أكاديمية الحكومة الإلكترونية الفلسطينية

The Palestinian eGovernment Academy

www.egovacademy.ps

Tutorial III:

Process Integration and Service Oriented Architectures

Session 15
UDDI

Prepared By

Mohammed Aldasht

European Commission

TEMPUS

About

This tutorial is part of the PalGov project, funded by the TEMPUS IV program of the Commission of the European Communities, grant agreement 511159-TEMPUS-1-2010-1-PS-TEMPUS-JPHES. The project website: www.egovacademy.ps

Project Consortium:

- Birzeit University, Palestine (**Coordinator**)
- Palestine Polytechnic University, Palestine
- Palestine Technical University, Palestine
- Ministry of Telecom and IT, Palestine
- Ministry of Interior, Palestine
- Ministry of Local Government, Palestine
- University of Trento, Italy
- Vrije Universiteit Brussel, Belgium
- Université de Savoie, France
- University of Namur, Belgium
- TrueTrust, UK

Coordinator:
Dr. Mustafa Jarrar
Birzeit University, P.O.Box 14- Birzeit, Palestine
Telfax:+972 2 2982935 mjarrar@birzeit.edu

© Copyright Notes

Everyone is encouraged to use this material, or part of it, but should properly cite the project (logo and website), and the author of that part.

No part of this tutorial may be reproduced or modified in any form or by any means, without prior written permission from the project, who have the full copyrights on the material.

Attribution-NonCommercial-ShareAlike
CC-BY-NC-SA

This license lets others remix, tweak, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms.

Tutorial Map

Intended Learning Objectives
A: Knowledge and Understanding
3a1: Demonstrate knowledge of the fundamentals of middleware.
3a2: Describe the concept behind web service protocols.
3a3: Explain the concept of service oriented architecture.
3a4: Explain the concept of enterprise service bus.
3a5: Understanding WSDL service interfaces in UDDI.
B: Intellectual Skills
3b1: Design, develop, and deploy applications based on Service Oriented Architecture (SOA).
3b2: use Business Process Execution Language (BPEL).
3b3: using WSDL to describe web services.
C: Professional and Practical Skills
3c1: setup, Invoke, and deploy web services using integrated development environment.
3c2: construct and use REST and SOAP messages for web services communication.
D: General and Transferable Skills
d1: Working with team.
d2: Presenting and defending ideas.
d3: Use of creativity and innovation in problem solving.
d4: Develop communication skills and logical reasoning abilities.

Title	T	Name
Session0: Syllabus and overview	0	Aldasht
Sesson1: Introduction to SOA	2	Aldasht
Session2: XML namespaces & XML schema	2	Aldasht
Session 3: Xpath & Xquery	4	Romi
Session4: REST web services	3	M. Melhem
Session5: Lab2: Practice on REST	3	M. Melhem
Session 6: SOAP	2	Aldasht
Session 7: WSDL	**3**	Aldasht
Session8: Lab 3: WSDL practice	3	Aldasht
Session9: ESB	4	Aldasht
Session10: Lab4: Practice on ESB	4	Aldasht
Session11: integration patterns	4	M. Melhem
Session12: Lab5: integration patterns	4	M. Melhem
Session13: BPEL	3	Aldasht
Session14: Lab6: Practice on BPEL	3	Aldasht
Session15: UDDI	2	Aldasht

Session 15: UDDI.

Session ILOs

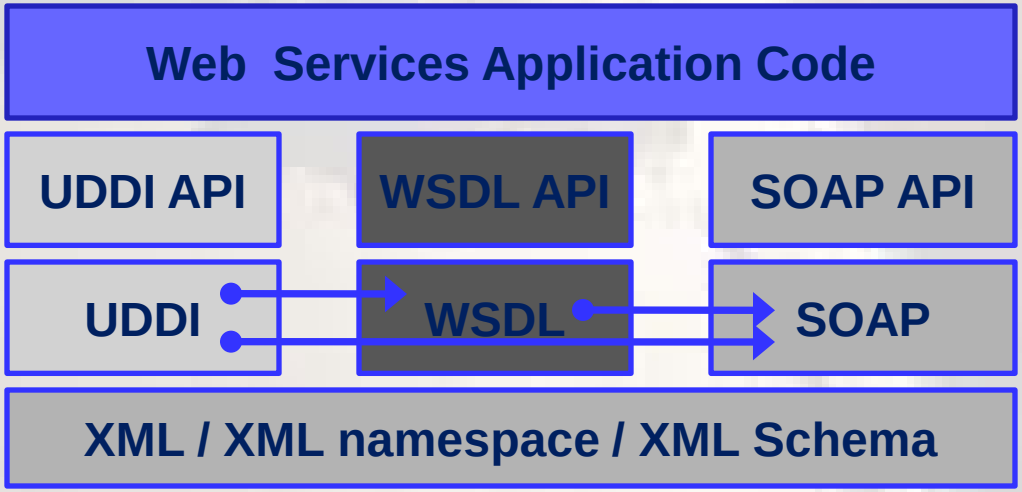
After completing this module students will be understood WSDL service interfaces in UDDI

Session Outlines

- **Introduction**
- UDDI registry structure
- Linking WSDL documents to a UDDI registry
- Private versus public UDDI registers

Overview

- A service discovery platform UDDI, offers a global repository containing [2]:
  - Service endpoint implementation.
  - Interface information
  - Business-level information.
- A UDDI registry may refer to WSDL document



Source, [2]

Introduction

- **Universal Description, Discovery and Integration** (**UDDI**) is a platform-independent, (XML)-based registry for businesses worldwide to list themselves on the Internet [1].
- UDDI is an open industry initiative, sponsored by the Organization for the Advancement of Structured Information Standards (OASIS).
- UDDI originally proposed as a core Web service standard.
- UDDI provides a mechanism to register and locate web service applications.

Introduction, cont.

- UDDI is designed to be interrogated by SOAP messages and to provide access to WSDL documents.
- UDDI registry may offer [2]:
  - Info about businesses and organizations offering web services.
  - Descriptions of web services that these organizations provide.
  - Info about technical interfaces to these web services.

Session Outlines

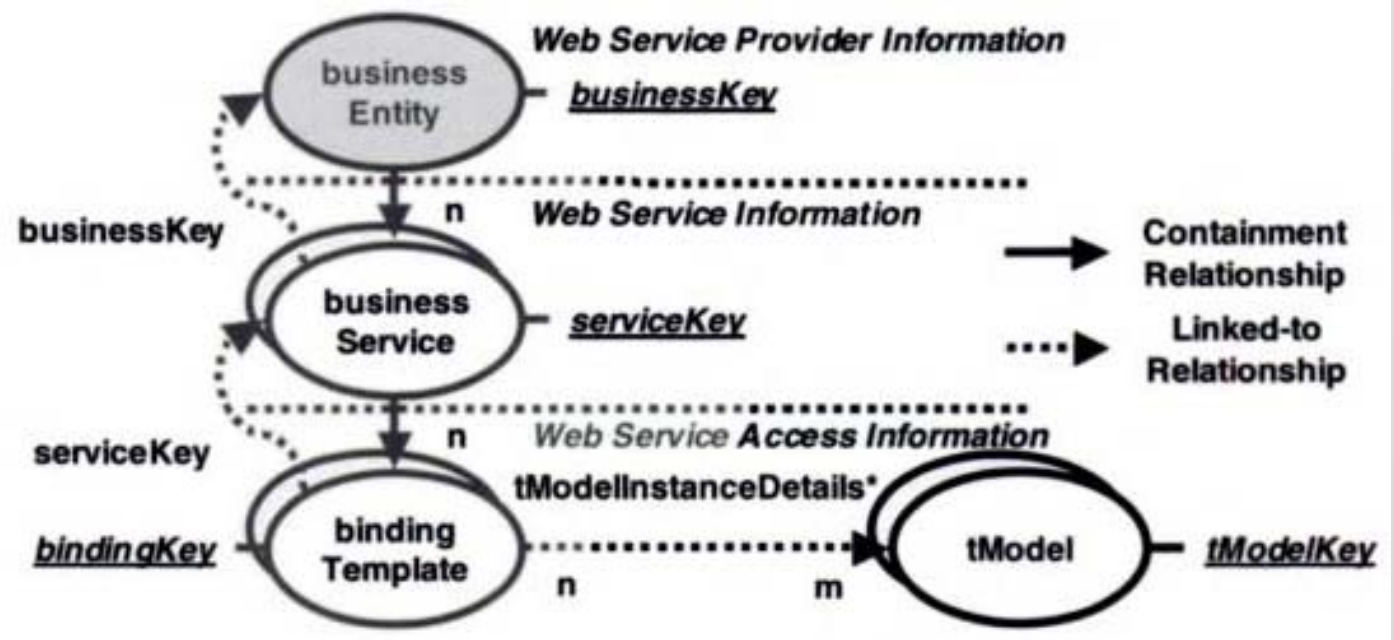
- Introduction
- **UDDI registry structure**
- Linking WSDL documents to a UDDI registry
- Private versus public UDDI registers

UDDI registry structure

- UDDI service registry data model provides two types of info:
  - Business info about web services and provider.
  - Technical info about web service binding.
- It is composed of 5 hierarchically organized data types:
  - `business entity`: a top object contains info about web service provider, like name.
  - `business service`: a data structure to represent a collection of correlated web services.
  - `binding template`: exposes a service endpoint address to access a distinct web service.
  - `tModel`: exposes web service interface information.
  - `publisher assertion`: supports publishing of business relationships between business entities.

Containment and reference relationship of UDDI data structures

- In the UDDI data model hierarchy:
  - One binding template may refer to several `tModel` "Technical Model" entities.
  - Different binding templates may refer to the same `tModel` entities



Source, [2]

Identification of Information Entities

- Individual UDDI information entities must be uniquely identified through the value of a key attribute.
- The table below lists the key attribute names for the main UDDI information entities.
- UDDI defines a specific data type for these key attributes.
  - Called Universal Unique Identifier (UUID)

Identification of Information Entities, cont.

- A UUID is basically a hex string [2],
  - Automatically generated by the UDDI server when an entity is saved for the first time.
  - It is a dash-separated character sequence.
  - e.g. “8-4-4-4-12”, the generation algorithm assures uniqueness of these IDs

Information entity	Key attribute name
`businessEntity`	`businessKey`
`businessService`	`serviceKey`
`bindingTemplate`	`bindingKey`
`tModel`	`tModelKey`

The Identifier Bag

- In addition to the descriptive information about businesses and their services, a formal identifiers are needed.
- A formal identification helps a consumer to find out who is offering a service.
- Important for deciding whether to invoke a service or not.
- e.g. on identification systems:
  - Data Universal Numbering System (D-U-N-S): uses 9-unique digits
  - Global Location Numbers (GLNs): uses 13-digit identification sequences.

Examples

e.g. on identifier bag, to be linked to namespace URI `urn:uddi-org:api_v2`:

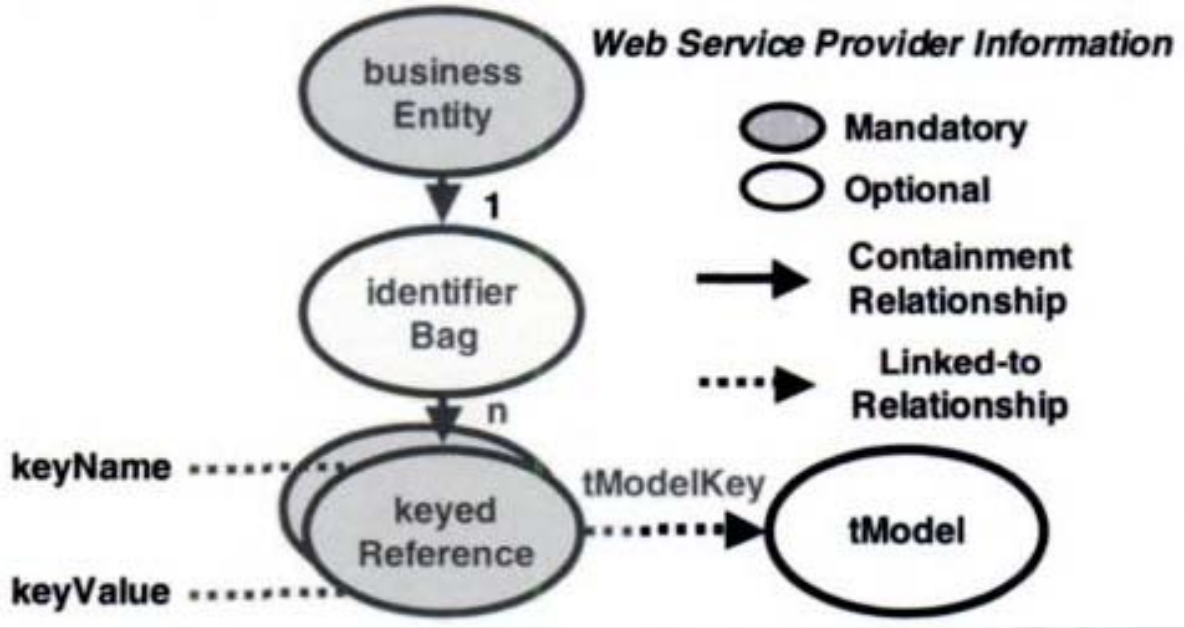
```
<element name="identifierBag" type="uddi:identifierBag" />
<complexType name="identifierBag">
  <sequence>
    <element ref="uddi:keyedReference" maxOccurs="unbounded" />
  </sequence>
</complexType>
```

A keyed reference data type represents the triplet:

```
<element name="keyedReference" type="uddi:keyedReference" />
<complexType name="keyedReference">
  <attribute name="tModelKey" type="uddi:tModelKey" use="optional" />
  <attribute name="keyName" type="string" use="optional" />
  <attribute name="keyValue" type="string" use="required" />
</complexType>
```

Identifier bag of a business entity

- May contain several keyed references for a business entity.
- Each reference identifies the business entity in a different identifier system.
- `tModels` may have an identifier bag supplement.



Source, [2]

Example on keyed reference identification

- Identifying companyx business entity.
- The `tModelKey` value specifies the identifier system D-U-N-S.
- `keyName` is human readable form of the identifier system together with the identified company.
- `keyValue` holds the nine-digit D-U-N-S identifier of companyx Inc.
  - Assuming the company is listed in the identification system

```
<!--  The keyValue “nn-nnn-nnnn” stands for an appropriate   -->
<!--  D-U-N-S identifier, character ‘n’ represents a digit.  -->
<identifierBag>
  <keyedReference
   tModelKey=“uddi:8609C81E-EE1F-4D5A-B202-3EB13AD01823”
   keyName=“D-U-N-S:companyx Inc.”
   keyValue=“nn-nnn-nnnn” />
</identifierBag>
```

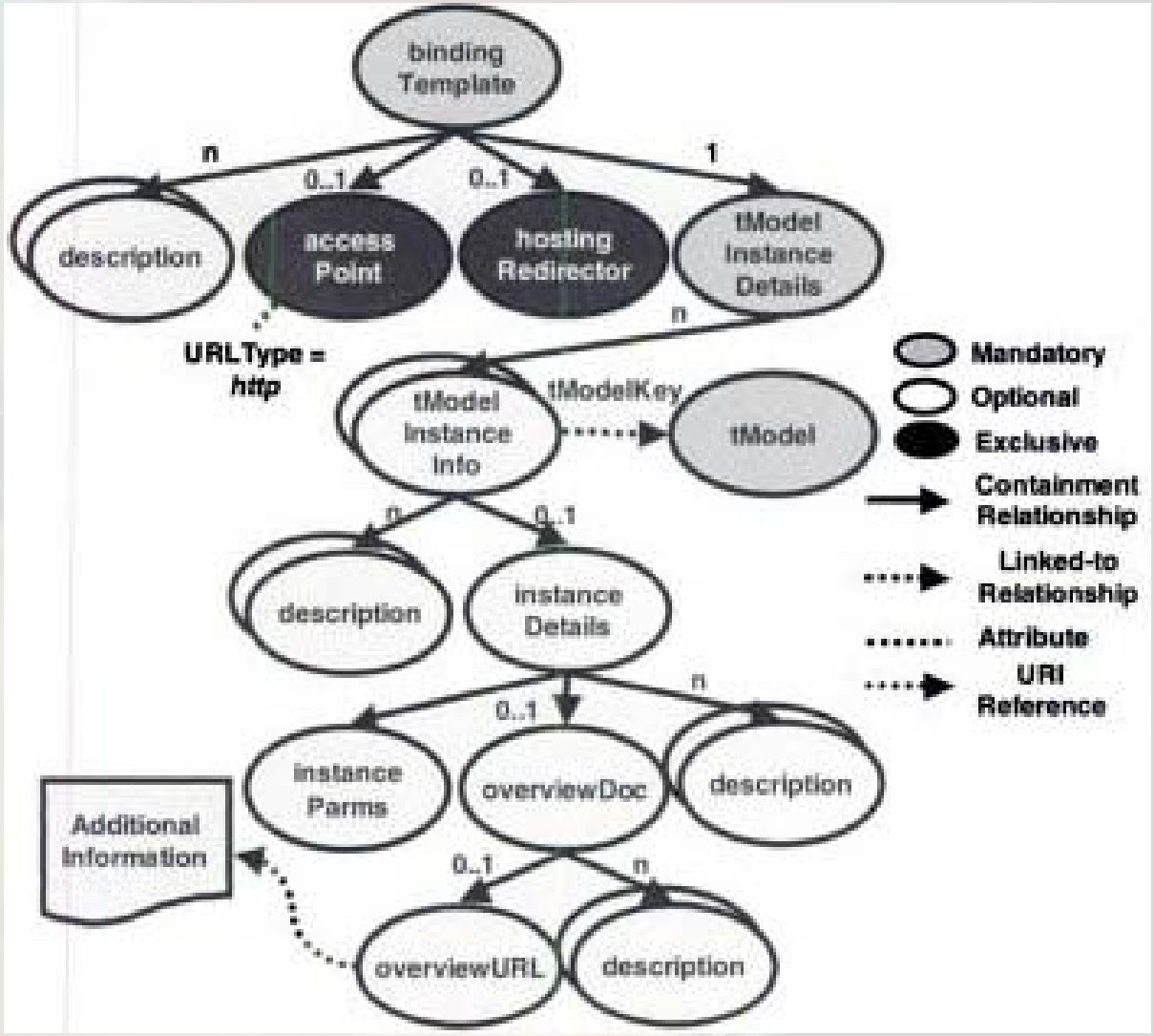
Source, [2]

Category Bag

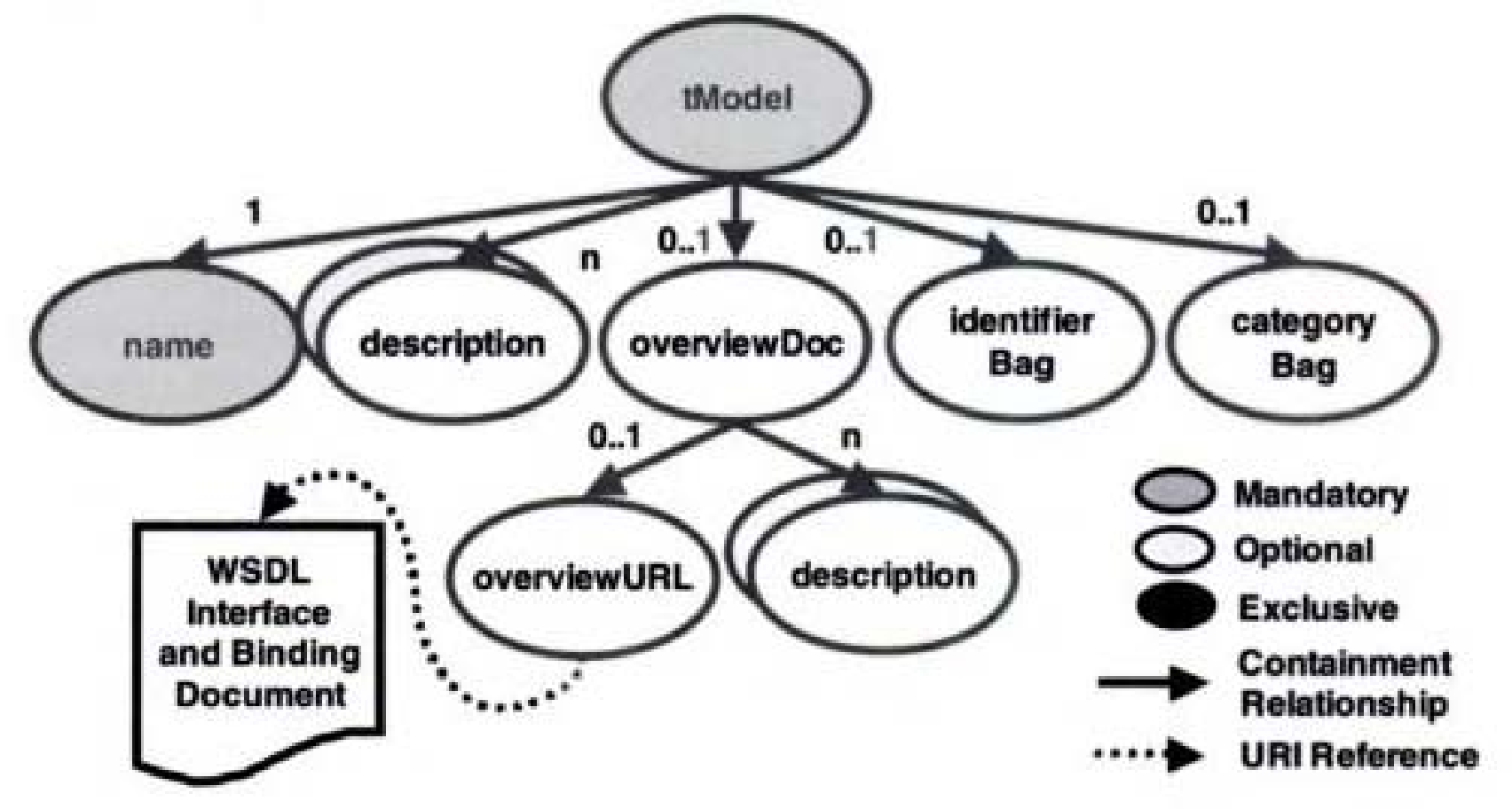
- Consumer must be able to select entities according categories.
  - Helpful, when looking for unknown service.
- Distinct categories like:
  - Region in which the service provider operates.
  - The industry it belongs to.
  - Products it offers to the market.
  - Technical means required to invoke the service.
- Category bag contains: `tModel`, `keyNames` and `keyValues`.
- Some categorization systems:
  - Universal Standard Products and Service Classification system (UNSPSC)
  - North American Industry Classification System (NICS)
  - International Standard for Geographical Regions (ISO 3166)

UDDI Binding Template

A `bindingTemplate` represents the technical description of a web service and exposes its access point information. Source, [2]



The UDDI tModel



Source, [2]

Session Outlines

- Introduction
- UDDI registry structure
- **Linking WSDL documents to a UDDI registry**
- Private versus public UDDI registers

Linking WSDL documents to a UDDI registry

- A web service consumer navigating through the UDDI business entities and services, must:
  - Discover service network address, and
  - Be delegated to an associated WSDL document describing the technical conditions for calling the service.
- UDDI `bindingTemplates` and `tModels` may implement this delegation mechanism.

WSDL authoring for UDDI registers

- WSDL `import` element supports an arbitrary breakup of individual WSDL information entities into separate files.
- The term WSDL *interface and binding* file is used to refer to the file containing the `portType` and `binding` elements.
- The term WSDL *implementation* file indicates the file holding the `service` and `port` elements.
- WSDL implementation file imports the interface and binding file
- The syntax of XML schema `import` element is semantically equivalent to the WSDL `import` element.

Session Outlines

- Introduction
- UDDI registry structure
- Linking WSDL documents to a UDDI registry
- **Private versus public UDDI registers**

Private versus public UDDI registries

Private UDDI registries

- Have a limited reach, e.g. within a company's intranet or extranet.
- Cann't publish or retrieve information from outside these UDDI registries.

Public UDDI registries

- Also known as *public cloud*.
- Accessible from the Internet.
- Any other company can publish or get information in/from the cloud
- Use a replicated UDDI nodes.
- e.g. IBM, Microsoft, ...etc.
- http://www.uddi.org/register.html`

Summary

During this session we have discussed how to publish WSDL service interfaces in UDDI; So, the following subjects have been covered:

1. UDDI registry structure
2. Linking WSDL documents to a UDDI registry
3. Private versus public UDDI registers

References

1. Extracted from: http://en.wikipedia.org/wiki/UDDI, August 2011.
2. Olaf Zimmermann, Mark Tomlinson, Stefan Peuser, “Perspectives on Web services-Applying SOAP, WSDL and UDDI to real-world projects, 2nd edition, Springer, 2005

Thanks

Mohammed Aldasht