Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery

Dimitrios Kourtesis, Iraklis Paraskakis

South East European Research Centre (SEERC),
Research Centre of the University of Sheffield and CITY College
Mitropoleos 17, 54624 Thessaloniki, Greece
{dkourtesis, iparaskakis}@seerc.org

Abstract. UDDI registries are included as a standard offering within the product suite of any major SOA vendor, serving as the foundation for establishing design-time and run-time SOA governance. Despite the success of the UDDI specification and its rapid uptake by the industry, the capabilities of its offered service discovery facilities are rather limited. The lack of machine-understandable semantics in the technical specifications and classification schemes used for retrieving services, prevent UDDI registries from supporting fully automated and thus truly effective service discovery. This paper presents the implementation of a semantically-enhanced registry that builds on the UDDI specification and augments its service publication and discovery facilities to overcome the aforementioned limitations. The proposed solution combines the use of SAWSDL for creating semantically annotated descriptions of service interfaces and the use of OWL-DL for modelling service capabilities and for performing matchmaking via DL reasoning.

Keywords: Semantic Web Services, Web Service Discovery, Universal Description Discovery and Integration (UDDI), Semantic Annotations for WSDL (SAWSDL), Web Ontology Language (OWL)

1 Introduction

The Universal Description, Discovery and Integration (UDDI) service registry specification [1] is currently one of the core standards in the Web service technology stack and an integral part of every major SOA vendor's technology strategy. The UDDI specification defines an XML-based data model for storing descriptive information about Web services and their providers, and a Web service-based programmatic interface for publishing this type of information to the registry and performing inquiries. A UDDI service registry may be deployed and used within a private corporate network, a restricted network of business partners, or even made available over the Web. In any of the three settings, a UDDI registry can serve as the basis for establishing service lifecycle management and is one of the fundamental building blocks for realising design-time and run-time SOA governance.

A UDDI registry allows service providers to describe the functionality of their advertised services by means of references to externally maintained technical specifications or classification schemes developed by service providers or third-parties. The UDDI specification is generic and does not prescribe the use of any specific method, formal or informal, for creating such specifications. Their definition and interpretation are beyond the scope of a UDDI registry and are left to the discretion of service providers and service consumers. A UDDI service advertisement may reference numerous such descriptions in order to represent different aspects of a Web service's functional and non-functional properties. This allows services advertised in UDDI registries to be searched for and discovered by service consumers based on their declared conformance to some technical specification, or their attributed categorisation within a classification system. In addition, service advertisements can be retrieved through a text-based search for keywords contained in service names, although the effectiveness of this mechanism is clearly rather limited.

The fundamental problem with the externally maintained specifications that service advertisements refer to, like WSDL [2] documents, is that even if they are machine-processable, they still lack the formal rigour and machine-understandable semantics that would make them amenable to logic-based reasoning and automated processing. As a result, UDDI registries cannot support fine-grained matchmaking based on the actual definitions of these technical specifications or classification systems, and effectively, cannot support truly automated service discovery. In a typical service discovery scenario a developer still needs to retrieve the WSDL document and any additional specification documents referenced by a UDDI service advertisement and inspect them manually, in order to assert that the advertised service is fully interoperable with other services assembled in a service composition. Semantic Web Services research aims at addressing this problem by bringing semantics into the realm of service specifications, such that service capabilities can be explicated in an unambiguous and machine-interpretable manner that not only allows for fully automated discovery in service registries, but enables the automation of a broad range of design-time and run-time activities in SOA.

This paper presents the implementation of the FUSION Semantic Registry, a semantically-enhanced service registry developed within the FUSION project[1] and released as open source software[2]. FUSION is an EU-funded research project aiming to promote business process integration and interoperability within and across enterprises, through a semantics-based approach for Enterprise Application Integration (EAI) in service-oriented business applications. Semantically-enhanced service discovery based on widely accepted standards is an essential requirement for the theoretical and technological approach that FUSION puts forward. The FUSION Semantic Registry relies on a combination of three standards from the domain of Web service and Semantic Web technologies to achieve its objectives: UDDI, for storing and retrieving syntactic and semantic information about services and service providers, SAWSDL [3], for creating semantically annotated descriptions of service interfaces, and OWL [4], for modelling service characteristics and performing fine-grained service matchmaking via DL reasoning.

---

[1] http://www.fusion-strep.eu/
[2] http://www.seerc.org/fusion/semanticregistry/

The remaining of this paper is organised as follows. Section 2 presents the service discovery requirements that the FUSION Semantic Registry addresses, and the approach put forward in the FUSION project for describing service advertisements and service requests in a semantically-enriched manner. Section 3 presents an overview of the FUSION Semantic Registry architecture and its interfaces. Section 4 provides a walkthrough of the core activities performed during service publication, while section 5 provides a walkthrough of the activities performed during service discovery. Section 6 examines related research works that focus on the enrichment of the discovery facilities that UDDI registries offer through semantic enhancements, and section 7 concludes the paper by summarising the main points of the presented work.

2 Service Discovery Requirements in the Scope of FUSION

The FUSION project aims at delivering a reference framework, a supporting methodology, and a set of tools for realising Enterprise Application Integration (EAI) through Semantic Web Service technology. Semantically-enhanced publication and discovery of services is central to the approach that FUSION puts forward, and encompasses two objectives in order to be sufficiently supported by the FUSION Semantic Registry.

Firstly, describing service advertisements and service requests in a machine-understandable form that captures their salient characteristics and allows for comparing them in a fully automated way. Secondly, augmenting the typical functions supported by UDDI registries (i.e. storing syntactic metadata about services and their providers) with the addition of a mechanism for semantic service matchmaking and indexing.

The latter, i.e. the technical means employed to augment UDDI-based service registries with semantic matchmaking extensions is discussed in subsequent sections. This section of the paper discusses the first objective, and more specifically, describes (i) *what* are the salient service characteristics that should be captured for the purpose of matchmaking in the scope of FUSION, and (ii) *how* these characteristics should be captured in a suitable semantic representation formalism.

2.1 Service Characteristics Considered for Matchmaking

The Semantic Web Services research literature features an abundance of different approaches for service matchmaking, each of them addressing a different set of requirements and therefore focusing on a different set of service properties, functional or non-functional ones. The service characteristics that the FUSION Semantic Registry considers during matchmaking are a combination of functional and non-functional properties. In the following we describe these matchmaking requirements in detail, as a means to allow for comparisons among the FUSION Semantic Registry and other implementations or specifications of semantically-enhanced service registries in the literature.

Functional Properties. The majority of research works in the literature focus on functional properties of services, and more specifically, on approaches for matchmaking among descriptions of service inputs, outputs, preconditions and effects (IOPE). IO and PE descriptions are means to represent two different aspects of a service's functional properties: the information transformation that a service produces through the inputs it consumes and the outputs it generates, and the state-wise conditions that need to hold before a service can be invoked (preconditions), or will eventually hold after the service's invocation (effects). The matchmaking requirements that the FUSION Semantic Registry addresses concern only the first aspect, i.e. the data semantics of a Web service. Extending the capabilities of the Semantic Registry to include matchmaking based on behavioural semantics is a subject of future work, beyond the scope of the FUSION project.

More specifically, the matchmaking that the FUSION Semantic Registry performs among the inputs or outputs of a service advertisement and a service request should be able to detect if *data-level interoperability* can be guaranteed among an advertised service and its prospective consumer. This requires evaluating the degree to which the consumer provides all input data that the advertised service expects to receive when invoked, and the degree to which the advertised service produces all output data that the consumer expects to obtain after execution. As discussed in more detail in [5], this is an essential requirement for guaranteeing flawless communication among the systems participating in a collaborative business process.

From a purely practical perspective, this poses some important requirements on: (i) the expressivity of the semantic representation formalism that is employed for describing Web service inputs and outputs (it should facilitate modelling of arbitrarily complex XSD schemata), and (ii) the sensitivity of the matchmaking mechanism that is employed for comparing the above semantic representations of inputs or outputs (it should be able to detect mismatches at a fine level of granularity). As presented in detail in [5], the FUSION Semantic Registry must be able to detect mismatches among inputs or outputs of a service advertisement and a service request at two distinct levels:

- *Message-level*: The goal here is to determine the degree to which a service can produce the set of data parameters that the requestor wants to obtain, and the degree to which the requestor can provide the set of data parameters that a service expects to receive when invoked. In the case of atomic, non-transactional Web service operations, this so-called set of data parameters corresponds trivially to an operation's request or response message. In the case of complex, transactional Web services that involve the invocation of numerous operations to fulfil one goal, the set of input data parameters corresponds to the superset of all sets of input data parameters exchanged as part of request messages for the operations involved, while the equivalent holds for output data parameters. Comparing sets of input or output data parameters rather than request and response messages allows us to abstract from the differences among complex and atomic Web services and support discovery for both.
- *Schema-level:* The goal here is to determine the degree to which the schema of some data parameter that is produced or consumed by an advertised service contains all attributes specified in the schema of the corresponding data parameter

at the service consumer's end. This type of matching is meaningful in cases where an advertised service and a service consumer share a data model specification as a basis for exchanging interoperable business objects or electronic documents, but are not obliged to instantiate or make use of all schema attributes for every entity defined in that model. As a result, the case may arise where the developers of different applications have chosen to instantiate the schema attributes of a base entity (e.g. address) in different ways, thus arriving to only partially overlapping and effectively incompatible definitions of data parameters that however share a common name.

Non-functional properties. Non-functional properties also play an important role in service discovery, and are increasingly attracting the interest of the Semantic Web Services research community. Non-functional properties may relate to quality of service (QoS), policy compliance, adherence to technical standards or protocols, or categorisation within a classification scheme. The only type of non-functional property that is taken into account for matchmaking by the FUSION Semantic Registry is the latter, i.e. the categorisation of a service advertisement with regard to some semantically represented classification scheme, in order to designate the functionality of that service and assist in simple tasks like coarse-grained filtering of services during matchmaking and browsing. Extending the capabilities of the Semantic Registry to include matchmaking based on additional non-functional properties will be a subject of future work.

The end goal in the categorisation-level matching that the FUSION Semantic Registry should support, is determining if the semantic categorisation class attributed to a service request is equivalent, more specific, or more generic than the one specified in some service advertisement. In order to have a positive match, the classification concept associated with a request must subsume the classification concept of an advertisement (i.e. the first must be equivalent or more generic than the second). As an example, consider the case of a service request classified under Supply Chain Management services, and some advertisement classified under Freight Costing services, a subcategory of Transportation services that is itself classified under Supply Chain Management services. A semantically represented service taxonomy and a suitable matchmaking mechanism should allow detecting that the service advertisement can satisfy the request, since the category of Supply Chain Management services subsumes the Freight Costing services category.

2.2   Semantic Representation of Service Characteristics

By using a semantic representation formalism to express the characteristics of Web services offered or needed, providers and requestors can create definitions of service capabilities that are automatically processable through reasoning and logic-based inference. In turn, this can facilitate high-precision retrieval for services that address the matchmaking requirements presented above. Evidently, the extent to which this can be achieved depends on the semantic representation formalism that is adopted for this purpose. The recent years have seen numerous Semantic Web Service frameworks being proposed and promoted for standardisation through W3C member

submissions. The most prominent ones are OWL-S [6], WSMO [7], WSDL-S [8], and more recently the W3C Recommendation of SAWSDL [3], which evolved from the WSDL-S specification.

Although the FUSION reference framework does not prescribe the use of any specific Semantic Web Service description framework, the tools that comprise the reference implementation of the FUSION System, including the FUSION Semantic Registry, build on SAWSDL. In contrast to developing Web service descriptions at a high conceptual level and then linking these specifications to concrete Web service interfaces that are described in WSDL (as proposed in OWL-S and WSMO), the approach that SAWSDL puts forward is bottom-up: the WSDL documents themselves are to be enriched with annotations that capture machine processable semantics by pointing to concepts defined in externally maintained semantic models. This approach has numerous advantages, but the most important one is that SAWSDL can be agnostic to the knowledge representation formalism one adopts for representing service characteristics.

The semantic model that serves as the basis for creating, storing, and reasoning upon representations of service characteristics in the FUSION project is the FUSION Ontology [9]. Its multi-faceted structure reflects different types of concepts necessary for modelling a service: the data structures a service exchanges through input and output messages (data semantics), the functionality categorisation of a service with regard to a taxonomy (classification semantics), and the behaviour it may expose within a complex and stateful process execution (behavioural semantics). As already discussed, behavioural semantics are not in the range of matchmaking requirements that the FUSION Semantic Registry addresses.

The FUSION Ontology is encoded in OWL-DL, a sublanguage of the Web Ontology Language (OWL) W3C standard that has been so named due to its correspondence with description logics. OWL-DL strikes a satisfactory balance between expressiveness and computational completeness [4] and facilitates decidable reasoning with the help of DL reasoning engines. The expressivity of the DL sublanguage is a prerequisite for modelling Web service inputs and outputs at a sufficient degree of complexity that preserves the semantics of XSD schemata defined in a WSDL document. The DL sublanguage adds a number of OWL modelling constructs to those offered by OWL-Lite that are essential for this purpose, such as the ability to define enumerated classes, Boolean combinations of classes (intersectionOf, unionOf, complementOf), disjoint classes, and also place restrictions on the values that properties may have, or on their cardinality.

In order to represent the functional and non-functional service properties that are of interest for matchmaking in the FUSION Semantic Registry, one needs to create a Functional Profile, and define its key attributes in terms of references to the abovementioned FUSION Ontology. As presented in [5], a Functional Profile is expressed as a named OWL class that is attributed a set of three different OWL object properties:

1. `hasCategory`: associates a `FunctionalProfile` with exactly one `TaxonomyEntity` concept from the service classification taxonomy that is part of the FUSION Ontology, to represent the service's categorisation.
2. `hasInput`: associates a `FunctionalProfile` with an `InputDataSet` concept, in order to represent the set of data parameters that a service expects to receive and

consume. The cardinality of this property is zero in the case of an *out-only* Message Exchange Pattern (MEP), or one, in the case of an *in-out* MEP.

3. `hasOutput`: associates a `FunctionalProfile` with an `OutputDataSet` concept, in order to represent the set of data parameters that a service will produce if invoked. The cardinality of this property is zero in the case of an *in-only* MEP, or one, in the case of an *in-out* MEP.

Finally, each `InputDataSet` and `OutputDataSet` concept is associated with one or more `DataFacetEntity` concepts through a `hasDataParameter` object property, in order to represent the data parameters exchanged. Depending on the perspective from which the Functional Profile is viewed, that of the provider or the requestor, we can differentiate among Advertisement Functional Profiles (AFPs) and Request Functional Profiles (RFPs). The first are created automatically by the FUSION Semantic registry at the time of service publication, while the latter are created by the service requestor at the time of discovery (or even at an earlier stage to be used as service request templates).

To allow for the automated construction of Advertisement Functional Profiles (AFPs) in the FUSION Semantic Registry, service providers need to augment the WSDL interfaces of their provided services with semantic annotations, as per the SAWSDL specification. According to the SAWSDL annotation conventions that are applied in the context of FUSION, the semantics of a Web service's input and output data should be captured by adding `modelReference` annotations to the appropriate `<xs:element>` entities under `<wsdl:types>`, while functionality categorisation semantics should be captured via `modelReference` annotations on `<wsdl:portType>` entities.

3   FUSION Semantic Registry Architecture

A UDDI-based service registry supporting semantically-enhanced publication and discovery can be realised in a multitude of ways. A number of relevant attempts, each addressing a different set of requirements, are reviewed in the related work section of this paper. This section provides an overview of the architecture employed in the development of the FUSION Semantic Registry, and the programmatic interfaces that it exposes.

A distinctive characteristic of the FUSION Semantic Registry architecture that is discussed in this section is that it augments the purely syntactic search facilities that a UDDI registry can offer with semantic matchmaking support, without requiring any modifications to the implementation of the UDDI server or the UDDI specification API. This is considered an important advantage compared to other approaches, as it allows adopters of this solution to use their existing or preferred UDDI server implementation without performing any changes, thus encouraging uptake of such technology by the industry.

As illustrated in Figure 1, we propose an architecture where the UDDI server stands independently to the semantically-enabled service registry modules. The FUSION Semantic Registry exposes two specialised Web service APIs to the client

for publication and discovery functions, and is responsible for performing the associated SAWSDL parsing, OWL ontology processing, and DL reasoning operations. Approaches based on this principle of accommodating semantic processing functions without imposing any changes to the UDDI server implementation or interface have been also proposed in [10], [11], and [12].
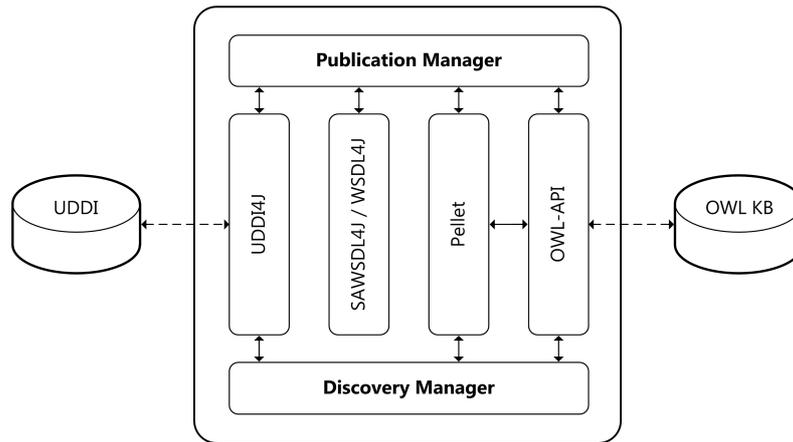

Fig. 1. FUSION Semantic Registry Architecture.

The UDDI module that is depicted in Figure 1 can be any UDDI server implementation that complies with the UDDI v2 or v3 specification as ratified by OASIS [1], although the FUSION Semantic Registry has been developed and tested using Apache jUDDI[3]. The OWL KB module is a typical OWL ontology with RDF/XML serialisation that the Semantic Registry uses for storing the Advertisement Functional Profiles it generates at the time of service publication, as will be explained in the next section of the paper. In the centre of the figure is the actual FUSION Semantic Registry, a Web Application that complies with the Java Servlet 2.4 specification and can be deployed on any compatible container implementation, such as Apache Tomcat.

The Publication Manager module of the FUSION Semantic Registry provides a Web service interface to the user for adding, removing, or updating Web service advertisements, as well as adding, removing, or updating descriptions of service providers. The Discovery Manager module provides a Web service interface for retrieving a specific service advertisement or service provider record via its key, discovering a set of services or service providers through keyword-based for terms contained in their names, and most importantly, discovering a set of services based on a Request Functional Profile. The dependencies that these two manager modules have on the third-party components that are depicted in the centre of the figure are examined in the following sections, along with the overviews of the semantic service publication and discovery processes.

---

[3] http://ws.apache.org/juddi/

4 The Publication Process

As already mentioned, the Publication Manager Module provides a Web service interface to the user for adding, removing, or updating descriptions of Web services, as well as adding, removing, or updating descriptions of service providers. This section of the paper focuses on the most important of these functions, the process of publishing a semantically-enhanced service description.

The publication query that initiates the publication process comprises: (i) the service provider ID (every service advertisement is associated to exactly one service provider that is identified by a UUID key), (ii) a URL pointing to the SAWSDL document that describes the service, (iii) an optional service name, and (iv) an optional free text description. The process that follows based on this input comprises a number of phases that are presented in the following subsections.

4.1 Parsing of SAWSDL Document

The first step that the Publication Manager performs is to retrieve the SAWSDL document from the specified URL and parse it to extract the semantic annotations it contains. As discussed in section 2, WSDL interfaces are augmented with potentially multiple `modelReference` annotations on `<xs:element>` entities, in order to capture the data semantics of the service (consumed inputs or produced outputs), and a single `modelReference` annotation on `<wsdl:portType>` entities to capture its functionality categorisation semantics. At the time of this writing the current implementation of the Semantic Registry SAWSDL parser relies on the WSDL4J[4] and SAWSDL4J[5] libraries to create an in-memory representation of the SAWSDL document and extract the URIs of the ontological concepts to which the `modelReference` annotations point.

4.2 Construction of UDDI Advertisement

The next step in the publication process is to map the information that was provided as part of the publication query (i.e. the service name, free text description, and service provider's UUID) and the information that was extracted by parsing the SAWSDL document (i.e. input, output, and category annotation URIs), into a UDDI service advertisement. Communication between the FUSION Semantic Registry and the UDDI server for this purpose is facilitated by UDDI4J[6].

As illustrated in Figure 2, this mapping requires creating a `uddi:businessService` entity and instantiating the values of its `uddi:name`, `uddi:description`, and `uddi:businessKey` attributes, as well as a `uddi:categoryBag` that includes one `uddi:keyedReference` entity for every extracted annotation URI. The FUSION Semantic Registry makes use of so-called canonical tModels for representing the different types of semantic annotations that can be placed on SAWSDL documents

---

[4] http://sourceforge.net/projects/wsdl4j
[5] http://knoesis.wright.edu/opensource/sawsdl4j/
[6] http://uddi4j.sourceforge.net/

(input, output, or category annotations). Depending on the type of semantic information being modelled, each `uddi:keyedReference` entity should point to the appropriate canonical tModel (Input Annotation tModel, Output Annotation tModel, or Category Annotation tModel). As depicted in Figure 2, an additional canonical tModel is used for indexing service advertisements with respect to the Request Functional Profiles that they can readily satisfy (Semantic Indexing tModel), but the `uddi:keyedReference` entities which point to this tModel are created at a later stage in the publication process.
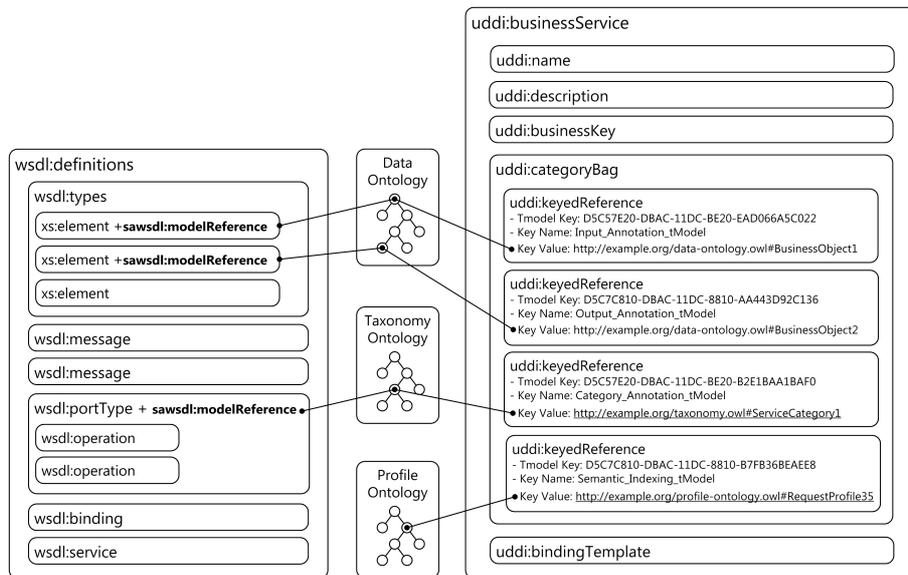

Fig. 2. SAWSDL to UDDI mapping methodology

4.3 Generation of Functional Profile and Publication-time Matchmaking

The next step in the process is to create an Advertisement Functional Profile (AFP) based on the extracted semantic annotations and add it to the registry's internal OWL Knowledge Base (KB) with the help of the OWL API library[7]. The construction of the AFP follows the modelling conventions analysed in section 2. Once the AFP has been constructed, the Pellet DL reasoner[8] is used for performing an "eager" semantic classification of the new AFP against all known Request Functional Profiles (RFPs). The purpose of this classification procedure is to identify RFPs representing service requests that the newly added service advertisement can readily satisfy.

---

[7] http://owlapi.sourceforge.net/
[8] http://pellet.owldl.com/

We refer to this classification procedure as "eager" since it takes place at publication-time. In contrast, a "lazy" classification procedure would not have taken place before the actual need for matchmaking arises during discovery-time. This approach may be placing an overhead on the time required to complete the publication of a service advertisement, but it substantially reduces the time required to perform matchmaking at discovery-time, so it is considered particularly beneficial.

Three conditions must hold in order to claim that the new service advertisement can satisfy a service request: (i) the `InputDataSet` concept associated with the RFP must be subsumed by the `InputDataSet` of the AFP, (ii) the `OutputDataSet` of the RFP must subsume the `OutputDataSet` of the AFP, and (iii) the `TaxonomyEntity` concept associated with the RFP must subsume the `TaxonomyEntity` of the AFP.

4.4 Indexing of Semantic Matching Results in UDDI

The last step in the publication process is to map the semantic matchmaking information that resulted from the publication-time matchmaking algorithm described above into the UDDI service advertisement. This requires retrieving the advertised `uddi:businessService` entity and its associated `uddi:categoryBag` from the UDDI server, and creating one `uddi:keyedReference` for every RFP that the service matches with. What this essentially achieves is indexing the service advertisement with respect to all service requests it can readily satisfy. As depicted in Figure 2, `uddi:keyedReference` entities should be made to point to the canonical tModel used for this purpose (the Semantic Indexing tModel), and the URI of each RFP should be specified as the Key Value of the `uddi:keyedReference`. When this step is completed, a new semantic service advertisement has been created, registered with the UDDI registry, and is available for discovery.

5 The Discovery Process

The Discovery Manager module provides a Web service interface for retrieving service advertisements or service provider records via their unique keys, discovering sets of services or service provider records through keyword-based search, and most importantly, discovering sets of services based on a Request Functional Profile that represents the requirements of the service consumer. This latter type of semantic matchmaking functionality is the focus of this section.

The discovery query that initiates the semantic matchmaking process comprises two elements: (i) a URI pointing to some Request Functional Profile (RFP), and (ii) an optional UUID designating the preferred service provider, i.e. the company, business unit, or specific business application that should expose the service. The RFP that the URI points to may be defined within an ontology that is shared by service providers and service requestors alike (i.e. be a reusable RFP defined in the FUSION Ontology), or within some third-party ontology that imports and extends the shared ontology (i.e. be a custom-built and non-shared RFP). Depending on which of the two cases holds, the algorithm would follow a different discovery path. Resolving the

location of the ontology in which the RFP is identified is therefore the first step in the discovery process.

If the RFP is defined in the shared FUSION Ontology the Discovery Manager will look for service advertisements indexed in UDDI with a reference to that RFP. This means looking for services with AFPs that have matched the requested RFP during the "eager" publication-time classification. To retrieve such advertisements the Discovery Manager places a simple syntactic matchmaking query to the UDDI server, looking for `uddi:businessService` entities having a `uddi:categoryBag` that contains a `uddi:keyedReference` which points to the Semantic Indexing tModel, and moreover, has a Key Value that is equal to the URI of the RFP.

Since the matchmaking and indexing process is repeated every time a new RFP is created and added to the shared ontology, the UDDI server's semantic matching index is bound to always be accurate and up to date. This means that if some service advertisement matches some RFP which is defined in the shared ontology, the registry is guaranteed to have this association indexed in the UDDI server, and be able to instantly retrieve the advertised service.

Due to the shared ontology assumption that is valid in the context of FUSION, this is the most typical type of discovery querying envisaged for the FUSION Semantic Registry, and is also the simplest and fastest type of matchmaking possible. Since the time-consuming process of subsumption reasoning and hierarchy classification has been already performed at publication-time, the computational complexity of discovery-time matchmaking for RFPs defined in a shared ontology is essentially as low as that of a conventional UDDI server. In other words, the use of semantics does not impose any noteworthy overhead compared to syntactic matchmaking.

If the RFP is defined in a non-shared ontology the Discovery Manager would need to load that ontology into memory and perform a complete semantic matchmaking process among the specified RFP and all AFPs stored in the OWL-KB. The conditions that need to be checked in order to assert that a service advertisement can satisfy the request are the same as the ones defined for publication-time matchmaking (see section 4.3).

The result of the discovery process, regardless of the ontology in which the RFP is defined, is a list of UUID keys corresponding to advertisements of services that comply with the matchmaking criteria modelled in the RFP. If a service provider UUID has been also specified in the discovery query, the UDDI server will restrict the result set to only those services offered by the specified provider.

6   Related Work

The use of semantics for representing service characteristics and facilitating semantically-enhanced matchmaking in UDDI-based registries has been the focus of numerous works in recent years. Largely due to the fact that OWL-S was the first Semantic Web Service framework to be developed, most of the relevant approaches in the research literature rely on OWL-S.

In [13] the authors propose that discovery in UDDI registries should be achieved through semantic matchmaking among service capability descriptions expressed as

DAML-S Profiles. To that end, they propose a matchmaking algorithm that can recognise various degrees of match among the inputs and outputs of advertisements and requests described in DAML-S, and also propose the incorporation of a matching engine inside the UDDI registry. In [14] the authors expand on the work introduced in [13] and define a mapping among DAML-S Profiles and UDDI data structures, such that semantic information can be recorded in UDDI. Subsequent work in [15] proposes a revised mapping between OWL-S Profiles and the UDDI data model, and also an improved version of the matchmaking algorithm from [13].

A research work by a different group that expands on the approach firstly introduced in [13] and [14] is presented in [16]. The authors in [16] present a method to improve the effectiveness of service discovery in UDDI based on a two-stage service discovery process, combining syntactic and semantic search for service inputs and outputs. They also propose an extension to the specification of the UDDI inquiry API in order to support automatic service composition based on DAML-S semantics.

Another approach for developing OWL-S-based semantically-extended UDDI registries is presented in [12]. The key feature of the proposed solution is that hierarchical relationships among ontology concepts are resolved at the time of publication and indexed in UDDI in a way that allows processing in a purely syntactic manner at the time of discovery. The modules for publishing and query processing are placed on the client-side and as a result no modifications to the UDDI server implementation or interface are mandated.

An approach that utilises the WSDL-S specification is introduced in [17] and elaborated in [18]. In [17] the authors present a theoretical approach for publishing Web service descriptions that have been semantically annotated with references to concepts defined in an OWL ontology. Their proposed approach includes a WSDL-S to UDDI mapping for storing these semantic annotations, and facilitating subsequent discovery of Web service operations based upon them. In a subsequent work [18] the authors describe the way in which Web service descriptions can be annotated, published and discovered using Radiant and Lumina, a pair of graphical tools that are integrated with the METEOR-S Web Services Discovery Infrastructure (MWSDI). Discovery with the proposed system is performed based on a semantic request template that specifies the desired functionality, inputs, and outputs, by references to ontological concepts. A number of research prototypes that support the WSDL-S specification and were developed in the context of METEOR-S are currently undergoing a process of harmonisation with the SAWSDL standard, but it seems that a UDDI-based service registry supporting the standard has not yet been realised.

In contrast to our matchmaking desiderata, the above described approaches do not seem to consider the problem of schema-level mismatch among inputs and outputs as an important use case, and thus do not seem to address the need for matchmaking at a fine level of granularity. With the exception of [17] and [18], classification-based matchmaking is not addressed either. Another distinction among the approach that the FUSION Semantic Registry puts forward and the other reviewed approaches, with the exception of [12], is in the proposed architecture for incorporating semantic matchmaking capabilities in UDDI. While most of the approaches necessitate some form of modification to the UDDI server's programmatic interface or internal business logic, the approach that we suggest in this paper does not mandate any such changes.

7 Conclusions

Despite its indisputable success and wide-spread adoption by the industry, the UDDI specification features a service discovery mechanism with some important limitations. The services advertised in a UDDI registry are currently being described and discovered by means of references to externally maintained technical specifications or classification schemes that lack the machine-understandable semantics that would be necessary to support fully-automated service discovery. This paper presents the implementation of the FUSION Semantic Registry, a semantically-enhanced service registry that builds on the UDDI specification and augments its service publication and discovery facilities in order to address this challenge and meet the requirements that the FUSION project puts forward for service discovery. We have presented a theoretical and technological solution that relies on a combination of three standards from the domain of Web service and Semantic Web technologies to achieve its objectives: UDDI, for storing and retrieving syntactic and semantic information about services and service providers, SAWSDL, for creating semantically annotated descriptions of service interfaces, and OWL-DL, for modelling service characteristics and performing fine-grained service matchmaking via DL reasoning. To the best of our knowledge the presented work represents the first attempt to combine these standards into a comprehensive and self-contained system. The proposed solution has been fully implemented and tested within the FUSION project, and is contributed to the community as open source software.

References

1. Clement, L., Hately, A., von Riegen, C., Rogers, T. (eds.).: UDDI Version 3.0.2 Specification, UDDI Specification Committee (October 2004)
2. Chinnici, R., Moreau, J.J., Ryman, A., Weerawarana, S. (eds.).: Web Services Description Language (WSDL) Version 2.0. W3C Recommendation (June 2007)
3. Farrell, J., Lausen, H. (eds.).: Semantic Annotations for WSDL and XML Schema. W3C Recommendation (August 2007)
4. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview, W3C Recommendation (February 2004)
5. Kourtesis, D., Paraskakis, I.: Web Service Discovery in the FUSION Semantic Registry. In: Abramowicz, W., Fensel, D. (eds.) BIS 2008, LNBIP 7, Springer, Heidelberg (2008)

6. 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.: OWL-S: Semantic Markup for Web Services. W3C Member Submission (November 2004)
7. Bruijn, J.d., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., Kifer, M., Konig-Ries, B., Kopecky, J., Lara, R., Lausen, H., Oren, E., Polleres, A., Roman, D., Scicluna, J., Stollberg, M.: Web Service Modeling Ontology (WSMO). W3C Member Submission (June 2005)
8. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.T., Sheth, A., Verma, K.: Web Service Semantics - WSDL-S. W3C Member Submission (November 2005)
9. Bouras, A., Gouvas, P., Mentzas, G.: ENIO: An Enterprise Application Integration Ontology. In: 1st International Workshop on Semantic Web Architectures For Enterprises, 18th International Conference on Database and Expert Systems Applications, Regensburg, Germany, September 3-7 (2007)
10. Colgrave, J., Akkiraju, R., Goodwin, R.: External Matching in UDDI. In Proceedings of the 2004 IEEE International Conference on Web Services (ICWS'04), San Diego, USA, July 2004.
11. Pokraev, S., Koolwaaij, J., Wibbels, W.: Extending UDDI with Context Aware Features based on Semantic Service Descriptions. In Proceedings of the 2003 International Conference on Web Services (ICWS'03), Las Vegas, USA, June 2003.
12. Luo, J., Montrose, B., Kim, A., Khashnobish, A., Kang, M.: Adding OWL-S Support to the Existing UDDI Infrastructure. In Proceedings of the 2006 IEEE International Conference on Web Services (ICWS'06), Chicago, USA, September 2006. ISBN 0-7695-2669-1.
13. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic Matching of Web Service Capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, Springer, Heidelberg (2002)
14. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Importing the Semantic Web in UDDI. In: Proceedings of Web Services, E-Business and Semantic Web Workshop, Toronto, Canada, May 2002, pp. 225–236 (2002)
15. Srinivasan, N., Paolucci, M., Sycara, K.: Adding OWL-S to UDDI, Implementation and Throughput. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, Springer, Heidelberg (2005)
16. Akkiraju, R., Goodwin, R., Doshi, P., Roeder, S.: A method for semantically enhancing the service discovery capabilities of UDDI. In: Proceedings of the Workshop on Information Integration on the Web (IIWeb 2003), Acapulco, Mexico (August 2003)
17. Sivashanmugam, K., Verma, K., Sheth, A., Miller, J.: Adding Semantics to Web Services Standards. In: Proceedings of the 2003 International Conference on Web Services (ICWS 2003), Las Vegas, USA (June 2003)
18. Li, K., Verma, K., Mulye, R., Rabbani, R., Miller, J., Sheth, A.: Designing Semantic Web Processes: The WSDL-S Approach. In: Cardoso, J., Sheth, A. (eds.) Semantic Web Services, Processes and Applications, pp. 163–198. Springer, Heidelberg (2006)