

UDDI and WS-Inspection

José M. Vidal

Sat Apr 3 15:14:56 EST 2004

We describe the Universal Description, Discovery, and Integration project. This talk is based on:

- James Snell, Doug Tidwell, and Pavel Kulchenko. Programming Web Services with SOAP¹. 2002. Chapter 6
- Tyler Jewell and Dave Chappell UDDI: Universal Description, Discovery, and Integration Part 1²

1 Introduction

- Once everyone posts the WSDL description of their service, how do you find the service you need?
- The Universal Description, Discovery, and Integration of Business for the Web³ project is an industry effort to define a registry where users can automatically discover the services they need (and rent them).
- They publish the official UDDI Specification⁴.
- UDDI has two parts
 - A registry of all the web service's metadata.
 - A set of WSDL port type definition for manipulating and searching the registry.

2 UDDI Registry

- Allows a business to publicly list a description of itself and the services it provides.
- Customers will go to it and locate what they want based on taxonomical information.
 - e.g. who provides weather reports for offshore vessels?
- The registry itself is defined as a hierarchy of business, service, and binding descriptions expressed in XML.

2.1 Business Entity

- The business entity structure represents the provider of web services.
- It contains information about the company itself: contact information, industry categories, business identifiers, and a list of services provided.

```

<businessEntity
  businessKey="uuid:dkeieie-sdfsdf-sdfs-3343334342234"
  operator="http://www.ibm.com"
  authorizedName="John Palms">
  <name>Acme Company</name>
  <description>
    We make things that go wrong.
  </description>
  <contacts>
    <contact useType="general info">
      <description>General Information</description>
      <personName>John Doe</personname>
      <phone>(803) 777-0234</phone>
      <email>jdoe@acme.com</email>
    </contact>
  </contacts>
  <businessServices>
    ...
  </businessservices>
  <identifierBag>
    <keyedReference
      TModelKey="UUID:8989df-09df0df9-eeed"
      name="D-U-N-S"
      value="123456"/>
  </identifierbag>
  <categoryBag>
    <keyedReference
      TModelKey="UUID:eer90-erere99-9898e-ee3"
      name="NAICS"
      value="12334455"/>
  </categorybag>
</businessEntity>

```

- It contains information such as the name, description, contacts, categories, and business keys.

2.2 Business Services

- Includes information about how to bind to the web service, what type of web service it is, and what taxonomical categories it belongs to.
- The Universally Unique Identifiers (UUID) are assigned by the registry when the information is first entered.

```

<businessService serviceKey="uuid:udjeei-ded3-d33343"
  businessKey="uuid:ccd kjd898-98cd9-cddd">
  <name>Hello World Web Service</name>
  <description>A basic web service</description>
  <bindingTemplates>
    ...
  </bindingtemplates>
  <categoryBag/>
</businessService>

```

2.3 Binding Templates

- Technical descriptions of the web services represented by the business service structure.
- One service might have multiple binding templates.
- The binding template represents the implementation of the web service, i.e., what protocol, network address, etc. it uses.

```
<bindingTemplate serviceKey="uudi:123"
  bindingKey="uudi:abd45">
  <description>Hellow World SOAP binding.</description>
  <accessPoint URLType="http">
    http://localhost:8080
  </accesspoint>
  <TModelInstanceDetails>
    <TmodelInstanceInfo
      TModelKey="uudi:abfdg">
      <instanceDetails>
        <overviewDoc>
          <description>
            references the description of the WSDL
            service definition.
          </description>
          <overviewURL>
            http://localhost/helloworld.wsdl
          </overviewURL>
        </overviewDoc>
      </instanceDetails>
    </TModelInstanceInfo>
  </bindingTemplate>
```

2.4 TModels

- A TModel is a way of describing the various business, service, and template structures stored within the UDDI registry.
- For example, you can define a TModel to represent a WSDL port type then associate the TModel with one of the business service's binding templates.

```
<TModel TModelKey="uudi:asdf34"
  operator="http://www.ibm.com"
  authorizedName="John Doe">
  <name>HelloWorldInterface Port Type</name>
  <description>
    An interface for a friendly web service
  </description>
  <overviewDoc>
    <overviewURL>
      http://localhost/helloworld.wsdl
    </overviewURL>
  </overviewDoc>
</TModel>
```

3 Federated and Private UDDI Registries

- UDDI is built of a number of linked registries, all of which implement the same interface.
- These can provide distributed search capability.
 - Not hierarchical like DNS.
 - A producer might only register with some registries.
- Another option is to deploy your own private UDDI registry.
- If you later want to open its services, its a simple matter since it already implements all the interfaces.

4 UDDI Interfaces

- The UDDI standard provides two SOAP interfaces:
- InquireSOAP is used by service consumers to find a service.
- PublishSOAP is used by providers to list a service.
- The services are described with WSDL.

4.1 Publisher Interface

- It defines 16 operations:
- `get_authToken`
- `discard_authToken`
- `save_business`: creates or updates a business' information.
- `save_service`: creates or updates information about the web services that a business provides.
- `save_binding`
- `save_TModel`
- `delete_business`
- `delete_service`
- `delete_binding`
- `delete_TModel`
- `get_registeredInfo`

4.2 Inquiry Interface

- It defines the following operations:
- `find_binding`
- `find_business`
- `find_TModel`
- `get_bindingDetail`

- `get_businessDetail`
- `get_businessDetailExt`
- `get_serviceDetail`
- `get_TModelDetail`

5 From WSDL to UDDI

- There is some overlap between them in the way they describe services.
 - The coalition has a document describing the best practices to follow when using a WSDL description to generate a UDDI registration.
1. Divide the WSDL description into two files.
 - One file includes the data `types`, `messages`, `portTypes`, and `bindings`. It is known as the **interface description**.
 - The other file includes just the `service` definition. It is known as the **implementation description**. Use `wsdl:import` to import the first one.
 2. Register the interface description as a UDDI `TModel` and mark the `TModel` as one that represents a WSDL interface description (whose WSDL file must be available in a website).
 3. Specify that the service is an instance of this `TModel`.
 4. Provide a link to the WSDL implementation description, which must also be available in a website.
 5. Initialize the proxy and call the `save_service` operation to register the service.
 - For example, IBM's Emerging Technologies Toolkit⁵ allows you to easily find a service with UDDI and invoke it via a dynamically configured WSDL-based proxy.

6 WS-Inspection

- UDDI has a lot of features. It might be overkill for a small (in-house) application.
- IBM and Microsoft got together and came up with the Web Services Inspection Language⁶.
- It can be used to create a simple index of service descriptions.
- An example of two services. The first provides stock quotes with a HTTP-accessible WSDL file and UDDI-accessible. The second provides a calculator with an FTP-accessible WSDL file.

```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/"
  xmlns:wsiluddi="http://schemas.xmlsoap.org/ws/2001/10/inspection/uddi/">
<service>
  <abstract>A stock quote service with two descriptions</abstract>
  <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
    location="http://example.com/stockquote.wsdl"/>
  <description referencedNamespace="urn:uddi-org:api">
    <wsiluddi:serviceDescription location="http://www.example.com/uddi/inquiryapi">
    <wsiluddi:serviceKey>4FA28580-5C39-11D5-9FCF-BB3200333F79</wsiluddi:serviceKey>
```

```
</wsiluddi:serviceDescription>
</description>
</service>
<service>
  <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
    location="ftp://anotherexample.com/tools/calculator.wsdl"/>
</service>
<link referencedNamespace="http://schemas.xmlsoap.org/ws/2001/10/inspection/"
  location="http://example.com/moreservices.wsil"/>
</inspection>
```

- These documents should be placed in a website which must also have a `http://host/inspection.wsil` file that lists the services.
- UDDI is useful when you don't know who provides the service you want. WS-Inspection can only be used if you know who you want to use.

Notes

¹<http://www.amazon.com/exec/obidos/ASIN/0596000952/multiagentcom/>

²http://www.onjava.com/pub/a/onjava/excerpt/jws_6/index1.html

³<http://www.uddi.org>

⁴<http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>

⁵<http://www.alphaworks.ibm.com/tech/ettk>

⁶<http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html>

This talk is available at <http://jmvidal.cse.sc.edu/talks/uddi>

Copyright © 2004 Jose M Vidal. All rights reserved.