

## Service Discovery

Matthias Klusch  
German Research Center for Artificial Intelligence (DFKI)  
Saarbruecken, Germany  
[klusch@dfki.de](mailto:klusch@dfki.de)

### Synonyms

Web services, Service-oriented architectures, Semantic web services, Service search and selection.

### Glossary

**CAN:** Content-Addressable Network  
**DHT:** Distributed Hash Table  
**JSON:** JavaScript Object Notation  
**NAICS:** North American Industrial Classification System  
**OWL-S:** Ontology Web Language for Services  
**REST:** Representational State Transfer  
**SA-WSDL:** Semantically Annotated WSDL  
**SA-REST:** Semantic Annotation of Web Resources  
**SIC:** Standard Industrial Classification  
**SML:** Service Modeling Language  
**SOA:** Service-Oriented Architecture  
**SOAP:** Simple Object Access Protocol  
**UDDI:** Universal Description, Discovery, Integration  
**UNSPSC:** United Nations Standard Products and Services Code  
**USDL:** Unified Service Description Language  
**UUID:** Unique Universal Identifiers  
**WADL:** Web Application Description Language  
**WSDL:** Web Service Description Language  
**WSML:** Web Service Modeling Language

### Definition

Service discovery is the process of locating existing services that are relevant for a given request based on the description of their functional and non-functional semantics. Approaches to service discovery differ in their support of service description language(s), the organization of the search, and the utilized means of service selection.

### Introduction

The continuous proliferation of web services which encapsulate business software and hardware assets, e-business, or social software applications in the web 2.0 holds promise to further revolutionize the way of interaction within today's society and economy. A service can be defined as a kind of action, performance, or promise that is exchanged for value between provider and client. In other words, it is a provider-client interaction that creates and captures value for all parties involved. At present, there are

tens of thousands of web services for a huge variety of applications and in many heterogeneous formats available for the common user of the web. One main challenge of web service technology is to provide scalable and effective means for an automated discovery of relevant services with minimal human intervention in any user and application context. This paper provides an overview of service discovery in a nutshell. For a more comprehensive survey on the subject, the interested reader is referred to, for example, (Crasso et al., 2011; Klusch, 2008b; Klusch, 2012).

### Preliminaries

Service discovery can be performed in different ways depending on how the services of the considered search space are described, how the search process is organized, and which means of service selection are used for the search.

**Service Description.** In general, a web service can be described in terms of what it does and how it actually works. These aspects of its functional semantics (aka capability) are described in a service profile and a service process model, respectively.

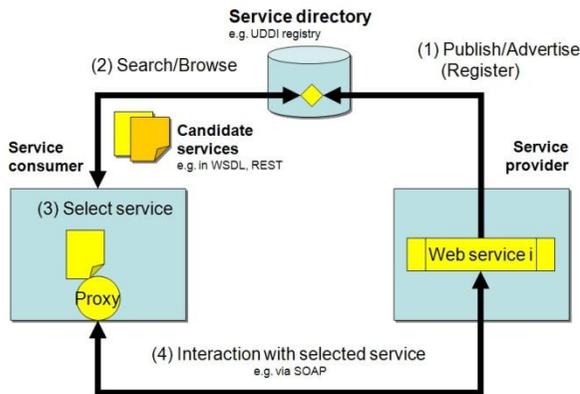
A *service profile* describes the signature of a service in terms of its input and output (I/O) parameters, and the service specification, i.e. the preconditions and effects (P/E) of the service execution. The profile also describes non-functional service semantics such as information about its provenance, name, business category, pricing, delivery constraints, and quality. Prominent approaches to represent such profiles are the XML-based web service description languages WSDL (Chinnici et al., 2007), SML (Pandit et al. 2009), USDL (Oberle et al., 2013), and WADL (Hadley, 2009) and the HTML micro-format hREST (Kopecky et al. 2008). Other examples are the textual documentations of RESTful services (Fielding et al., 2002) and the ontology-based service description languages OWL-S (Martin et al., 2004), WSML (De Bruijn & Lausen, 2005), SAWSDL (Farrell & Lausen, 2007), SA-REST (Gomadam et al., 2010), and Linked USDL (Pedrinaci & Leidig, 2011).

A *service process model* describes the operational behavior of a service in terms of its internal control and data flow. Such models are described, for example, in OWL-S, WSML and USDL by use of standard workflow operators like sequence, split+join, and choice, while other representation approaches are adopting process algebraic languages like the pi-calculus, and Petri-nets for this purpose.

**Discovery Architectures.** Approaches to organize the service search can be classified as either directory-

based (aka structured) or directory-less (aka unstructured), or hybrid peer-to-peer (P2P). In the scenario of a directory-based search, service providers register their services with either one central and possibly replicated directory, or multiple distributed (federated) service directories at distinguished nodes of the underlying network. Service consumers are informed about available services in the network only through these directory nodes.

*Centralized directory-based service discovery* can be performed by using either a contemporary web search engine, or a specialized web service search engine, or a dedicated and authoritative web service directory with query interface. In any case, the W3C web service interaction lifecycle for service-oriented architectures (SOA) expects a central service directory to act as an intermediary between provider and consumer (cf. Fig. 1) though it represents a potential single point-of-failure and performance bottleneck for dependant applications.



**Figure 1: W3C web service interaction life-cycle**

*Decentralized directory-based service discovery* relies on a structured P2P network overlay and a respective query routing protocol. In this case, services are placed and discovered by all peer nodes according to the global distribution or replication scheme and the location mechanism of the network. Classic examples of structured P2P overlays are the DHT-based Chord ring, Pastry, Tapestry, CAN, P-Grid, or a compound routing index, and a hierarchically structured federation of service directories with super-peers. In general, this type of service discovery provides a search guarantee in the sense of total recall and logarithmic complexity in the size of the network for finding popular, i.e. highly replicated, as well as rare services. On the other hand, it comes at the cost of high communication overhead for publishing and maintaining the structured overlay when peers are

joining or leaving the network, or the set of services which they provide changes.

*Directory-less service discovery* is performed in an unstructured P2P network without any given overlay structure. Each peer initially knows only about services provided by its own or its direct neighbor peers. Prominent examples of service location or query routing schemes in such networks are query flooding and k-random walks with replication and caching strategies, as well as informed probabilistic adaptive search. This type of service discovery is effective for finding popular but not rare services and provides only probabilistic search guarantees, i.e., incomplete recall.

*Hybrid P2P service discovery* is performed in networks with structured and unstructured overlay parts. For example, service requests can be routed to super-peers in the structured overlay part in order to find relevant rare services, or processed with restricted flooding or broadcasting to peers of the unstructured network part to find relevant popular services.

**Service Selection.** The performance of service discovery depends, in particular, on the used service selection method. The process of service selection (aka service matchmaking) encompasses (a) the pairwise semantic matching of a given service request with each service that is registered with the matchmaker, and (b) the semantic relevance ranking of these services. In contrast to service brokers, a matchmaker only returns a rank list of relevant services and related provenance information to its human user or application but does not handle the interaction with selected services. In principle, a matchmaker can be used for any organizational approach to service discovery. For example, matchmakers can be part of either the query interface of one central directory, or federated directories, or local directories owned by peers in an unstructured P2P network (Klusck & Sycara, 2001).

**Types of service selection.** Current approaches to the semantic matching of web services can be classified as non-logic-based, logic-based, or hybrid depending on the nature of reasoning means used for this purpose. Non-logic-based semantic matching exploits, for example, means of graph matching, schema matching, data mining, and text similarity measurement, while logic-based semantic matching performs logical reasoning on service descriptions. Hybrid semantic matching is a combination of both types of matching, while adaptive selection means learn how to best aggregate different matching filters off or on line. In any case, it is commonly assumed

that service requests and offers are given in the same format, or are appropriately transformed by the considered service matchmaker.

**Benchmarking.** Systems and tools for service discovery, in particular service matchmakers, can be evaluated according to the following five criteria: (1) the support of different service description formats and languages, (2) the usability of the tool and required amount of effort for its configuration, (3) the support of service composition planning through, for example, context-aware pruning of the search space or interactive recommendations for a step-wise forward or backward chaining of services by the user, (4) the policy to preserve user data privacy, and (5) the service retrieval performance in terms of correctness and average query response time over given service test collections. Correctness is commonly evaluated with classical information retrieval measures such as average precision and macro-averaged precision at standard recall levels for binary relevance, as well as the normalized discounted cumulative gain or Q-measure for graded relevance. Current evaluation initiatives include the WS-Challenge and the SWS Challenge for (semantic) web service composition, and the S3 Contest for semantic web service selection (Klusch, 2012; Küster et al., 2009).

## Web Service Discovery

Most web services are described in the standard WSDL, USDL, or according to the REST paradigm of the web. Some service providers also publish the functional description of their services in multiple formats and languages. The number and variety of web services which are available in the public web appears tremendously high, though there are still no common and comprehensive statistics on the subject available. However, the portal seekda.com reported about 30k web services in November 2011, and the public directory programmableweb.com alone already offered about 16k single or composite RESTful web services in March 2013. In this section, we focus on the discovery of WSDL and RESTful services.

**WSDL Services.** The W3C web services framework offers a set of technical specifications including WSDL and SOAP that codify mechanisms for XML-based interoperability between business services that are accessible in the web over stateless HTTP. A web service which profile is described in WSDL (in short: WSDL service) exposes one or multiple operations which consume inputs and produce outputs both encoded in XML. Applications or other services can interact with these operations by means of XML-SOAP messaging.

**Description.** The XML-based W3C standard language WSDL describes the functionality of a service by the set of signatures of its service operations, and the set of network endpoints or ports (URIs) at which these operations can be invoked and how this can be achieved. In particular, each port is associated with a respective interface which binds the operation to a given protocol for transport and messaging. The definitions of the I/O messages of each service operation include references to their data types which are defined in common XMLS namespaces. Several non-functional service parameters can be added to such a WSDL service profile on demand. The description of service profiles in WSDL remains stateless, since the specification of service preconditions and effects is not part of the standard. Besides, a WSDL service description does not include any process model. In this respect, WSDL is commonly considered as weak in describing what the service actually does.

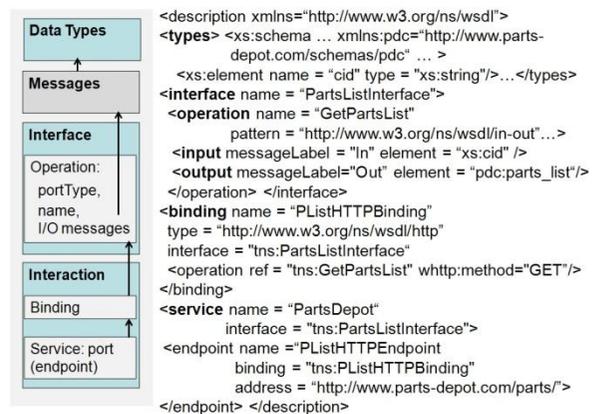


Figure 2: Example of web service description in WSDL.

**Discovery and selection.** Most approaches to directory-based or directory-less discovery of WSDL services utilize means of non-logic-based semantic selection, in particular, structural XML and text similarity-based matching.

*Central directory-based discovery* of WSDL services is most popular. One classic example is the instantiation of the W3C service interaction lifecycle (cf. Fig. 1) with some UDDI-compliant (Bellwood et al., 2004) registry of WSDL services and using SOAP (Mitra & Lafon, 2007) for service interaction. In such a XML-based UDDI business registry (UBR), the services and their providers are categorized with standard taxonomies such as NAICS, SIC and UNSPSC. Registration of WSDL services and their retrieval from an UBR is through its APIs PublishSOAP and InquireSOAP. In general, an UBR

may provide information on the business entities of services (aka white pages), service categories (aka yellow pages), and the technical model (tModel) of services (aka green pages). Search queries to an UBR are regular expressions with identifiers and keywords for service tModels, names and categories. Accordingly, service selection by an UBR is, in principle, based on string matching without any logical reasoning on service relationships or non-functional service parameters. Thus, it requires a rather cumbersome browsing of the registry by the user to find relevant services. Since 2005, UDDI is not supported by its originally main supporters IBM and Microsoft.

Examples of non-UDDI compliant WSDL service directories are RemoteMethods.com, Xmethods.net, WebserviceX.net, webservicelist.com, service-repository.com, and wsindex.org. Most of them rely on keyword search, and service category or simple list browsing. An example of a specialized web service search engine is Woogle (Dong et al., 2004) which retrieves and indexes WSDL services from a given set of UBRs. The WSDL service selection tool WSDLAnalyzer (Zinnikus et al., 2006) returns a rank list of similar WSDL services for a given WSDL service, and produces a mapping between their I/O messages. In particular, it recursively computes the XML-tree similarity of a given pair of WSDL files with integrated text matching of tree node names using WordNet-distance and string matching, and a binary compatibility check of XMLS data types. Other approaches to WSDL service selection exploit techniques for matching software components, graphs or schemas (Wang & Stroulia, 2005), or perform a full text matching of service names or the content of WSDL files as a whole. In addition, there are approaches to preference-, trust- or reputation-based matching of non-functional parameters including quality of service, pricing, and service policies (Crasso et al., 2011; Garofalakis et al., 2006).

*Decentralized directory-based discovery* of WSDL services in structured P2P networks still appears in its infancies. One example is the PWS system (Li et al., 2004) in which WSDL files and requests are distributed and located in a Chord ring of service peers. The DUDE system (Banerjee et al. 2005) enables WSDL service discovery in a hierarchical DHT-based overlay for multiple local UDDI registries. There is no approach to directory-less discovery of WSDL services in unstructured P2P networks available yet.

**REST Services.** Web service interaction is not restricted to XML-SOAP messaging. A RESTful web

service (in short: REST service) represents resources which states shall be accessed only over the stateless HTTP according to the REST paradigm of the web. The call of a REST service with given input values may return output values in XML or in the text-based JSON or RSS formats. For example, the call of some REST service “books” hosted at a portal www.bookstore.com with input parameter “subject” for books on the topic Eclipse is of the form `http://www.bookstore.com/books/?subject=computers/eclipse` and may return book list entries like `<booklist:book url = http://www.bookstore.com/books/0321288157 title="Eclipse Distilled"/>` in XML.

**Description.** At present, there is no standard for describing the functionality of REST services. Most REST service APIs are documented by their developers on dedicated, public HTML pages in more or less plain text and tables; some APIs are described in XML-based WADL files or the HTML micro-format hRESTS. This heterogeneity is a major barrier for the automated discovery of REST service APIs in the web to date.

**Discovery and selection.** Centralized *directory-based* discovery of REST services can be performed with the prominent directory programmableweb.com. It offers about 9k REST service APIs and 7k REST service mash-ups (as of April 2013). Another open source REST API directory in the web is APIS.io. The selection of relevant REST services through their query interfaces is done by keyword search which relies on the textual description of the registered service APIs, or other meta-information provided by their developers. The web services search engine seekda! identifies relevant REST service APIs based on adaptive text classification and feature extraction. An approach to automated extraction of information from REST service APIs like service operation name, description, and URI is proposed in (Ly et al., 2012). It integrates means of DOM processing, information extraction, and natural language processing. An approach to structural and textual matching of REST services is proposed in (Khorasgani et al., 2011). In this case, a given pair of REST service APIs is first semi-automatically converted into WADL descriptions. The REST service matching score is then computed as the maximum flow in the graph of WADL service description elements.

Approaches to *directory-less* discovery of REST services in mobile ad-hoc networks mostly rely on simple look-up methods based on the matching of service classes, UUID, or service attribute names (Schiele et al., 2004).

## Semantic Web Service Discovery

One major challenge of automated service discovery is to make service-based applications or intelligent software agents actually “understand” the semantics of service requests and offers. From the perspective of strong AI, this requires some well-founded logic specification of service profile and process model. However, contemporary web service descriptions are lacking such formal semantics. It is well-known that this problem can be addressed by exploiting semantic web technologies (Hitzler et al., 2011).

**Description.** The key idea of encoding web service semantics not only in a machine-readable but machine-understandable way is as follows: The semantics of web service interface elements are described by references to appropriate concepts and rules which are formally defined in a shared ontology in some W3C standard ontology language like RDFS or OWL2. Such semantically annotated web services are called semantic web services (in short: semantic services). Current frameworks for semantic service description include OWL-S (Martin et al., 2004), WSML (De Bruijn & Lausen, 2005), the W3C standard SAWSDL (Farrell & Lausen, 2007), and Linked USDL (Pedrinacci & Leidig, 2011) which is USDL modeled in RDFS. These ontology-based semantic service description languages mainly differ in their formal logic-based foundation and the possible extent of annotating services.

**OWL-S.** In OWL-S the service I/O parameters are annotated with concepts which are exclusively defined in the formal logic-based W3C standard ontology language OWL2 (cf. Fig. 3). Service preconditions and effects may be specified in the formal semantic web rule language SWRL.

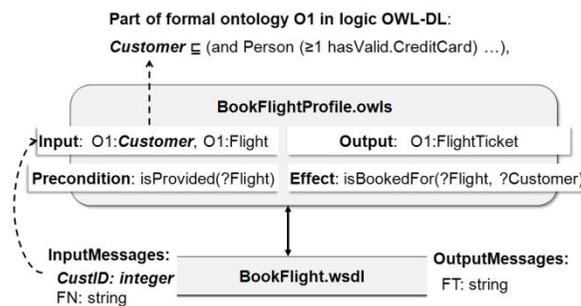


Figure 3: Example of semantic service profile in OWL-S

**WSML.** The description of service profile semantics in one of five variants of WSML is formally grounded in the respective variant of the logic programming language F-Logic (Fensel et al., 2010). Both, WSML and OWL-S, are also providing the developer with a

set of workflow operators like sequence, iterate, choice, and split+join for specifying the operational semantics of a single or composite service in its process model. The process model can be mapped to service orchestrations in BPEL as the semantic service can be grounded with a WSDL service.

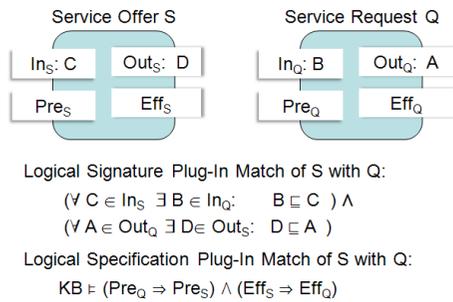
**SAWSDL and SA-REST.** The W3C standard SAWSDL allows the annotation of WSDL service elements with references to web resources of any media type such as plain text, video, picture, audio podcast, and concepts in a formal ontology. The same approach is taken in the SA-REST framework for semantically annotating REST service APIs (Gomadani et al., 2010). Both SAWSDL and SA-REST do not allow the specification of preconditions and effects, and the handling of semantic annotations is completely outside these frameworks. In this sense, unlike OWL-S and WSML, neither of both has unique formal semantics. For more details on semantic service description, the reader is referred to, for example, (Klusch, 2008) and the above cited relevant technical specifications.

At present, there are no public statistics about semantic web services available. A survey conducted with the semantic service search engine Sousuo (Klusch & Xing, 2008) in April 2013 reported about 3500 semantic services in OWL-S, WSML, WSDL-S, and SAWSDL in the public web, though most of them are available only in distinguished test collections.

**Discovery and Selection.** In the past decade, the semantic web research community has developed a wide range of solutions for the automated discovery and selection of semantic services. The degree of semantic correspondence between a pair of semantic web services particularly relies on the matching of the semantic annotations of their service profile and/or process model.

**Types of selection.** The types of semantic service selection are logic-based, non-logic-based and hybrid semantic. Classical examples of logic-based semantic matching filters are the logical I/O concept subsumption-based plugin match of service signatures, and the logical specification plugin match of preconditions and effects (cf. Fig. 4). Logical and full functional (IOPE) profile matching combines the scores of logical signature (IO) and specification (PE) matching. Non-logic-based semantic matching of annotated service signatures is mostly based on the textual similarity of the concept names, or the text of their logical unfolding in the referenced ontology. Additional examples include the structural similarity-based matching of I/O concepts in terms of the

shortest path or upward co-topic distances between them in the shared ontology.



**Figure 4: Logic-based semantic service plugin matching**

Currently, most approaches to semantic service selection are hybrid, i.e. they combine non-logic-based with logic-based semantic service matching. Besides, the majority of them support either OWL-S or SAWSDL, but only a few are devoted to WSML, or other description formats, and hardly any matchmaker is even language-agnostic (Klusch, 2012). In the following, we focus on approaches to the discovery and selection of services in OWL-S and SAWSDL. More information on the subject is provided, for example, in (Klusch, 2008b; Klusch, 2012).

**Centralized discovery and selection.** There are quite a few tools and systems for central directory-based discovery of semantic services available.

**Matchmakers.** For example, the matchmaker iSeM (Klusch & Kapahnke, 2012) performs an adaptive and hybrid semantic selection of OWL-S services. Its logic-based semantic matching of services relies on the computation of strict and approximated logical I/O concept subsumption relations, and the logical specification plugin relation. Like its predecessor OWLS-MX2 (Klusch et al., 2009), it also performs non-logic-based semantic matching with different classical token-based text similarity measures, as well as ontology-based structural matching of signature annotation concepts. Finally, it learns how to best aggregate the results of its matching filters by use of a binary SVM relevance classifier with an evidential coherence-based weighting scheme.

An example of a hybrid semantic and adaptive matchmaker for SAWSDL services is LOG4SWS (Schulte et al., 2010). Like iSeM it performs a logical service signature matching which is complemented with ontology-based structural matching based on the shortest path lengths between concepts. In case there are no semantic annotations of WSDL service signature elements, it exploits the WordNet distance

between the element names. LOG4SWS does not consider service preconditions and effects, but learns off line how to best aggregate the matching results by use of an ordinary least square-based classifier.

The logic-based semantic service matchmaker SPARQLent (Sbodio et al., 2010) considers the full functional profile of OWL-S services. It performs a RDF entailment rule-based matching of I/O concepts, preconditions and effects described in SPARQL.

According to the results of the international S3 contest (Klusch, 2012), iSeM and LOG4SWS are currently the best performing matchmakers for OWL-S and SAWSDL services, respectively. In fact, they provide the best trade-off between average precision and response time.

An example of a hybrid semantic matchmaker for WSML services is WSMO-MX (Klusch & Kaufner, 2009): It recursively determines service matching degrees based on ontology-based signature parameter type matching, logical constraint (PE) matching, and syntactic matching with text similarity measurements.

**Specialized search engines.** Examples of search engines for semantic services are S3E (Giantsiou et al., 2009) and Sousuo (Klusch & Xing, 2007). The latter performs a meta-search through the public web search engines Google and A9, and complements it by crawling the web with its own focused topic crawler. It also utilizes the semantic web search engine Swoogle for an inverse ontology-based search, and performs a full text search of the public scientific archive citeseer in the web. Service selection through Sousuo's query interface relies on full-text or keyword search in its XML-encoded service index.

Alternatively, the S3E engine is encoding the profiles of crawled semantic services in RDF. The selection of services from an internal RDF store with SPARQL relies, in particular, on textual matching of profile parameters. Another search engine which is restricted to a QoS-based discovery of semantic services is presented in (Vu et al., 2006).

**Registries.** At present, there are no central and authoritative registries of semantic services available in the public web. Public collections of semantic services are, for example, the prominent OWLS-TC for OWL-S services, the SAWSDL-TC for SAWSDL services, and hREST-TC for annotated REST services; each of these collections is available at the portal semwebcentral.org. iServe (Pedrinaci et al., 2010) is a software platform that can be used to build and maintain a registry of semantic services described in SAWSDL, OWL-S, MicroWSMO, and WSMO-Lite. The services are internally represented in iServe according to a minimal service model and then exposed in HTML and RDF as linked services with a unique and resolvable HTTP URI. Any iServe registry

can be queried through a SPARQL endpoint. For service selection, iServe provides means of keyword search, functional classification, and service I/O parameter matching based on RDFS reasoning.

**Centralized P2P search.** An example for the discovery of WSDL-S (a predecessor of SAWSDL) services in a structured P2P system is the METEOR-S system (Verma et al., 2005). It consists of a set of service providing and consuming peers which may form groups on given domains or topics, and one central super-peer which serves as a central service matchmaker for all peers. For this purpose, the super-peer maintains and utilizes a global registry ontology which covers the concept taxonomies of all local service registries of peers in the network. The super-peer also provides the peers with mappings between the message types and signature annotation concepts of registered services. The non-logic-based semantic selection of services by the super-peer relies on structural XMLS matching, and the computation of N-Gram-based text similarities and taxonomic relations. The super-peer can be replicated for reasons of scalability.

**Decentralized discovery and selection.** A *directory-based* discovery of OWL-S services in structured P2P systems can be performed, for example, with the AGORA-P2P system (Küngas & Matskin, 2006). It relies on a Chord ring for distributed storage and location of services. In particular, the service signature concept labels are hashed as literals to unique integer keys such that peers holding the same key are offering services with equal literals in the circular key space. Service selection for multi-key queries relies on exact key matching.

*Directory-less discovery* of semantic services can be performed with, for example, the RS2D system (Basters & Klusch, 2006). It is a solution for informed and adaptive probabilistic service search in unstructured P2P networks. In particular, each peer dynamically builds and maintains its local view of the semantic overlay of the network, and uses the OWLS-MX matchmaker for hybrid semantic service selection. A peer also learns the average query-answering behavior of its direct neighbors in the network. The peer's decision to whom to forward a semantic service request is then driven by its estimated probabilistic risk of routing failure in terms of semantic loss and communication costs. Other examples are discussed, for example, in (Klusch, 2008b; Staab & Stuckenschmidt, 2006).

## Future Directions

Despite the progress made in the field in the past decade, a major open problem is the scalable and

dynamic interleaving of discovery of services with their composition, negotiation, and execution in the converging Internet of Things and Internet of Services. Examples of potential applications of solutions are intelligent condition monitoring based on large-scale, wireless and semantic sensor service networks, the intelligent collaborative design of products in shared 3D spaces, and mobile ad-hoc and context-aware business travel planning or product recommendation services.

## Cross-references

**WSDL, Web Service Composition, OWL (Web Ontology Language), RDF**

## References

1. Bellwood, P.; Capell, S.; Clement, L.; Colgrave, J.; Dovey, M.J.; Feygin, D.; Hatley, A.; Kochman, R.; Macias, P.; Novotny, M.; Paolucci, M.; von Riegen, C.; Rogers, T.; Sycara, K.; Wenzel, P.; Wu, Z. (2004): UDDI 3.02. [uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)
2. Banerjee, S.; Basu, S.; Garg, Sh.; Garg, S.; Lee, S.J.; Mullan, P.; Sharma, R. (2005): Scalable Grid Service Discovery based on UDDI. Proc. 3<sup>rd</sup> Intern. Workshop on Middleware for Grid Computing; ACM Press.
3. Basters, U.; Klusch, M. (2006): RS2D: Fast Adaptive Search for Semantic Web Services in Unstructured P2P Networks. Proc. 5<sup>th</sup> Intern. Semantic Web Conference, Athens, USA; Springer.
4. Chinnici, R.; Moreau, J.J.; Ryman, A.; Weerawarana, S. (2007): Web Services Description Languages 2.0. W3C Recommendation: [www.w3.org/TR/wsdl20/](http://www.w3.org/TR/wsdl20/), WSDL 1.1 (2001): [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)
5. Crasso, M.; Zunino, A.; Campo, M. (2011): A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. *J Database Management*, 22(1), IGI Global.
6. De Bruijn, J.; Lausen, H. (2005) eds.: Web Service Modeling Language (WSML). [www.w3.org/Submission/WSML/](http://www.w3.org/Submission/WSML/)
7. Dong, X.; Halevy, A.Y.; Madhavan, J.; Nemes, E.; Zhang, J. (2004): Similarity Search for Web Services. Proc. 30<sup>th</sup> Conference on Very Large Databases.
8. Farrell, J.; Lausen, H. (2007): Semantic Annotations for WSDL and XML Schema. [www.w3.org/TR/sawSDL/](http://www.w3.org/TR/sawSDL/)
9. Fensel, D.; Lausen, H.; Polleres, A. (2010): Enabling Semantic Web Services. Springer.
10. Fielding, R.T.; Taylor, R.N. (2002): Principled Design of the Modern Web Architecture. *J Transactions on Internet Technology*, 2(2); ACM Press.
11. Garofalakis, J.; Panagis, Y.; Sakkopoulos, E.; Tsakalidis, A. (2006): Contemporary web service discovery mechanisms. *J Web Engineering*, 5(3), Rinton Press.
12. Giantsiou, L.; Loutas, N.; Peristeras, V.; Tarabanis, K. (2009): Semantic service search engine (S3E): An

- Approach for Finding Services on the Web. Proc. 2<sup>nd</sup> World Summit on the Knowledge Society. Springer.
13. Gomadam, K.; Ranabahu, A.; Sheth, A. (2010): SA-REST: Semantic Annotation of Web Resources. [www.w3.org/Submission/2010/SUBM-SA-REST-20100405/](http://www.w3.org/Submission/2010/SUBM-SA-REST-20100405/)
  14. Hadley, M. (2009): Web Application Description Language. [www.w3.org/Submission/wadl/](http://www.w3.org/Submission/wadl/).
  15. Hitzler, P.; Krötzsch, M.; Rudolph, S. (2011): Foundations of Semantic Web Technologies. CRC Press.
  16. Khorasgani, R.R.; Stroulia, E.; Zaiane, O.R. (2011): Web service matching for RESTful web services. Proc. 13<sup>th</sup> IEEE Intern. Symposium on Web Systems Evaluation. IEEE Press.
  17. Klusch, M. (2008): Semantic Web Service Description. In: M. Schumacher, H. Helin, H. Schuldt (Eds.) CASCOM - Intelligent Service Coordination in the Semantic Web. Chapter 3. Birkhäuser.
  18. Klusch, M. (2008b): Semantic Web Service Coordination. In: M. Schumacher, H. Helin, H. Schuldt (Eds.) CASCOM - Intelligent Service Coordination in the Semantic Web. Chapter 4. Birkhäuser.
  19. Klusch, M. (2012): Overview of the S3 Contest: Performance Evaluation of Semantic Service Matchmakers. In: Blake, M.B.; Cabral, L.; König-Ries, B.; Küster, U.; Martin, D. (eds.): Semantic Web Services; Chapter 2; Springer
  20. Klusch, M.; Sycara, K. (2001): Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In: A. Omicini et al. (eds.) Coordination of Internet Agents; Chapter 8; Springer.
  21. Klusch, M.; Kapahnke, P. (2012): The iSeM Matchmaker: A Flexible Approach For Adaptive Hybrid Semantic Service Selection. *J Web Semantics*, 15, Elsevier.
  22. Klusch, M.; Kaufer, F. (2009): WSMO-MX: A Hybrid Semantic Web Service Matchmaker. *J Web Intelligence and Agent Systems*, 7(2), IOS Press.
  23. Klusch, M.; Xing, Z. (2008): Deployed Semantic Services for the Common User of the Web: A Reality Check. Proceedings of 2<sup>nd</sup> IEEE Intern.Conference on Semantic Computing (ICSC), Santa Clara, USA, IEEE Press.
  24. Klusch, M.; Fries, B.; Sycara, K. (2009): OWLS-MX: A Hybrid Semantic Web Service Matchmaker for OWL-S Services. *J Web Semantics*, 7(2), Elsevier.
  25. Kopecky, J.; Gomadam, K.; Vitvar, T. (2008): HTML Microformat for Describing RESTful Web Services and APIs. Proc. Intern. Conference on Web Intelligence; Sydney, Australia; IEEE Press.
  26. Küster, U.; Koenig-Ries, B.; Klusch, M. (2009): Evaluating Semantic Web Service Technologies: Criteria, Approaches and Challenges. In: Progressive Concepts for Semantic Web Evolution: Applications and Developments. IGI Global.
  27. Küngas, P.; Matskin, M. (2006): Semantic Web Service Composition through a P2P-Based Multi-Agent Environment. Proc. 4<sup>th</sup> Intern. Workshop on Agents and Peer-to-Peer Computing., LNCS 4118, Springer.
  28. Li, Y.; Zou, F.; Wu, Z.; Ma, F. (2004): PWSO: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network. Proc. APWeb04, LNCS 3007; Springer.
  29. Ly, P.A.; Pedrinaci, C.; Domingue, J. (2012): Automated information extraction from Web APIs documentation. Proc. 13th Intern. Conference on Web Information System Engineering. Springer.
  30. Martin, D.; Burstein, M.; Hobbs, J.; Lassila, O.; McDermott, D.; McIlraith, S.; Narayanan, S.; Paolucci, M.; Parsia, B.; Payne, T.; Sirin, E.; Srinivasan, N.; Sycara, K. (2004): OWL-S.: Semantic Markup for Web Services. [www.w3.org/Submission/OWL-S/](http://www.w3.org/Submission/OWL-S/)
  31. Mitra, N.; Lafon, Y. (2007): SOAP 1.2 Primer [www.w3.org/TR/2007/REC-soap12-part0-20070427/](http://www.w3.org/TR/2007/REC-soap12-part0-20070427/)
  32. Pandit, B.; Popescu, V.; Smith, V. (2009): Service Modeling Language 1.1. [www.w3.org/TR/sml/](http://www.w3.org/TR/sml/)
  33. Pedrinaci, C.; Liu, D.; Maleshkova, M.; Lambert, D.; Kopecky, J.; Domingue, J. (2010): iServe: A Linked Services Publishing Platform. Proc. 7<sup>th</sup> Extended Semantic Web Conference Workshop on Ontology Repositories and Editors for the Semantic Web. LNCS, Springer.
  34. Pedrinaci, C.; Leidig, T. (2011): Linked USDL Core. [www.linked-usdl.org/ns/usdl-core](http://www.linked-usdl.org/ns/usdl-core)
  35. Oberle, D.; Barros, A.; Kylau, U.; Heinzl, S. (2013): A unified description language for human to automated services. *J Information Systems*, 38(1); Elsevier.
  36. M. L. Sbodio, D. Martin, C. Moulin (2010): Discovering Semantic Web services using SPARQL and intelligent agents. *J Web Semantics*, 8(4), Elsevier.
  37. Schiele, G.; Becker, C.; Rothermel, K. (2004): Energy-Efficient Cluster-based Service Discovery for Ubiquitous Computing. Proc. 11th ACM SIGOPS European Workshop; Belgium.
  38. Schulte, S.; Lampe, U.; Eckert, J.; Steinmetz, R. (2010): LOG4SWS.KOM: Self-Adapting Semantic Web Service Discovery for SAWSDL. Proc. 6<sup>th</sup> World Congress of Services. IEEE Press.
  39. SOAP (2007): W3C Recommendation SOAP 1.2 [www.w3.org/TR/2007/REC-soap12-part0-20070427/](http://www.w3.org/TR/2007/REC-soap12-part0-20070427/)
  40. S. Staab, H. Stuckenschmidt (2006) eds.: Semantic Web and Peer-to-Peer. Springer.
  41. Stroulia, E.; Wang, Y. (2005): Structural and Semantic Matching for Assessing Web-Service Similarity. *J Cooperative Information Systems*, 14; World Scientific.
  42. Verma, K.; Sivashanmugam, K.; Sheth, A.; Patil, A.; Oundhakar, S.; Miller, J. (2005): METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *J Information Technology and Management*, 6(1); KluwerAcademics
  43. Vu, L-H.; Hauswirth, M.; Porto, F.; Aberer, K. (2006): A search engine for QoS-enabled discovery of semantic web services. *J Business Process Integration and Management*, 1(4).
  44. Zinnikus, I.; Rupp H.-J.; Fischer, K. (2006): Detecting Similarities between Web Service Interfaces: The WSDL Analyzer. Proc. 2<sup>nd</sup> Intern. Workshop on Web Services and Interoperability. Wiley.