Government is divided between the various departments including for example, health, education and defense. The slider on the right enables the user to move between different financial years.

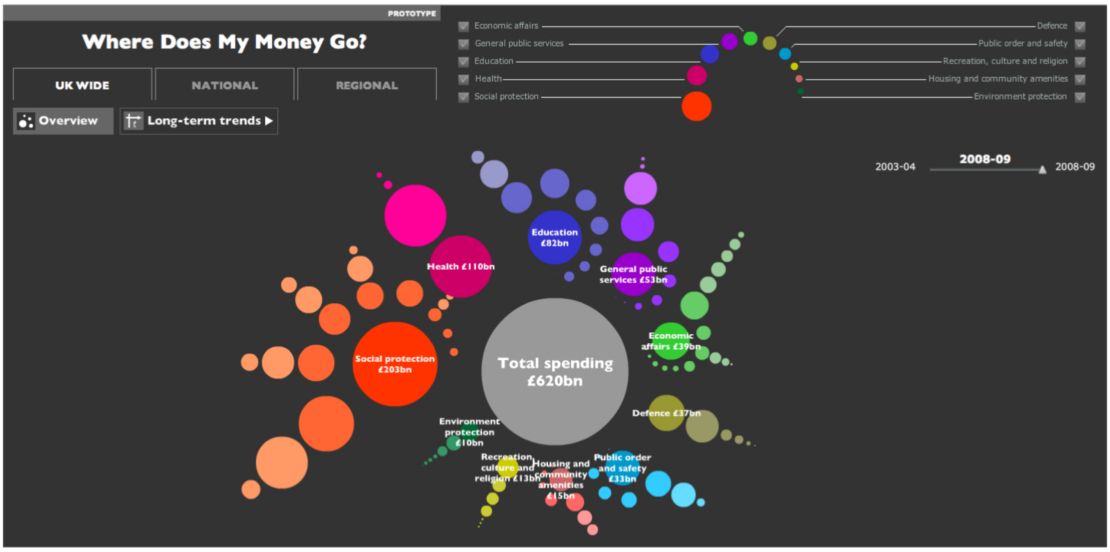

Figure 3. The "Where Does My Money Go?" Linked Data App available from the data.gov.uk website.

One of the interesting aspects of these apps is that they are built in general by the external community and not by any UK Government office. The main action from the UK Government is to facilitate the release of the data. In effect Linked Data Apps mimic the way content in developed within Web 2.0 communities - anyone can build applications and make them available to the wider community.

Linked Data is based on four principles:

1. Use URIs 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 the standards (RDF, SPARQL).
4. Include links to other URIs, so that they can discover more things.

With hindsight it is obvious that the spread of Linked Data is based on the fact that the technology is relatively simple, as with the Web, and also one can take an incremental approach to the quality of the data released.

In contrast to the simplicity and coherence of the Linked Data principles currently Linked Data Apps though are built in an adhoc fashion. In SOA4All we have been considering how could a services approach be used to support the creation and management of Linked Data Apps. In particular how can Linked Data Apps and the components of Linked Data Apps be discoverable, composable and reusable. Our approach, captured in figure 4, presents two main perspectives: using Linked Services for exposing data from legacy systems as Linked Data, and exploiting Linked Data through the composition of Linked Services.

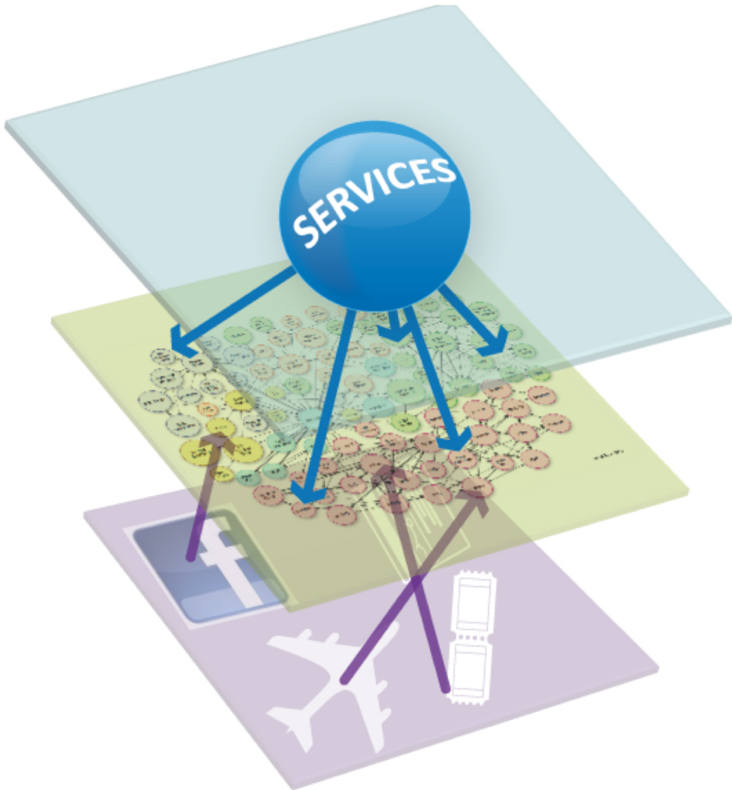

Figure 4. Our Linked Services Approach.

Exposing Web APIs Data as Linked Data using Linked Services

Currently a good part of the Web of Data is generated from existing databases by using tools such as D2R [16]. There is, however, a large body of information owned by companies, which is published on the Web via APIs that provide controlled access to the data. The large majority of these Web APIs do not provide raw data as Linked Data, despite the technical advantages this manner of publishing data has. This is mainly because of the commercial value and sensitivity of the information in some cases or simply because they do not have the technical skills or interest in exposing the information as Linked Data in others. Similarly, there is a growing number of streams of data from the Internet of Things provided by sensors through highly heterogeneous formats and interfaces, which exhibits considerable integration and processing limitations [41].

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 [41] and the Flickr Wrappr[12]. These legacy systems are depicted at the bottom layer of figure 4. 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 [32].

---

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

We previously highlighted that Linked Services are such that their inputs and outputs are RDF. As a consequence, they contain an inherent mechanism for being exposed 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, and Linked Data could be obtained on demand. By doing so, 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 in a systematic manner. The service models [42, 43], related conceptualizations for automating authentication [44], and the execution machinery devised in SOA4All represent a solid starting point towards this goal. Similarly, initiatives like LOS!, also originating from SOA4All, are investigating means for unleashing data behind Web APIs as Linked Data [45].

Supporting the Processing of Linked Data with Linked Services

The 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 [16]. 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, or even more advanced as envisioned for the Semantic Web [10].

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 [28] and RDF mashups [45] with the important difference is 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. Again, SOA4All provides models, tooling and machinery for supporting the creation of mashups, thus establishing a solid basis for this work. Additional focus should be made on providing native primitives for manipulating RDF directly to better support the creation of RDF-based mashups.

It is worth noting in this respect the benefit brought by having service 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 or using advanced semantic discovery mechanisms from service registries like iServe. 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 a growing body of reusable components ready to be combined and exploited.

Socialising Services

In May 2010 Facebook announced their Open Graph Protocol[13] which is based on RDFa and facilitates the integration of web resources into a Facebook social graph. A Facebook 'like' button can be embedded in any web page allowing Facebook users to 'like' any web resource. Figure 5 below shows this facility in use in an Open University news system enabling readers to express preferences over published stories. As can be seen in the figure three readers have expressed that they like the story. Figure 6 shows how the Facebook News feed for one of the story readers has been automatically updated with the resulting comments. These preferences also allow site owners to track the demographic data of users visiting their site.

In the last few months a number of commercial companies have now built sites around this feature. Levi's have a dedicated store[14] which incorporates a likes button for every product. Also, Amazon have integrated their recommendation system to use Facebook profiles through Open Graph. Facebook have also recently integrated Open Graph into the Facebook SDK for the iPhone and Android platforms.

There are two main reasons for highlighting this deployment of semantic technology. Firstly, we now in effect have 500 million (and currently growing) Facebook users semantically annotating the Web from fixed and mobile devices. The probability is that this will in the short to medium term be a major source for semantic data. When making the announcement Facebook's CEO Mark

[13] http://developers.facebook.com/docs/opengraph.
[14] http://store.levi.com/.

Zuckerberg claimed that the technology would result in over 1 billion like buttons spreading across the Web in the first 24 hours[15].
The second more general aspect about the announcement is that one of the world's largest web companies deems Semantic Web technology a suitable choice on which to centre its corporate strategy. In particular, Facebook currently claim that Open Graph is: "the most transformative thing we've ever done for the web"[16].

Figure 5. A screen snapshot from the online news system of the Knowledge Media Institute where Facebook users can say that they like a story.

Of interest here with respect to Open Graph is the extent to which social network platforms will form the basis for all future web based applications. The thesis is

---

15 http://techcrunch.com/2010/04/21/facebook-like-button/

16 http://technology.timesonline.co.uk/tol/news/tech_and_web/the_web/article7104354.ece

that as humans are social entities we normally (in the physical world) have a tendency to use our social contacts to find information and interpret data rather than carry out search. Thus, a web application platform should incorporate a social networking component.

Although social technologies have mostly been applied to products (e.g., Amazon ratings for books) and documents (e.g., Facebook 'like'), these techniques are in principle applicable to any addressable Web resource. Service descriptions as produced by SOA4All tools, as we saw earlier, are transformed into Web resources and published as Linked Data. They become therefore addressable resources on which one can apply social networking solutions and technologies. In SOA4All we have also explored and validated the application of social networking technologies to services. In particular it is possible to generate comments and taggings for services and these annotations are used by a recommender system to suggest services that are likely to interest a given user. Further exploitation of the relationship between users and services (including crowdsourcing the annotations) appear to be one of the major directions for future work in the area.

Figure 6. A screen snapshot the user's Facebook News Feed were the actions in the News website have been reflected.

Conclusions

The Internet and the Web are among the most successful artefacts engineered by the human being thus far. The outstanding evolution of the Web together with the latest technological advances such as mobile devices has recently taken us from an environment mostly dominated by user-to-machine Web-based oriented communication, to a platform providing way more diverse modes of interaction than those based on HTTP and HTML.

In this very period, the importance and interest on services which slowed gradually in the last 5 years after the initial hype and high expectation, is growing back again impelled by a new wave of lighter services originating from

Web 2.0 technologies which are called Web APIs. This "new" kind of technology has been embraced successfully by major Web sites and Web-based companies leading to an unprecedented level of technological innovation on the Web most prominently exemplified by mashups.

A large number of these services are based on the exploitation of (large quantities of) valuable data as opposed to providing mere functions from a pure computational perspective which was perhaps the main role played by "traditional" Web Services. It seems now that the economic relevance of these publicly provided services together with the need for a continuous innovation on the Web are providing the necessary ingredients for a wide use of services in the Open Web. The interest on providing eServices over the Internet is more accentuated than ever.

Other trends like the ever growing quantities of heterogeneous data that need to be efficiently handled on a daily basis are generating a number of opportunities, notably for cloud computing solutions, but are also accentuating the need for improving the means by which data is shared and integrated. Linked Data is the main initiative in this respect and has already gathered a notable take up as highlighted by the 250 datasets providing around 25 billion RDF statements.

In this paper we have highlighted that, given the aforementioned trends and technological evolution, the widespread use of "classical" Web Services on a Web scale as initially envisioned following the hype behind WSDL and related technologies, does not seem realistic. However, it is increasingly evident that the use and provisioning of services over the Internet (including the Web) is gaining a central role; one that is most likely to increase over the years. These services which we refer to as Internet Services will be predominantly business oriented and will contemplate a wide range of distribution channels in addition to the Web per se.

In the last section of this paper we have presented a technical outlook on the main aspects that we believe will drive the future of Internet Services including notably cloud computing, the Web of Data and the social Web. In this context, we have highlighted, where appropriate, the role and solutions that SOA4All technologies can provide. Notably, we have presented the vision of Linked Services as a promising refocus of Semantic Web Services and have highlighted the main principles and technologies these shall build upon. Although by no means exhaustive this paper sketches a number of lines for future work and development embracing and exploiting latest technological and socio-economic trends that we believe will characterise the Future Internet.

Acknowledgements

This work is partly funded by the EU SOA4All project. The authors would like to thank all the members of the SOA4All project.

References

[1] J. Davies, , J. Domingue, , C. Pedrinaci, , D. Fensel, , R. Gonzalez-Cabero, , M. Potter, , M. Richardson, , S. Stincic, Towards the Open Service Web. BT Technology Journal. 26, (2009)

[2] T. Pilioura, , A. Tsalgatidou, Unified Publication and Discovery of Semantic Web Services. ACM Trans. Web. 3, 1--44 (2009)

[3] Erl, T. SOA Principles of Service Design. Prentice Hall (2007)

[4] C. Pedrinaci, , J. Domingue, , A. Sheth, Handbook of Semantic Web Technologies, Semantic Web Applications. 2, (2010)

[5] J. Hendler, , J. Golbeck, Metcalfe's law, Web 2.0, and the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web. (2008)

[6] A. Sheth, Computing for human experience: semantics-empowered sensors, services, and social computing on the ubiquitous web. Ieee internet computing. 14, 88--91 (2010)

[7] T. O'reilly, What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. (2005)

[8] E. Chi, The Social Web: Research and Opportunities. Computer. 41, 88 -- 91 (2008)

[9] A. Halevy, , P. Norvig, , F. Pereira, The unreasonable effectiveness of data. IEEE Intelligent Systems. 24, 8-12 (2009)

[10] T. Berners-Lee, , J. Hendler, , O. Lassila, The Semantic Web. Scientific American. 34--43 (2001)

[11] D. Brickley, , R. V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema. (2002)

[12] P. F. Patel-Schneider, , P. Hayes, , I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax. (2004)

[13] N. F. Noy, , M. Sintek, , S. Decker, , M. Crubezy, , R. W. Fergerson, , M. A. Musen, Creating Semantic Web Contents with Protege-2000. IEEE intelligent systems. 2, 60--71 (2001)

[14] J. Broekstra, , A. Kampman, , F. Van Harmelen, Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. (2002)

[15] Haarslev, V. , Moller, R. Racer: A Core Inference Engine for the Semantic Web. Second International Semantic Web Conference ISWC 2003 (2003)

[16] C. Bizer, , T. Heath, , T. Berners-Lee, Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS). (2009)

[17] S. Auer, , C. B. G. Kobilarov, , J. Lehmann, , R. Cyganiak, , Z. Ives, DBpedia: A Nucleus for a Web of Open Data. The Semantic Web. 722--735 (2008)

[18] T. Berners-Lee, , J. Hollenbach, , K. Lu, , J. Presbrey, , E. P. D'ommeaux, , M.C. Schraefel, Tabulator redux: writing into the semantic web. (2007)

[19] E. Oren, , R. Delbru, , M. Catasta, , R. Cyganiak, , H. Stenzhorn, , G. Tummarello, Sindice.com: a Document-oriented Lookup Index for Open Linked Data. Ijmso. 3, 37-52 (2008)

[20] M. D'aquin, , E. Motta, , M. Sabou, , S. Angeletou, , L. Gridinoc, , V. Lopez, , D. Guidi, Toward a New Generation of Semantic Web Applications. IEEE Intelligent Systems. 23, 20--28 (2008)

[21] Becker, C. , Bizer, C. DBpedia Mobile: A Location-Enabled Linked Data Browser. Linked Data on the Web (LDOW2008) (2008)

[22] T. Andrews, , F. Curbera, , H. Dholakia, , Y. Goland, , J. Klein, , F. Leymann, , K. Liu, , D. Roller, , D. Smith, , S. Thatte, , I. Trickovic, , S. Weerawarana, Business Process Execution Language for Web Services Version 1.1. (2003)

[23] L. C. A. Hately, , C. Von Riegen, , T. Rogers, UDDI Specification Version 3.0.2. (2004)

[24] Stollberg, M. , Hepp, M. , Hoffmann, J. A Caching Mechanism for Semantic Web Service Discovery. 6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007) (2007)

[25] Al-Masri, E. , Mahmoud, Q. Investigating Web Services on the World Wide Web. 17th International World Wide Web Conference (2008)

[26] Richardson, L. , Ruby, S. RESTful Web Services. O'reilly Media, Inc. (2007)

[27] R. T. Fielding, Architectural styles and the design of network-based software architectures. (2000)

[28] D. Benslimane, , S. Dustdar, , A. Sheth, Services Mashups: The New Generation of Web Applications. Ieee internet computing. 12, 13--15 (2008)

[29] Pautasso, C. , Zimmermann, O. , Leymann, F. RESTful Web Services vs. "big"' Web Services: making the right architectural decision. WWW '08: Proceeding of the 17th international conference on World Wide Web (2008)

[30] Curbera, F. , Duftler, M. , Khalaf, R. , Lovell, D. Bite: workflow composition for the web. ICSOC '07: Proceedings of the 5th international conference on Service-Oriented Computing (2007)

[31] C. Pautasso, RESTful Web Service Composition with BPEL for REST. Data & Knowledge Engineering. 68, 851 - 866 (2009)

[32] Maleshkova, M. , Pedrinaci, C. , Domingue, J. Investigating Web APIs on the World Wide Web. European Conference on Web Services (ECOWS) (2010)

[33] M. Hadley, Web Application Description Language. (2009)

[34] The Economist, Clash of the clouds. The Economist. 392, 80 - 82 (2009)

[35] C. Anderson, M. Wolff, The Web Is Dead. Long Live the Internet. (2010)

[36] C. Pedrinaci, , J. Domingue, Toward the next wave of services: linked services for the web of data. Journal of universal computer science. 16, 1694--1719 (2010)

[37] D. Martin, , M. Burstein, , J. Hobbs, , O. Lassila, , D. Mcdermott, , S. Mcilraith, , S. Narayanan, , M. Paolucci, , B. Parsia, , T. Payne, , E. Sirin, , N. Srinivasan, , K. Sycara, OWL-S: Semantic Markup for Web Services. (2004)

[38] 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 (2007)

[39] J. Farrell, , H. Lausen, Semantic Annotations for WSDL and XML Schema (SAWSDL). (August 2007)

[40] Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J. , Domingue, J. iServe: a Linked Services Publishing Platform. Proceedings of Ontology Repositories and Editors for the Semantic Web at 7th ESWC (2010)

[41] Sheth, A., Henson, C., Sahoo, S. Semantic Sensor Web. Internet Computing, IEEE. 12, 78--83 (2008)

[42] Kopecky, J. , Gomadam, K. , Vitvar, T. hRESTS: an HTML Microformat for Describing RESTful Web Services. The 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI2008) (2008)

[43] Vitvar, T. , Kopecky, J. , Viskova, J. , Fensel, D. WSMO-Lite Annotations for Web Services. Proceedings of the 5th European Semantic Web Conference (2008)

[44] Maleshkova, M. , Pedrinaci, C. , Domingue, J. , Alvaro, G. , Martinez, I. Using semantics for automating the authentication of web apis. International Semantic Web Conference (ISWC) (2010)

[45] Phuoc, D. L. , Polleres, A. , Hauswirth, M. , Tummarello, G. , Morbidoni, C. Rapid Prototyping of Semantic Mash-ups Through Semantic Web Pipes. WWW (2009)

[46] R. Krummenacher, B. Norton, and A. Marte. Towards linked open services and processes. In 3rd Future Internet Symposium, Berlin, September 2010.