Web Services

Based partially on Sun Java Tutorial at
http://java.sun.com/webservices/
Also, XML, Java and the Future of The Web, Jon Bosak.
And WSDL Tutorial at:
http://www.w3schools.com/wsdl/
World-Wide Web
(Tim Berners-Lee & Cailliau ’92)
Topics

- What are Web Services?
- XML – Extensible Markup Language
- WSDL – Web Service Definition Language
- Java APIs for Web Services
  - XML Processing
  - XML Messaging (SOAP)
  - XML Registries
  - XML-based RPC (SOAP)
What are Web Services?

- Services available via the Web.

- Meant mainly for application to application communication (as opposed to users directly)
  - Enables Business-to-Business transactions.
  - Toward a “Semantic Web”.

- E.g., a web service is contacted on a URL using the SOAP protocol over HTTP.
Web Service Examples

• A stock quote service.
  – An application requires the current value of a stock, the web service returns it.

• A route finder for delivery of goods.
  – Given an initial and a final location, find the most cost-effective delivery route.

• A weather service, a map service, a web search service…
  – any composition of Web services.
HTML Limitations

Lack of Extensibility
No new tags/attributes allowed.

Fixed Tag Structure
Emphasis on presentation in markup.

No Validation
No data-checking or types.

In contrast to SGML (Standard Generalized Markup Language).
But SGML is too complex to be appealing.
So, XML comes to the rescue.
What is XML?

- Extensible Markup Language.
- HTML++, SGML--.
- Document Type Definitions (DTD) precisely define valid tags and their grammar.
- Not backward compatible with HTML.
- System-independent and vendor-independent.
- Product of the World Wide Web Consortium (W3C), trademarked by MIT.
<xml version="1.0"?>
<PUBLICATION>
<TITLE>Why I am Overworked</TITLE>
<AUTHOR role="author">
<FIRSTNAME>Fred</FIRSTNAME>
<LASTNAME>Smith</LASTNAME>
<COMPANY>Jones and Associates</COMPANY>
</AUTHOR>
<ABSTRACT>This is the abstract</ABSTRACT>
</PUBLICATION>
<?xml version="1.0"?>
<!DOCTYPE PUBLICATION [
<!ELEMENT PUBLICATION (TITLE, AUTHOR+, ABSTRACT*)>
<!ELEMENT AUTHOR (FIRSTNAME, LASTNAME, (UNIVERSITY | COMPANY)?)> 
<!ATTLIST AUTHOR role (author|techwriter) "author" >
<!ELEMENT FIRSTNAME (#PCDATA)>
<!ELEMENT LASTNAME (#PCDATA)>
<!ELEMENT UNIVERSITY (#PCDATA)>
<!ELEMENT COMPANY (#PCDATA)>
<!ELEMENT ABSTRACT (#PCDATA)>
]>
What Makes XML Portable?

- The schema (DTD) is associated with a document which allows to perform validation on the document.
- Human-readable/writable.
- Independent of presentation (formatting).
Syntactic vs Semantic Interoperability

- While XML is portable, communicating parties still need to agree on:
  - Document type definitions
  - Meaning of tags
  - “Operations” on data (interfaces).
  - Meaning of those operations.

- Semantic interoperability is still a problem!
What is WSDL?

- Web Services Description Language
- WSDL is written in XML
- WSDL is an XML document
- WSDL is used to describe Web services
  - What operations does the service expose?
- WSDL is also used to locate Web services
  - Where is the web service located?
### WSDL Major Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Defines</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;portType&gt;</code></td>
<td>The operations performed by the web service</td>
</tr>
<tr>
<td><code>&lt;message&gt;</code></td>
<td>The messages used by the web service</td>
</tr>
<tr>
<td><code>&lt;types&gt;</code></td>
<td>The data types used by the web service</td>
</tr>
<tr>
<td><code>&lt;binding&gt;</code></td>
<td>The communication protocols used by the web service</td>
</tr>
</tbody>
</table>
WSDL Structure

```xml
<definitions>
  <types>
    definition of types...
  </types>
  <message>
    definition of a message.
  </message>
  <portType>
    definition of a port...
  </portType>
  <binding>
    definition of a binding
  </binding>
</definitions>
```
WSDL Sample Fragment

<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
WSDL Ports

- The `<portType>` element is the most important WSDL element.
- It defines a web service, the operations that can be performed, and the messages that are involved.
- The `<portType>` element can be compared to a function library (or a module, or a class) in a traditional programming language.
WSDL Messages

• The `<message>` element defines the data elements of an operation.
• Each messages can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.
WSDL Types

- The `<types>` element defines the data type that are used by the web service.
- For maximum platform neutrality, WSDL uses XML Schema syntax to define data types.
WSDL Bindings

• The `<binding>` element defines the message format and protocol details for each port.
## WSDL Operation Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-way</td>
<td>The operation can receive a message but will not return a response</td>
</tr>
<tr>
<td>Request-response</td>
<td>The operation can receive a request and will return a response</td>
</tr>
<tr>
<td>Solicit-response</td>
<td>The operation can send a request and will wait for a response</td>
</tr>
<tr>
<td>Notification</td>
<td>The operation can send a message but will not wait for a response</td>
</tr>
</tbody>
</table>
WSDL Sample Binding

```xml
<binding type="glossaryTerms" name="b1">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation>
<soap:operation
    soapAction="http://example.com/getTerm"/>/
<input> <soap:body use="literal"/> </input>
<output> <soap:body use="literal"/> </output>
</operation>
</binding>
```
Java APIs for XML

- **JAXP -- Java API for XML Processing**
  - processes XML documents using various parsers
- **JAX-RPC -- Java API for XML-based RPC**
  - sends SOAP method calls to remote parties over the Internet and receives the results
- **JAXM -- Java API for XML Messaging**
  - sends SOAP messages over the Internet
- **JAXR -- Java API for XML Registries**
  - provides a standard way to access business registries and share information
**JAX-RPC and SOAP**

- **JAX-RPC** -- Java API for XML-based RPC.
- **SOAP** – Simple Object Access Protocol
- **In JAX-RPC**, a remote procedure call is represented by an XML-based protocol such as SOAP.
- **The SOAP specification** defines envelope structure, encoding rules, and a convention for representing remote procedure calls and responses.
- **These calls and responses** are transmitted as SOAP messages over HTTP.
JAX-RPC -- SOAP

• JAX-RPC hides this complexity from the application developer.

• On the server side, the developer specifies the remote procedures by defining methods in an interface.

• The developer also codes one or more classes that implement those methods.

• Client programs create a proxy, a local object representing the service, and then simply invokes methods on the proxy.
JAX-RPC -- Java API for XML-based RPC

- A JAX-RPC client can access a Web service that is not running on the Java platform and vice versa.
- This flexibility is possible because JAX-RPC uses technologies defined by the World Wide Web Consortium (W3C): HTTP, SOAP, and WSDL.
HelloWorld Example

- HelloClient Program
- Stubs
- JAX-RPC Runtime

- HelloWorld Service
- Ties
- JAX-RPC Runtime

SOAP Message

HTTP
Downloading and Running the HelloWorld Example

- Detailed instructions for running the HelloWorld example can be found at:

http://java.sun.com/webservices/docs/1.0/tutorial/doc/JAXRPC3.html