IS 651: Distributed Systems
Chapter 6: WS-*

Jianwu Wang

Spring 2016
Learning Outcomes

• After learning chapter 6, you should be able to
  ▪ Know the reason for additional WS specifications
  ▪ Know the differences between orchestration and choreography
  ▪ Build server-side XSLT and know its differences from client-side XSLT
WS-*

• Complex information systems require many more non-functional services that support the functional ones.

• These include workflow, management, security, reliability, transactions, and quality of service (QoS)

• A great number of web service specifications have arisen to offer these kinds of services called WS-*

<table>
<thead>
<tr>
<th>Process</th>
<th>WS-BPEL, CDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS</td>
<td>WS-Security, WS-ReliableMessaging, and Addressing</td>
</tr>
<tr>
<td>Discovery</td>
<td>UDDI (and others)</td>
</tr>
<tr>
<td>Description</td>
<td>WSDL</td>
</tr>
<tr>
<td>Messaging</td>
<td>SOAP</td>
</tr>
<tr>
<td>Data</td>
<td>XML, XMLSchema, Namespaces</td>
</tr>
<tr>
<td>Transport</td>
<td>HTTP, URI, Unicode</td>
</tr>
</tbody>
</table>
Orchestration and Choreography

- Deals with business process and workflow
- Orchestration is intra-organization
- Choreography is inter-organization
- Very important examples of WS-*
Orchestration

• Any business process can be broken down into several steps. Each step can be further broken down into a series of business activities.

• WS-BPEL specification provides an elegant means to express these activities with the help of a set of pre-defined XML elements
  - Declaration elements: process, variable, partnerLink.
  - Activity elements for processes: assign, throw, wait, empty, ...
  - Activity elements for services: invoke, reply, wait, receive, ...
  - Structured Activity elements: while, pick, flow, sequence, ...
  - ...

• A BPEL process can be exposed as a service by defining its WSDL and linking to it
  - An important feature for hierarchical service composition
Orchestration in Graphic User Interface (GUI)

BPEL source code: http://userpages.umbc.edu/~jianwu/is651/programs/chp6/bpel.xml
BPEL Execution Environment

• A BPEL execution environment contains a web services framework and a BPEL server for executing the process instances
• The BPEL execution environment provides all the necessary life-cycle requirements, such as instantiation, correlation, transaction management, and so on as per the WS-BPEL specifications
• Typically a BPEL execution environment works in combination with a web services environment such as Java Platform and/or .NET environment
Choreography

- WS-CDL specification is designed to promote the business interaction and information interchange among multiple cooperating business organizations.
RSS (Really Simple Syndication)

- A different family of XML vocabularies for blogs and other uses
- All RSS is XML with various schemas depending on version
RSS Example

An RSS document, called "feed" or "channel", includes text and metadata of the text

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
  <channel>
    <title>RSS Title</title>
    <description>This is an example of an RSS feed</description>
    <link>http://www.somexamplersssdomain.com/main.html</link>
    <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000</lastBuildDate>
    <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    <item>
      <title>Example entry</title>
      <description>
        Here is some text containing an interesting description.
      </description>
      <link>http://en.wikipedia.org/wiki/RSS</link>
      <guid>unique string per item</guid>
      <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    </item>
  </channel>
</rss>
```
Server-side XSLT

• We saw client-side XSLT in Chapter 5, but most XSLT processing is done on the server using dedicated software library
• You will learn how to use SAXON (http://saxon.sourceforge.net/), a java-based XSLT processor, to do the transformation by doing the homework
Server-side XSLT Example

• XML doc:
  http://userpages.umbc.edu/~jianwu/is651/programs/chp6/book_sample.xml

• XSLT output:
  http://userpages.umbc.edu/~jianwu/is651/programs/chp6/book_sample.xhtml