A Semantic Web based Peer-to-Peer Service Registry Network

Uwe Thaden, Wolf Siberski, Wolfgang Nejdl

***Abstract*—While Web Services already provide distributed operation execution, the registration and discovery with UDDI is still based on a centralized design. In this paper we show to build a distributed discovery service, based on a peer-to-peer infrastructure. Furthermore, we use DAML-S service descriptions to provide enhanced semantic search capabilities. Our prototype implementing this approach based on the peer-to-peer infrastructure Edutella is described.**

***Index Terms*—Semantic Web Services, peer-to-peer networks, Web Service registry.**

I. Introduction and Motivation

TODAY distributed computing is connected with the idea of Web Services. These are (parts of) programs that can be accessed over a network using well defined protocols. An important aspect is that the interactions should be done automatically by computers. Currently one of the main problems is the locating of Web Services which provide the desired functionality.

This paper is divided as follows. In section II we give an overview of UDDI and WSDL. In section III we describe our idea and an example of a peer-to-peer and semantic web based registry for Web Service descriptions. We show concrete parts of our prototype that we implemented based on our Edutella-infrastructure in section IV.

II. Background: UDDI and WSDL

A. Running Example

Throughout this paper we will use as example Web Services for image downloads. MD Smith is a medical instructor. He wants to prepare a lecture about leukemia. For this purpose he needs images of leukocytes (white blood cells). He wants to include a diagram showing how the leukocytes are structured. He is a bit peculiar about the image quality, and therefore insists of having the diagram as vector graphic.

Today there are probably hundreds of medical image databases available via internet, many of them owned by universities. Besides, the graphics needed could also be provided by general image providers, e. g. press agencies or commercial image archives. Currently most of them are accessible via HTML user interface only, but for our example we assume that access to these image archives via Web Services is already provided.

B. UDDI Design and Infrastructure

To archive Web Service integration, one must be able to locate specified services. The typical process is shown in figure 1. In a first step a Web Service provider publishes his offerings (publish). After that clients can search for a Web Service (find) and will retrieve information how to use the provider that offer the asked functionality, so it can connect to and use the Web Service (bind).

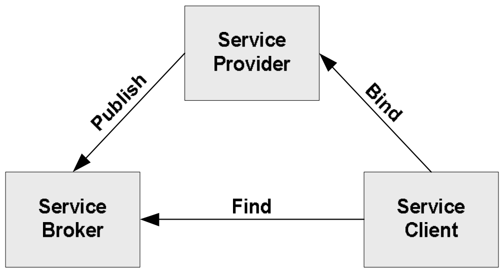

Fig. 1. Web Service registration and discovery

Universal Description, Discovery and Integration (UDDI) is a specification for business registries from Ariba, IBM and Microsoft [1]. It is a central (possibly replicated) registry which contains information about businesses and their provided Web Services (private registries are possible, but lead to drawbacks as discussed in the next section). Centralized approaches combined with replication have many drawbacks, e. g. poor scalability and less consistency on large registries. UDDI can be seen as typical yellow pages, pointing to registered Web Services which can be located elsewhere.

UDDI is a registry; it does not specify any kind of registry to manage models or even schemas or metadata. Thus, relationship information cannot be provided.

An item in a UDDI registry consists of the following parts:

- businessEntity: Describes a business or other organization that typically provides Web Services.
- businessService: Describes a collection of related

Manuscript received February 17, 2003.

All Authors are with Learning Lab Lower Saxony, University of Hanover, Expo Plaza 1, D-30539, Hanover, Germany.

{thaden, siberski, nejdl}@learninglab.de.

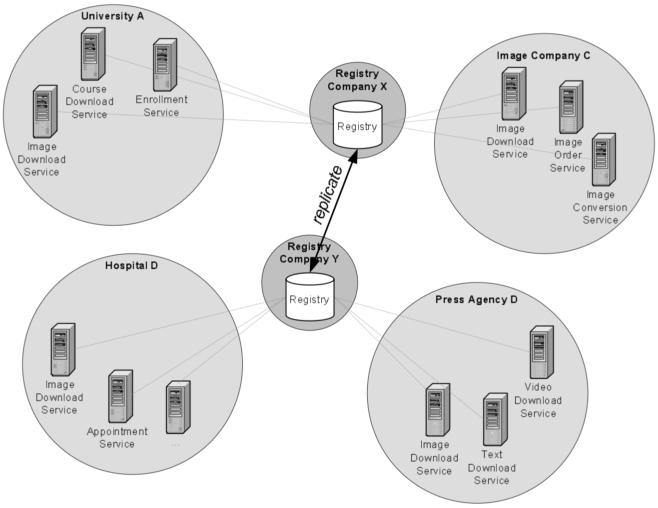

Fig. 2. Services registered at central registries

Web Services offered by an organization described by a businessEntity.

- bindingTemplate: Describes the technical information necessary to use a particular Web Service.
- tModel: Describes a "technical model" representing a reusable concept, such as a Web Service type, a protocol used by Web Services, or a category system.
- publisherAssertion: Describes, in the view of one businessEntity, the relationship that the businessEntity has with another businessEntity.
- subscription: Describes a standing request to keep track of changes to the entities described by the subscription.

Only tModels enable a search in the UDDI-registry that comes near to our idea (however, one will barely search for a Web Service based on the providing company).

For putting in a new Web Service one has to know the exisiting tModels. Furthermore any kind of relation between tModels cannot be expressed; a new tModel can't be described as a new version of an existing one. Thus, any hierarchy is impossible.

Furthermore, the search facility is limited; one can search by keywords, but cannot ask for "something similar" since UDDI does not provide a vocabulary (ontology). The tModels as well as the WSDL descriptions are not stored in the UDDI registry, only the names and some describing keywords.

Following the UDDI-specification it is provided to

1. Find Web Services implementations that are based on a common abstract interface definition.
2. Find Web Services providers that are classified according to a known classification scheme or identifier system.

A tModel represents an abstract service type (a generic representation of a registered service) in the UDDI registry. Each Web Service is categorized according to a defined list of service types.

Each tModel has a name, an explanatory description and an unique identifier. The tModel name identifies the service, e. g. "Online order placement". The description gives additional information, e. g. "Place an order online".

In our example the user wants to find a Web Service that allows downloading images in vector format. Using a UDDI-tModel only allows describing the first part of that, but it is not possible to describe that a Web Service should have a vector graphic output.

C. WSDL Service Description

The services registered with UDDI are described in the Web Service Description Language (WSDL, [2]). This language allows describing all technical information needed to access the service, including its operations with parameters and results. For example, an image download service could be described as follows:

```
<definitions name="MyImageArchive"
targetNamespace="http://example.org/wsdl/MyImageArchive">
…
  <message name="ImageArchive_search">
    <part name="keyword" type="xsd:string"/>
    <part name="imageType" type="xsd:string"/>
  </message>
  <message name="ImageArchive_searchResponse">
    <part name="image" type="xsd: base64Binary" "/>
  </message>
  </message>
[…]
  <portType name="ImageArchive">
    <operation name="search"
      parameterOrder="keyword imageType">
      <input message="tns:ImageArchive_search"/>
      <output message="tns:ImageArchive_searchResponse"/>
    </operation>
  </portType>
[…]
  <service name="MyImageArchive">
    <port name="ImageArchivePort"
      binding="tns:ImageArchiveBinding">
      <soap:address
        location="http://example.org/imageservice:9765"/>
    </port>
  </service>
</definitions>
```

WSDL perfectly describes how a service is called, and thus (in combination with SOAP) facilitates platform and programming language independent remote operation execution. However, WSDL is less suitable to describe what a service does. For example, while we can express that the service returns binary data in base64 format, we can't state that this data is in fact the representation of an image.

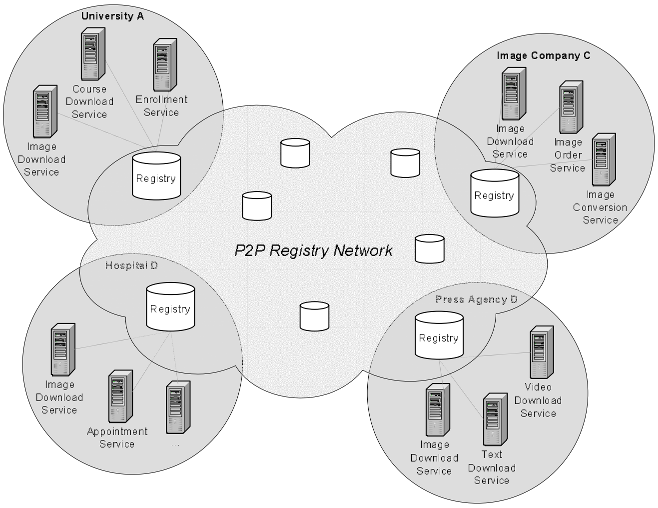

Fig. 3. Peer-to-peer registry network

III. Peer-to-Peer and DAML-S

A. *Connecting Registries with Peer-to-Peer Technology*

While the idea of UDDI is a centralized storage of Web Service descriptions, currently most providers of Web Services tend to set up their own so called private registries. Several commercial UDDI registry products are available which help businesses to do that. A new trend is to integrate UDDI registry features in more general company directories (e. g. Microsoft Active Directory, Novell eDirectory [3]). This will further enforce the trend to company-wide UDDI registries and limit the usage of central public registries.

This situation makes it difficult to discover services. Technically, it would be possible to replicate registry information from all private registries to the public central nodes. However, this needs a replication contract between both registry providers, and manual system administration for each new private registry. Therefore, while technically possible, practically replication from private to public UDDI registries doesn't occur.

To alleviate this problem, we propose moving from a central design to an (de facto already existing) distributed approach by connecting private registries with peer-to-peer technology. The power of the internet comes topmost from moving from centralized solutions to non-structured information based at each computer connected to the net. Using peer-to-peer architecture follows this idea and adds value by creating a virtual global registry from all connected local registries [4].

Peer-to-peer enables the Web Service registries. Thus, companies as well as universities can build their own Web Service registries which are maintained by themselves. Being a peer in a P2P-network makes it easy to search all local registries.

The idea of a P2P network of UDDI service registries has already been mentioned in [5]. [6] have proposed to combine Web Services, peer-to-peer and Semantic Web technology to build the next-generation service driven systems. An ontology-based peer-to-peer topology suitable for service discovery has been described in [7].

Speed-R is a project at University of Georgia that developed a distributed registry based on the current UDDI without semantically enriched Web Service description [8]. While we envision a registry partition where institutions maintain their private registries, the Speed-R system assumes a partition based on business domains. The most important difference is that we use DAML-S as service description language instead of UDDI tModels. This is the prerequisite for enhanced semantic search capabilities, as described in the following section.

B. *Service Description and Discovery with DAML-S*

To support discovery requests beyond a yellow pages-lookup style, semantic matching capabilities are a prerequisite. For our purpose we use DAML-S [9] to describe the provided functionality of a Web Service. A DAML-S description always consists of four parts as shown in figure below.

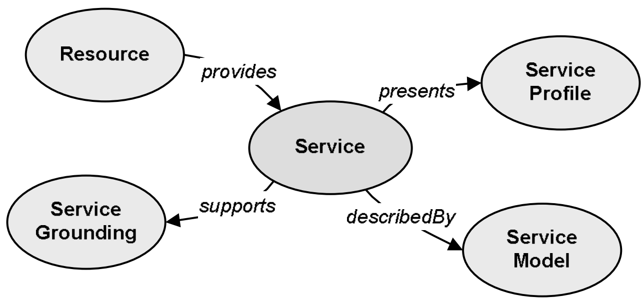

Fig. 4. DAML-S service description overview

A resource (here a program (-part)) offers a service, thus our Web Service. This service is described by three parts:

- Service Profile(s)
  The specification of what a service requires and provides for an external („black-box") view of the service, e. g. required inputs.
- Service Model(s)
  The specification of how a service works for an internal („glass-box") view of the service, e. g. steps for a transaction.
- Service Grounding(s)
  Implementation specific details (e. g. message formatting).

[10] has shown that DAML-S can be seen as an extension of WSDL: It is possible to map a WSDL description to DAML-S, and provide information not expressible in pure WSDL as add-on. In our prototype we therefore use DAML-S to describe services. However the standardized Web Service

architecture is used to finally interact with a service.

In our example we use an ontology describing different types of Web Services. We have a super class defining a service at all. Subclasses are “ImageService”, subdivided into “DeliverByMailImageService” and “Downloadable-ImageService”. For example, the latter one describes all Web Services that can be used somehow to receive URLs of images to download (satisfying given criteria like output URL must point to a vector image).

```
[…]
<rdfs:Class
   rdf:about="&serviceType;DownloadableImageService">
   <rdfs:subClassOf
      rdf:resource="&serviceType;ImageService"/>
   <rdfs:label>Downloadable Image Service</rdfs:label>
</rdfs:Class>
[…]
```

For the Web Service description it’s enough to write down the service profile since it holds all necessary information like Web Service description, inputs, and outputs. The interesting parts are shown below:

```
 […]
<serviceType:DownloadableImageService
   rdf:about="&example;MyImageArchive">
  <profile:input>
    <profile:ParameterDescription
      rdf:about="&example;Keyword">
     <profile:parameterName>
         keyword
     </profile:parameterName>
     [...]
    </profile:ParameterDescription>
  </profile:input>
  <profile:input>
    <profile:ParameterDescription
      rdf:about="&example;ImageType">
    <profile:parameterName>
        image type
     </profile:parameterName>
     [...]
    </profile:ParameterDescription>
  </profile:input>
  <profile:output>
    <profile:ParameterDescription
      rdf:about="&example;Image">
      <profile:parameterName>
        ImageFound
     </profile:parameterName>
    <profile:restrictedTo rdf:resource="&image;VectorImage" />
   </profile:ParameterDescription>
  </profile:output>
</serviceType:DownloadableImageService>
[…]
```

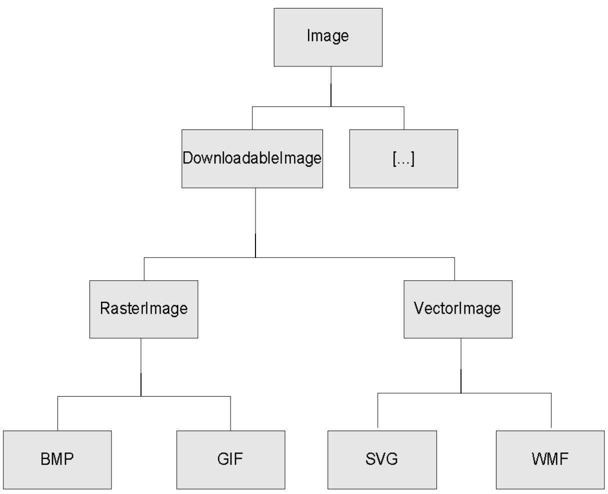

Fig. 5. Image ontology

Based on this kind of description, it is possible to search for all services of type DownloadableImageService which have an output of type VectorImage. Such a query wouldn’t have been possible with tModels only, because it isn’t possible to characterize parameter and result types (except assigning XML Schema types).

An effective matching algorithm for DAML-S between service descriptions and requests has been developed [16].

C. Implementation

For our implementation we reuse the existing project Edutella. Edutella is a general peer-to-peer infrastructure for storing, querying and exchanging metadata [11]. It is built on the open source project JXTA, a framework which provides basic peer-to-peer network features [12]. Edutella can connect highly heterogeneous peers (heterogeneous in their uptime, performance, storage size, functionality, number of users etc.). To achieve the desired interoperability, it is crucial to adhere to standards [13]. Therefore Edutella is based on metadata standards defined by the SemanticWeb initiative of the WWW Consortium [14], namely RDF and RDFS. Each Edutella peer can make its metadata information available as a set of RDF statements, suitable for describing distributed resources.

This information can be queried using the Edutella query exchange language (QEL). QEL defines several capability levels, starting with simple conjunctive queries (which allow a query-by-example style of request) up to capabilities comparable to query languages of state-of-the-art relational databases. QEL queries are distributed to all relevant peers (see [15] for the query distribution mechanism), and matches sent back to the query originator. We have implemented wrappers to several RDF stores which translate the QEL queries to the backend-specific query language, and can therefore integrate various kinds of backends into the network, from files to RDBMS systems and inferencing databases.

As DAML-S descriptions are already provided in RDF format, Edutella is perfectly suited to provide them. To search for services, a peer sends an appropriate QEL query to the network.

For simple queries, QEL provides a query-by-example syntax, where arbitrary nodes in the RDF graph are tagged as variables. The discovery query for all services of type DownloadableImageService which have an output of type VectorImage looks as follows:

```
<edu:QEL1Query rdf:about="#serviceQuery">
   <edu:hasVariable rdf:resource="#Service"/>
   <edu:hasVariable rdf:resource="#Output"/>
   <edu:hasResultType rdf:resource="&edu;TupleResult"/>
</edu:QEL1Query>

<edu:Variable rdf:about="#Service"/>
<edu:Variable rdf:about="#Output"/>

<serviceType:DownloadService rdf:about="#Service">
  <rdf:type rdf:resource="&serviceType;DownloadService"/>
   <profile:output>
    <profile:ParameterDescription rdf:about="#Output">
    <profile:restrictedTo rdf:resource="&image;VectorImage" />
    </profile:ParameterDescription>
   </profile:output>
</serviceType:DownloadService>
```

As described, such a query can be processed by already existing peer implementations. For a complete implementation we only had to add a registration facility for new or changed services. Services have to provide their DAML-S description when registering. This information is stored as RDF and can be accessed using the default query mechanism. The WSDL description is stored as well; service clients can use it to access discovered services via SOAP.

Our prototype user interface for discovery allows searching for services by selecting concepts from several ontologies. For services found a simple user interface is generated, using the WSDL information, and the service can be called manually.

IV. Conclusion and Further Work

We have shown that introducing peer-to-peer technology for Web Service registries provides the advantages of easy setup and maintenance of local registries while preserving the ability for global service discovery. This is achieved by distributing discovery requests to all registry peers within the network. By introducing DAML-S service profile descriptions in combination with ontologies we gain enhanced semantic query capabilities.

It would be interesting to combine our discovery mechanism with the work done by Hendler et al. to create composite Web Services [17]. Service elements could be found using the registry network and then plugged-in into the designed composite service.

Currently only discovery is done via the peer-to-peer network. The next logical step would be to let the network also handle the service binding and execute the services on behalf of the query originator. Thus, in our example, a client would send the service parameters together with the service discovery request, and receive the images directly instead of service descriptions.

Acknowledgement

Stefan Decker proposed the initial idea for distributed Web Services in one of our discussions, which finally led to the work described in this paper.

References

[1] *"The UDDI Specification V3"*, http://www.uddi.org

[2] Roberto Chinnici, Martin Gudgin, Jean-Jacques Moreau, Sanjiva Weerawarana, *Web Services Description Language (WSDL) Version 1.2,* http://www.w3.org/TR/wsdl12/

[3] Cathleen Moore. *"Novell rolls out UDDI server"*. Infoworld article. http://www.infoworld.com/article/02/12/11/021211hnnoveluddi_1.html

[4] Andy Oram (Ed.), *"Peer-to-Peer: Harnessing the Power of Disruptive Technologies"*, O'Reilly, 2001

[5] Graham Glass. *Web Services.* Prentice Hall, 2002

[6] A. Maedche, S. Staab, *"Services on the move – Towards P2P-Enabled Semantic Web Services"*, Tenth International Conference on Information Technology and Travel & Tourism, ENTER 2003, Helsinki, 2003.

[7] Mario Schlosser, Michael Sintek, Stefan Decker, Wolfgang Nejdl, "*HyperCuP - Hypercubes, Ontologies and Efficient Search on P2P Networks"*, International Workshop on Agents and Peer-to-Peer Coputing, Bologna, Italy, July 2002

[8] Kaarthik Sivashanmugam, Kunal Verma, Ranjit Mulye, Zhenyu Zhong. "*Speed-R: Semantic P2P Environment for diverse Web Services Registries"*. http://webster.cs.uga.edu/~mulye/SemEnt/final.html

[9] DAML-S Coalition, *"DAML-S: Web Service Description for the Semantic Web"*, First International Semantic Web Conference, Sardinia, Italy, 2002.

[10] M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara. *"Importing the Semantic Web in UDDI"*. Workshop on Web Services, e-Business, and the Semantic Web (WES). Fourteenth International Conference on Advanced Information Systems Engineering, Toronto, Canada, 2002.

[11] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M.Nilsson, M. Palmér_ and T. Risch, "Edutella*: A P2P Networking Infrastructure Based on RDF"*. Eleventh International World Wide Web Conference (WWW2002). Available at: *http://edutella.jxta.org/reports/edutella-whitepaper.pdf.*

[12] L. Gong, *"Project JXTA: A Technology Overview"*, Sun Microsystems, 2001. Available at *http://www.jxta.org/project/www/docs/TechOverview.pdf.*

[13] R. Dornfest, and D. Brickley, *"The Power of Metadata"*, in *Peer-to-Peer: Harnessing the Power of Disruptive Technologies* [4]. Available at *http://www.openp2p.com/pub/a/p2p/2001/01/18/metadata.html*

[14] T. Berners-Lee, J. Hendler, O. Lassila, *"The Semantic Web"*, Scientific American, May 2001. Available at *http://www.sciam.com/2001/0501issue/0501berners-lee.html.*

[15] Nejdl, Wolfgang; Wolpers, Martin; Siberski, Wolf; Schmitz, Christoph; Schlosser, Mario; Brunkhorst, Ingo; Löser, Alexander. *"Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks"*. Accepted for Publication at *12th International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.

[16] M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara. *"Semantic Matching of Web Services Capabilities"*. First International Semantic Web Conference, Sardinia, Italy, 2002.

[17] E. Sirin, J. Hendler, B. Parsia, *„Semi-automatic Composition of Web Services using Semantic Descriptions"*. .Accepted to *Web Services: Modeling, Architecture and Infrastructure* workshop in conjunction with ICEIS2003.