

# Publishing Semantic Descriptions of Services

Luc Moreau, Simon Miles, Juri Papay, Keith Decker, Terry Payne

{L.Moreau, sm, jp, ksd, trp}@ecs.soton.ac.uk

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ UK

## Abstract

Service discovery in large scale, open distributed systems is difficult because of the need to filter out services suitable to the task at hand from a potentially huge pool of possibilities. Semantic descriptions have been advocated as the key to expressive service discovery, but the most commonly used service descriptions and registry protocols do not support such descriptions in a general manner. In this paper, we present an approach and implementation for service registration and discovery that uses an RDF triple store to express semantic service descriptions and other task/user-specific metadata, using a mechanism for attaching structured and unstructured metadata. The result is an extremely flexible service registry that can be the basis of a sophisticated semantically-enhanced service discovery engine, an essential component of a Semantic Grid.

## 1 Introduction

Service discovery is a difficult task in large scale, open distributed systems such as the Grid and Web, due to the potentially large number of services advertised. In order to filter out the most suitable services for the task at hand, many have advocated the use of semantic descriptions that qualify functional and non-functional characteristics of services in a manner that is amenable to automatic processing [2, 5, 13].

*Semantic discovery* is the process of discovering services capable of meaningful interactions, even though the languages or structures with which they are described may be different. Typically, a semantic discovery process relies on *semantic annotations*, containing high-level abstract descriptions of service requirements and behaviour. In this paper, we focus on the means to *register* such semantic annotations, and *discover* services using them.

Current standards in the Web Services and Grid communities (including UDDI and WSDL) do not directly support semantic discovery of services [10]. However, given

that these standards have been agreed upon by the community, their existence promotes inter-operability with components such as workflow enactment engines.

An essential element in semantic discovery is the ability to augment service descriptions with additional information, i.e. *metadata*. Providers may adopt various ways of describing their services, access policies, contract negotiation details etc. However, many resource consumers also impose their own selection policies on the services they prefer to utilise, such as provenance, derived quality of service, reputation metrics etc. Furthermore, it is useful to add such *metadata* not only to service descriptions, but also to any other concept that may influence the discovery process, e.g. supported operations, types of arguments, businesses, users. Such metadata may be structured according to published ontologies, facilitating unambiguous interpretation by multiple users, especially in the case of a public registry; alternatively, such metadata may also be raw and unstructured, in the case of a personal registry used by a single user.

Since current Grid and Web Services standards are not capable of semantic service descriptions, we believe that an information model supporting not only UDDI and WSDL descriptions, but also general metadata attachment [9], would provide us with a uniform way of querying and navigating service information. We see the use of RDF triples [11] (subject, predicate, object) as the means to represent all the information in a uniform manner. This information will be stored in a triple store, which can be queried uniformly through the use of a query language such as RDQL [7]. Besides an information model, it is also critical to offer programmatic interfaces that would allow both publishers and third-party users to register their semantic information. Therefore, we have implemented the UDDI interface to this triple store, and additional interfaces to publish metadata and discover services according to metadata.

Our work is a component of the myGrid architecture for semantic service discovery ([www.mygrid.org.uk](http://www.mygrid.org.uk)). The functionality we are discussing here allows the attachment of metadata in the form of semantic annotations to services descriptions; such semantic descriptions can be retrieved, and used for reasoning by a *Semantic Find* component, whose description and interaction with the current component are discussed in a companion paper [8]. The specific contributions and the remaining sections of this paper are the following: (i) A discussion of limitation of existing technologies; (ii) An illustration of the kind of annotations supported by our service directory; (iii) An overview of the interfaces we provide to register and use such annotations.

## 2 Limitation of Existing Approaches

The UDDI service directory (Universal Description, Discovery, and Integration) [12] has become the de-facto standard for service discovery in the Web Services community. Service queries are typically white or yellow pages based: services are located based on a description of their provider or a specific classification (taken from a published taxonomy) of the desired service type. A query typically returns a list of available services, from which a subset may conform to a known and/or informally agreed

upon policy and thus can be invoked. Such approaches work well within small, closed communities, where a priori definitions of signatures and data formats can be defined. However, across open systems, no assumption can be made about how desired services are described, how to interact with them, and how to interpret their corresponding results. Additionally, service providers typically adopt different ways to model and present services, often because of the subtle differences in the service itself. This raises the problem of *semantic inter-operability*, which is the capability of computer systems to operate in conjunction with one another, even though the languages or structures with which they are described may be different. *Semantic discovery* is the process of discovering services capable of semantic inter-operability.

Current standards in the Web Services and Grid communities do not support semantic discovery of services [10]. UDDI supports a construct called tModel which essentially serves two purposes: it can serve as a namespace for a taxonomy or as a proxy for a technical specification that lives outside the registry [6]. We believe that such a tModel construct has some intrinsic limitations. While there is no doubt that service classifications are useful, services are not the only entities to be classified. For instance, classifications can also be defined for individual operations or their argument types. However, it is not convenient to use searching mechanisms for services that are distinct from those for their argument types. Likewise, a tModel's reference to an external technical specification, such as a WSDL file describing a service interface, also implies that a different mechanism is required for reasoning over service interfaces.

UDDI provides no data structures to represent either the abstract or concrete details contained within a WSDL document, but only a standard way to express that a service implements a particular WSDL interface. A new proposal allows tModels to reference specific bindings and port types [4]. However, this extension still does not provide access to, or queries over, operations or messages, which would allow the discovery of services capable of specific operations.

WSDL, the interface definition language of Web Services, itself suffers from some limitations, as illustrated by Figure 1 displaying the interface of an existing bioinformatics service (BLAST). It identifies a portType composed of one operation, which takes an input message comprising two message parts `in0` and `in1`. These parts are required to be of type `string`, but the specification does not tell us what the meaning of these strings is supposed to be. In fact, these are supposed to be biological sequences, for which many formats are supported. This example was chosen because it precisely illustrates limitations of existing service descriptions. While this interface specification could easily be refined by using an XSD complex type [3], it is unrealistic to assume that all services in an open environment will always be described with the appropriate level of detail. Moreover, should it be so, we cannot expect all service providers to always use type definitions expressed with the terms of reference adopted by a user.

Other relevant initiatives are DAML-S and BiMOBY, which we cannot describe here due to space constraints. Both approaches offer some form of semantic annotation, but are restrictive, in particular, because they are not compatible with the UDDI

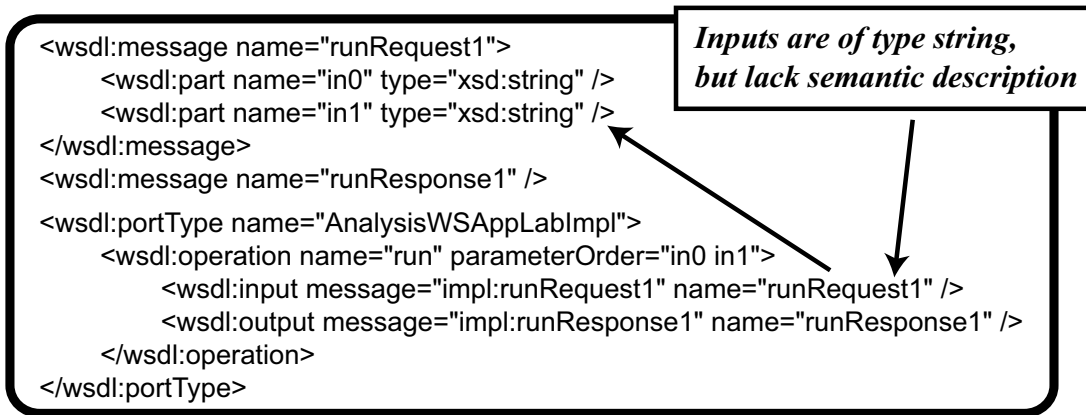


Figure 1: Basic Local Alignment Search Tool (BLAST) Interface Excerpt

standard.

### 3 Extending Service Descriptions

Having discussed the limitations of existing technologies, we now focus on the capabilities of our service directory. Specifically, we look at the ways of attaching ratings and functionality profiles to services, and semantic types to operation arguments. Our presentation is based on examples that were generated by dumping the contents of our service directory. The notation adopted in the presentation is N3 format [1].

In Figure 2, we show the representation of a service annotated by two numerical ratings, with different values, and provided by different authors at different time. The node b1 of the linear representation is anonymous node denoting the service with the metadata attachment of type “NumericRating”.

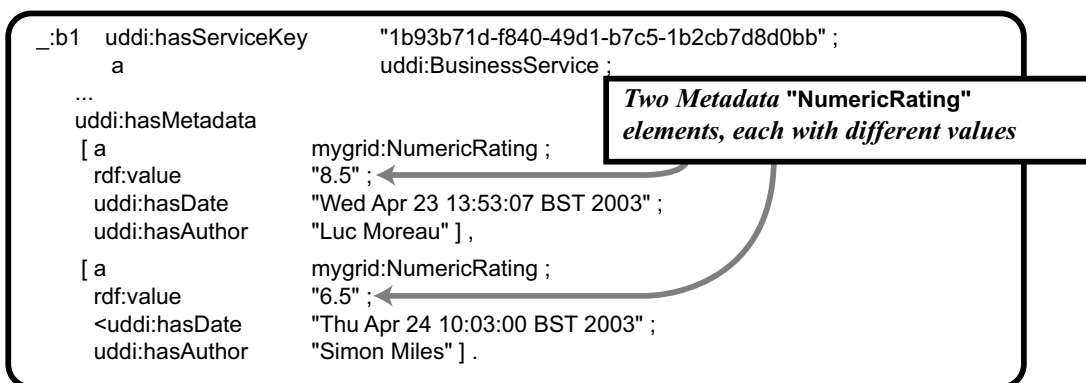


Figure 2: Rating Attachment (N3 Notation)

In myGrid, we describe services by a service profile [13] specifying

which kind of method they use (*uses\_method*), which task they perform (*perform\_task*), which resources they use (*uses\_resources*) and what application they are wrapping (*is\_function\_of*). A relevant excerpt of the service directory contents is displayed in Figure 3, with *b1* denoting a service and *Pe577955b-d271-4a5b-8099-001abc1da633* the “myGrid profile”.

```

mygrid:Pe577955b-d271-4a5b-8099-001abc1da633
  mygrid:uses_resources    mygrid2:nucleotide_sequence_database ;
  mygrid:uses_method      mygrid2:mygrid_bioinformatics_primitive_service_operation;
  mygrid:performs_task    mygrid2:pairwise_local_aligning ;
  mygrid:is_function_of   mygrid2:blastn .

```

**myGRID service profile**

```

_:b1 uddi:hasServiceKey    "e577955b-d271-4a5b-8099-001abc1da633" ;
  a uddi:BusinessService ;
  uddi:hasName
    [ rdf:_1
      a uddi:NameBag ] ;
  uddi:hasMetadata
    [ a mygrid:Profile ;
      uddi:hasDate    "Wed Apr 23 15:06:23 BST 2003" ;
      rdf:value       mygrid:Pe577955b-d271-4a5b-8099-001abc1da633 ;
      uddi:hasAuthor  "Luc Moreau" ] .

```

**MetaData Attachment**

Figure 3: Attachment of a myGrid profile (N3 Notation)

In Figure 4, we show a semantic description of parameter *in0* declared in the interface of Figure 1. The node *rdf:\_1* denotes the message part with name *in0*. It is given a metadata attachment, with value *mygrid2:nucleotide\_sequence\_data*, which refers to a term in an ontology of bioinformatics concepts [13].

```

rdf:_3 [ a wsdl:Message ;
  wsdl:hasMessagePart
    [ rdf:_1
      [ a wsdl:MessagePart ;
        wsdl:hasName    "in0" ;
        wsdl:hasTypeName
          [ a wsdl:QName ;
            wsdl:hasNameSpace    "http://schemas.xmlsoap.org/xsd/" ;
            wsdl:hasLocalName    "string" ] ;
        uddi:hasMetadata
          [ a mygrid:semantic_type ;
            uddi:hasDate    "Fri Aug 22 11:12:29 BST 2003" ;
            rdf:value       mygrid2:nucleotide_sequence_data ;
            uddi:hasAuthor  "Luc Moreau" ] ;
          ] ;
    ] ;
]; ... ]];

```

**Semantic Description of in0  
rdf:\_1 denotes its message part**

Figure 4: Attachment of Semantic Types to Arguments (N3 Notation)

## 4 Service Directory Interfaces

We have adopted RDF triples [11] to represent all descriptions of services, which we store in a triple store [7]. We have designed and implemented a set of interfaces to this triple store in order to offer a service directory functionality. In this section, we present the methods that are relevant to metadata attachment.

The interfaces to publish metadata and discover services according to metadata were designed in a similar style to the UDDI interface, so that UDDI clients could easily be extended to support such features. As an illustration, Figure 5 shows some of the methods that allow the attachment of metadata, respectively to a business service, to a business entity and to a message part. All these methods not only attach some metadata to the respective entity, but also add some provenance information such as author and date of creation. The associated metadata can be structured or unstructured. Symmetrically, services can be discovered by using metadata filtering mechanism. An example of metadata-based search method appears in Figure 5.

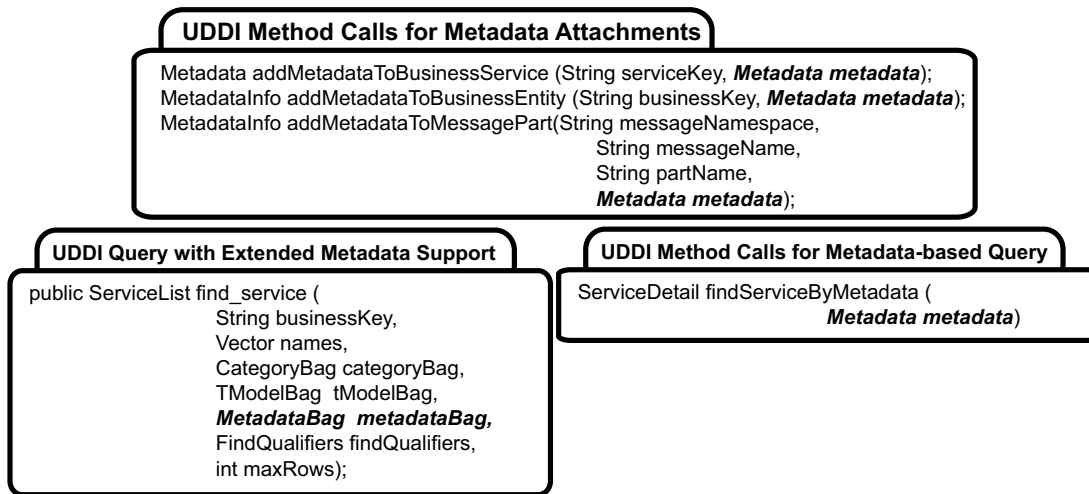


Figure 5: Metadata Attachment Methods

As all the information is represented in a triple store, a more direct interface to the triple store allows users to query the service directory using the RDQL query language [7]. An API that allows users to store triples in the triple store is also provided.

Several interfaces currently provide access to our general information model. Some of them preserve compatibility with the existing standards UDDI, and ensure interoperability within the Web Services community. Others, such as the interface to the triple store, directly expose the information model, and offer a powerful and radically different way of discovering services through the RDQL interface. While such functionality is very useful, its radically different nature does not offer a smooth transition for clients implementors wishing to adopt semantic discovery.

The benefit of our approach is the ability to extend some existing interfaces in an incremental manner, so as to facilitate an easier transition to semantic discovery for

existing clients. For instance, we have extended the UDDI `find_service` method to support queries over metadata that would have been attached to published services. In the method specification of Figure 5, `metadataBag`, a new criterion for filtering services is introduced, which contains a set of metadata that a service must satisfy.

## 5 Conclusion

In this paper, we have presented a mechanism to publish semantic descriptions about services in order to promote semantic inter-operability. Our approach relies on a metadata attachment mechanism, capable of attaching metadata to any entity within a service description. Such metadata need not be published by service providers but can be published by third-party users. Our design extends the standard interface UDDI to provide semantic capabilities, hereby offering a smooth transition to semantic discovery for UDDI clients. We have used these facilities to register service descriptions as specified by the myGrid ontology [13]. Our future work will focus on providing service descriptions to Grid services.

## 6 Acknowledgement

This research is funded in part by EPSRC myGrid project (reference GR/R67743/01) and supported by Gridnet. Keith Decker from the University of Delaware was on sabbatical stay at the University of Southampton when this work was carried out. We acknowledge Carole Goble, Phillip Lord and Chris Wroe for their comments on the paper.

## References

- [1] Tim Berners-Lee. Notation 3. <http://www.w3.org/DesignIssues/Notation3>, 1998.
- [2] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.
- [3] Paul V. Biron and Ashok Malhotra. Xml schema part 2: Datatypes. <http://www.w3.org/TR/xmlschema-2/>, May 2001.
- [4] John Colgrave and Karsten Januszewski. Using wsdl in a uddi registry (version 2.0). <http://www.oasis-open.org/committees/uddi-spec/doc/draft/uddi-spec-tn-wsdl-20030319-wd.htm>, 2003.
- [5] DAML-S Coalition:, A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara. DAML-S: Web Service Description for the Semantic Web. In *First*

*International Semantic Web Conference (ISWC) Proceedings*, pages 348–363, 2002.

- [6] Java API for XML Registries (JAXR). <http://java.sun.com/xml/jaxr/>, 2002.
- [7] Jena semantic web toolkit. <http://www.hp1.hp.com/semweb/jena.htm>.
- [8] Phillip Lord, Chris Wroe, Robert Stevens, Carole Goble, Simon Miles, Luc Moreau, Keith Decker, Terry Payne, and Juri Papay. Semantic and personalised service discovery. In *Proceedings of Workshop on Knowledge Grid and Grid Intelligence (KGGI'03), in conjunction with 2003 IEEE/WIC International Conference on Web Intelligence/Intelligent Agent Technology*, page 8 pages, Halifax, Canada, October 2003.
- [9] Simon Miles, Juri Papay, Vijay Dialani, Michael Luck, Keith Decker, Terry Payne, and Luc Moreau. Personalised grid service discovery. Technical report, University of Southampton, 2003.
- [10] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia Sycara. Importing the Semantic Web in UDDI. In *Web Services, E-Business and Semantic Web Workshop*, 2002.
- [11] Resource Description Framework (RDF). <http://www.w3.org/RDF/>, 2001.
- [12] Universal Description, Discovery and Integration of Business of the Web. [www.uddi.org](http://www.uddi.org), 2001.
- [13] Chris Wroe, Robert Stevens, Carole Goble, Angus Roberts, and Mark Greenwood. A suite of daml+oil ontologies to describe bioinformatics web services and data. *International Journal of Cooperative Information Systems*, 2003.