



CSC309: Introduction to Web Programming

Lecture 12

Wael Aboulsaadat



Web Services



What are Web Services?

■ Web Service

- Distributed Computing Model
- Self-Contained Modular Applications
- Platform Independent
- Language Independent

Or

- An unpaid act of performing intricately contrived work for others that ensnares all?

The problem is . . .

- How can we automate the communication between heterogeneous independent systems ?





Hello World

```
public class HelloWorld {  
  
    public java.lang.String getMessage() {  
        return "Hello World!" ;  
    }  
  
    public static void main(String[] args) {  
        HelloWorld hw = new HelloWorld() ;  
        System.out.print(hw.getMessage()) ;  
    }  
}
```



What are Web Services?

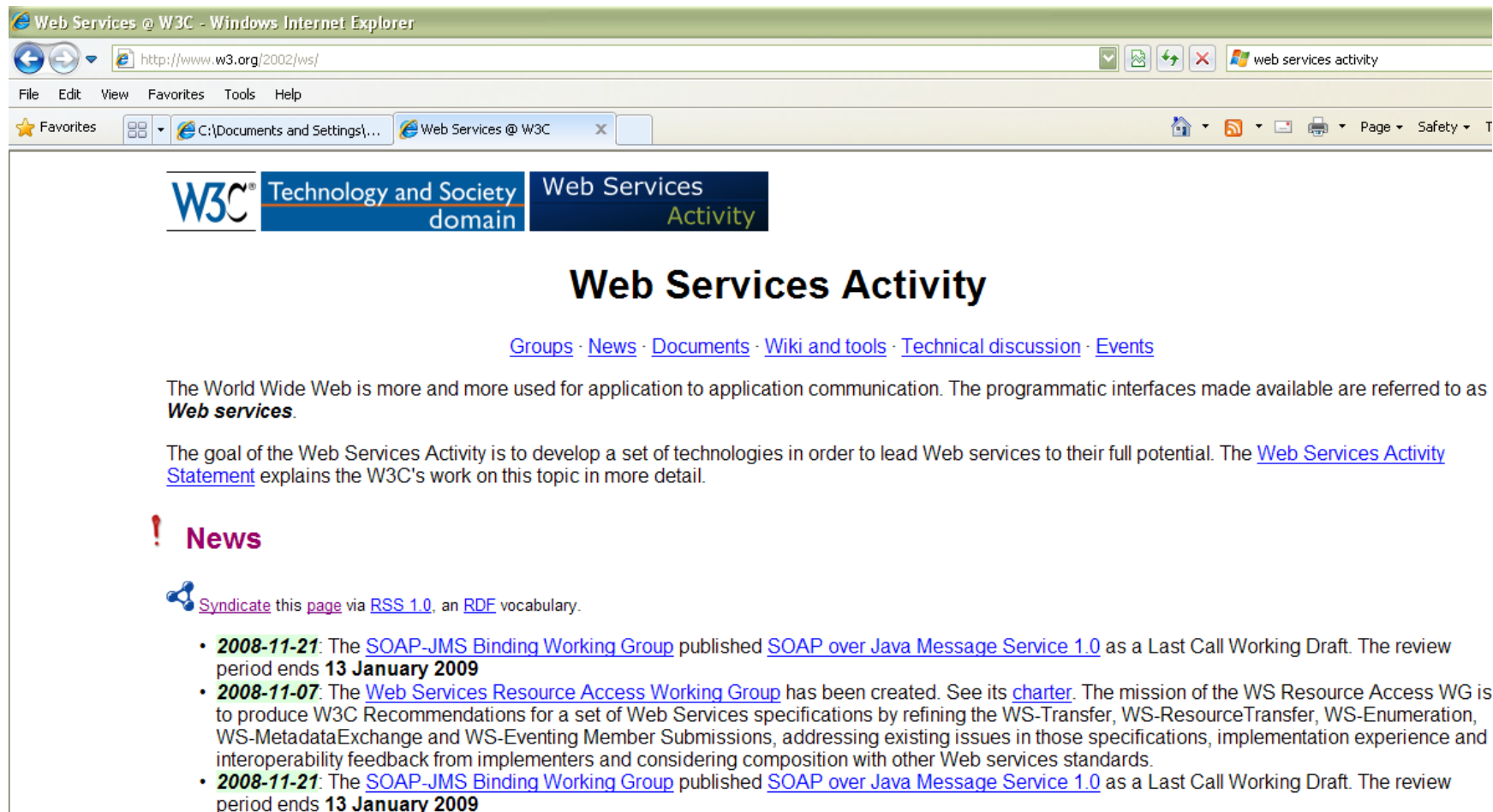
A Service that is accessed via the Web!



Who is in Control?

- W3C (www.w3c.org)
 - WSDL
 - SOAP/XML Protocol
 - Web Service Activity
- Oasis (www.oasis-open.org)
 - ebXML
 - UDDI
- WS-I (www.ws-i.org)

W3C Web Services Activity



Web Services @ W3C - Windows Internet Explorer

http://www.w3.org/2002/ws/

File Edit View Favorites Tools Help

Web Services @ W3C

W3C[®] Technology and Society domain Web Services Activity


Web Services Activity

[Groups](#) · [News](#) · [Documents](#) · [Wiki and tools](#) · [Technical discussion](#) · [Events](#)

The World Wide Web is more and more used for application to application communication. The programmatic interfaces made available are referred to as **Web services**.

The goal of the Web Services Activity is to develop a set of technologies in order to lead Web services to their full potential. The [Web Services Activity Statement](#) explains the W3C's work on this topic in more detail.

! News

 [Syndicate this page](#) via [RSS 1.0](#), an [RDF](#) vocabulary.

- **2008-11-21**: The [SOAP-JMS Binding Working Group](#) published [SOAP over Java Message Service 1.0](#) as a Last Call Working Draft. The review period ends **13 January 2009**
- **2008-11-07**: The [Web Services Resource Access Working Group](#) has been created. See its [charter](#). The mission of the WS Resource Access WG is to produce W3C Recommendations for a set of Web Services specifications by refining the WS-Transfer, WS-ResourceTransfer, WS-Enumeration, WS-MetadataExchange and WS-Eventing Member Submissions, addressing existing issues in those specifications, implementation experience and interoperability feedback from implementers and considering composition with other Web services standards.
- **2008-11-21**: The [SOAP-JMS Binding Working Group](#) published [SOAP over Java Message Service 1.0](#) as a Last Call Working Draft. The review period ends **13 January 2009**

OASIS

OASIS: Advancing open standards for the global information society - Windows Internet Explorer

http://www.oasis-open.org/home/index.php

File Edit View Favorites Tools Help

★ Favorites C:\Documents and Settings\... OASIS: Advancing open s... X

English/Other languages

OASIS Advancing open standards for the information society

About | Members | Join | News | Events | Members Only | Cover Pages | XML.org

CONSORTIUM

- OASIS Standards
- How to Participate
- Policies and Procedures
- Site Map

TECHNICAL WORK

- Committees by Name
- Committees by Category
- Adoption Services
- Computing Mgmt
- Document-Centric
- e-Commerce
- Law & Government
- Localisation
- Security
- SOA
- Standards Adoption
- Supply Chain
- Web Services
- XML Processing
- TC Guidelines
- Mailing List Directory

MEMBER SECTIONS

- CGM Open
- COSL
- eGov
- Emergency
- IDtrust
- LegalXML

OASIS is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society.

PRESS ANNOUNCEMENTS XML	PARTICIPATE
<p>[19 Mar 2009] OASIS Members Approve Open Standard for Accessing Unstructured Information</p>	<p>eGov Washington Workshop 17 April 2009 - Washington, DC Registration Now Open</p> <p>Register for the complimentary webinar, "An Introduction to the Key Management Interoperability Protocol (KMIP)."</p>
<p>[05 Feb 2009] OASIS Members Approve Nine Web Services Standards</p>	<p>Proceedings Available eGov Workshop on Public Finance on ICT Solutions Using SOA & Web Services</p>
<p>[27 Jan 2009] OASIS Members Form Committee to Address Gaps in SOA Standards for Telecommunications</p>	<p>Subscribe to SmartGrid interest list on interoperable eCommerce standards for energy</p>
<p>[08 Jan 2009] OASIS Members Approve Emergency Data Exchange Standards for Resource Messaging and Hospital Availability</p>	<p>OASIS Blue: applying lessons of e-commerce to distributed, interactive energy</p>
<p>[17 Nov 2008] OASIS Members Form</p>	<p>REVIEW</p> <p>60-day Public Review of WS-Discovery v1.1 ends 3 Apr</p>

New to OASIS?
[Take a Tour](#)

FOUNDATIONAL SPONSORS

SPONSOR MEMBERS
include

WS-I

The screenshot shows a web browser window titled "Welcome to the WS-I Organization's Web site - Windows Internet Explorer". The address bar shows "http://www.ws-i.org/". The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help) and a Favorites bar. The website content features a navigation menu with links: About Us, Join Us, Deliverables, Community, Resources, News & Events, Member Login, and a search box. The main content area includes the WS-I logo (Web Services Interoperability Organization) and a photograph of a group of people in a meeting. A central text box reads: "Establishing Best Practices for WEB SERVICES STANDARDS. The Web Services Interoperability Organization (WS-I) is an open industry organization chartered to establish Best Practices for Web services interoperability, for selected groups of Web services standards, across platforms, operating systems and programming languages. [Learn More >>](#)". To the right, a "Deliverables" section lists: "Reliable Secure Profile Usage Scenarios version 1.0" (December 15, 2008 - Final), "RSP 1.0 Test Scenarios" (November 24, 2008 - Working Group Draft), and "Basic Profile 1.2 Interoperability Scenarios" (November 24, 2008 - Working Group Draft). Below this, there are "News" and "Events" sections. The "News" section lists: "March 30, 2009: Web Services Interoperability Organization (WS-I) to Exhibit at HIMSS 2009 Conference in Chicago, April 4-8" and "February 16, 2009: The Problem With Wrapped Notifications". The "Events" section lists: "April 4, 2009: April 4-8, 2009: WS-I is exhibiting at HIMSS09". A "Comment on WS-I documents >>" link and a "View All Deliverables >>" link are also present. A footnote at the bottom right states: "Support for the Basic Profile is the baseline for interoperable Web services. Customers".



How is this different?

- RPC Model Exists!

- CORBA

- COM/DCOM

- RMI

- ...



How is this different?

- RPC Model Exists!

- CORBA

- COM/DCOM

- RMI

- ...

- **Web Services use XML!!!!!!**

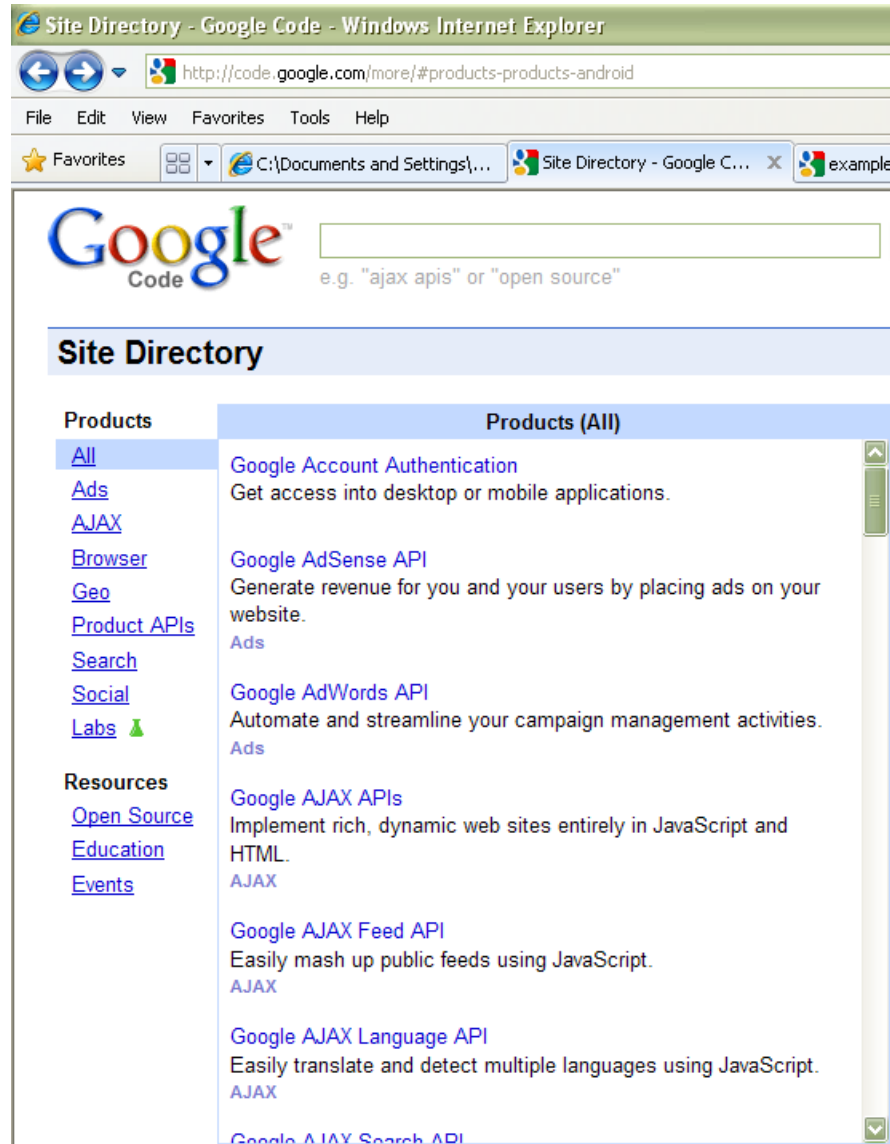


Practical Examples

- Business to Business
 - Inventory Records
 - Purchase Orders

- Business to Consumer
 - Financial Data
 - Spelling/Searching
 - Product Listings
 - Airline Reservations

Google



The screenshot shows a Windows Internet Explorer browser window displaying the Google Code Site Directory. The address bar shows the URL `http://code.google.com/more/#products-products-android`. The page features the Google Code logo and a search input field with the placeholder text "e.g. 'ajax apis' or 'open source'". Below the search field is a "Site Directory" section with a "Products" sidebar and a main content area titled "Products (All)".

Products	Products (All)
All	Google Account Authentication Get access into desktop or mobile applications.
Ads	
AJAX	
Browser	Google AdSense API Generate revenue for you and your users by placing ads on your website. Ads
Geo	
Product APIs	Google AdWords API Automate and streamline your campaign management activities. Ads
Search	
Social	
Labs	
Resources	
Open Source	Google AJAX APIs Implement rich, dynamic web sites entirely in JavaScript and HTML. AJAX
Education	Google AJAX Feed API Easily mash up public feeds using JavaScript. AJAX
Events	Google AJAX Language API Easily translate and detect multiple languages using JavaScript. AJAX
	Google AJAX Search API

Amazon

The screenshot shows the Amazon Web Services (AWS) homepage in a Windows Internet Explorer browser. The browser's address bar shows the URL <http://aws.amazon.com/>. The page features the AWS logo, navigation links for About AWS, Products, Solutions, Resources, Support, and Your Account, and a 'Contact Us' and 'Create an AWS Account' link. A main banner promotes 'Hadoop + The AWS Cloud' with a 'Sign Up Now' button. Below the banner are sections for 'Explore Products' (listing services like Amazon EC2, S3, and SQS), 'News & Events' (with a table of recent announcements), and 'Get Started' (with links to developer resources and business manager information).

Amazon Web Services

[About AWS](#) | [Products](#) | [Solutions](#) | [Resources](#) | [Support](#) | [Your Account](#)

[Contact Us](#) | [Create an AWS Account](#)

Hadoop + The AWS Cloud

Introducing Amazon Elastic MapReduce—the Hadoop-based infrastructure service that lets you build and deploy large-scale data processing applications in the cloud.

[Learn More...](#)

Explore Products

- Infrastructure Services**
 - Amazon Elastic Compute Cloud (Amazon EC2)
 - Amazon SimpleDB
 - Amazon Simple Storage Service (Amazon S3)
 - Amazon CloudFront
 - Amazon Simple Queue Service (Amazon SQS)
 - Amazon Elastic MapReduce
 - AWS Premium Support
- Payments & Billing**
- On-Demand Workforce**
- Amazon Web Services**

News & Events

What's New?	Media Coverage	Events
Apr 02, 2009	Announcing Amazon Elastic MapReduce	
Mar 30, 2009	Celebrating 3 Years of Amazon S3 with 3 Months of Transfer-In for 3 Cents/GB	
Mar 25, 2009	Announcing the AWS Toolkit for Eclipse	
Mar 24, 2009	Write Your SimpleDB Data Faster With Batch Put	
Mar 12, 2009	Amazon EC2 Introduces Reserved Instances	

[view all](#)

Get Started

Sign up for a free AWS account.

[Sign Up Now](#)

Developers

Simply sign up & start developing in the cloud with these resources and tools:

- Technical Documentation**
- Cloud Architectures Whitepaper** (pdf)
- Amazon Machine Images**
- AWS Community Forums**

Business Managers

Learn how Amazon Web Services enables you to reach business goals faster:

- AWS Solutions for Enterprise Customers**
- Security Whitepaper** (pdf)
- Case Studies & Customer Testimonials**
- AWS Blog**



Practical Benefits

- Programmatic Access
- Platform/Language Independent
- Compose/Distribute



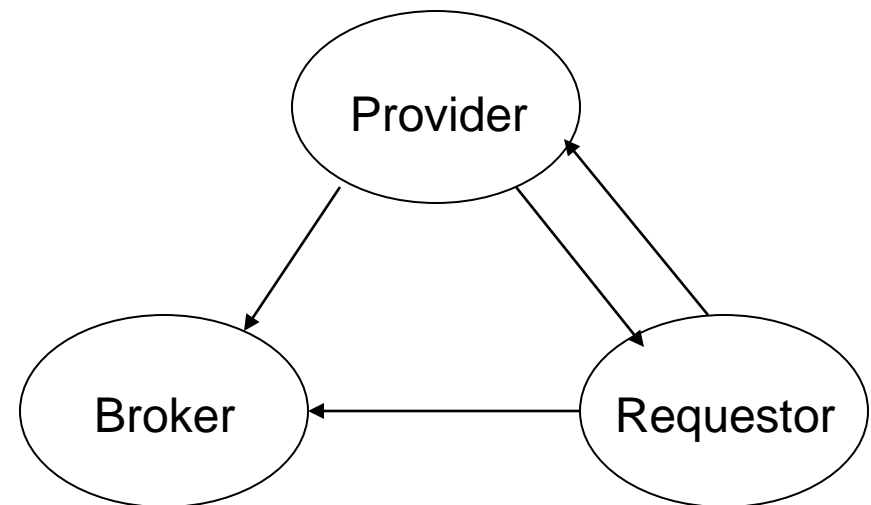
Web Service Paradigm

- Service Oriented Programming
 - Dynamically Locate Services
 - Services are “ON” the Network
 - Services can be coupled
- Multiple Transport Protocols
 - HTTP, SMTP, FTP, ...
- Multiple Message Encodings
 - SOAP, XML-RPC, XP(?), ...

Web Service Architecture

■ Three Primary Roles

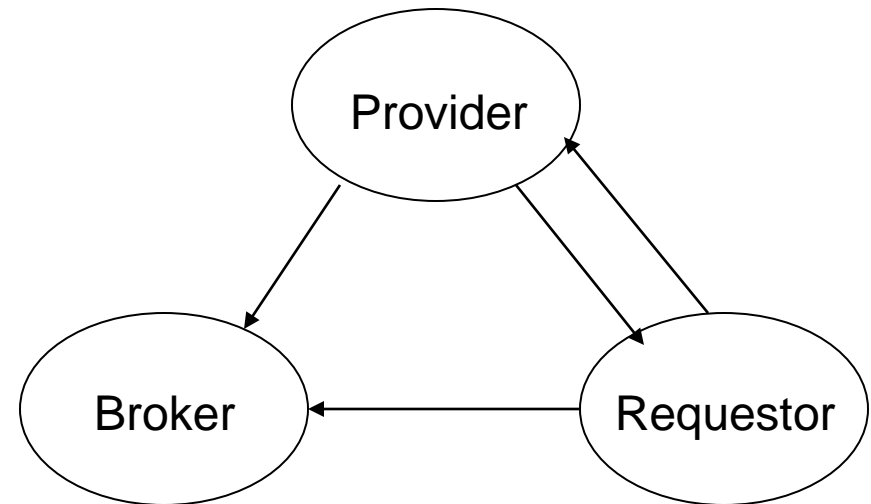
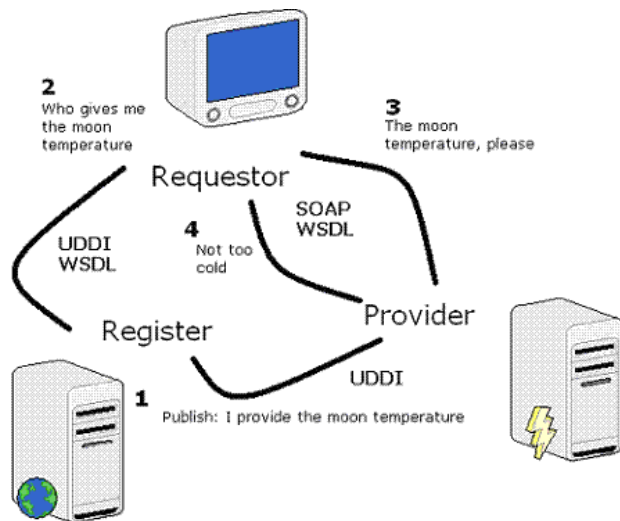
1. Service provider
2. Service requester
3. Service broker



Web Service Architecture

■ Three Primary Roles

1. Service provider
2. Service requester
3. Service broker





Web Service Architecture

- Framework must support
 1. Publishing Service
 2. Finding a Service
 3. Binding a Service



Web Service Lifecycle

1. Service Must be Created
2. Service Must Be Published
3. Service Must be Easily Located
4. Service Must be Invoked/Called
5. Service must be Unpublished



Service Provider

- **Creates the Service**
 - New Service
 - Wrap Legacy Service
 - Wrap “Other” Services
- **Publishes the Service**
 - Registries
 - Standard Hierarchies
- **Supports the Web Service**
- **Unpublishes the Service**



Service Broker

- Maintains Service Registry
- Simplifies Service Location
 - Categorization
 - Query Support



Service Requestor

- Locates Service
- Invokes Service
 - Direct Request
 - Indirect Request



The Big Three

- Service Description – WSDL
 - The most important, everything else derives from this
- Service Invocation – SOAP
 - Dominant Communication Protocol (XML Protocol)
- Service Publication – UDDI
 - Being Pushed Hard, but future not clear. (OGSA)



Describing a Service: WSDL

- Web Services Description Language (WSDL)
 - <http://www.w3.org/TR/wsd1>
 - XML Document that provides the public interface to a Web Service
 - Public Methods
 - Data Type Information (IN/OUT)
 - Transport Protocol Binding Information
 - Service Location
 - The What, Where, and How!

Invoking a Service: SOAP

- Simple Object Access Protocol (SOAP)
 - Although as of V1.2 SOAP is no longer an acronym
<http://www.w3.org/2000/xp/Group/>
- XML protocol for exchanging messages
 - Platform/Language Independent
- Different Transport Protocols (General Case)
 - HTTP/HTTPS
 - SMTP
 - FTP
 - BEEP
 - ...

Publishing a Service: UDDI

- Universal Description, Discovery, and Integration (UDDI)
 - <http://www.uddi.org> (Now under OASIS)
- Technical specification for building WSDL document repositories
 - Documents can be published
 - Document can be searched
 - Formal Hierarchy
- UDDI Registry implements the specification
 - IBM, Microsoft, SAP, etc. have public Registries
 - astrouddi.org (?)

Hello World (WSDL Style)

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:impl="http://localhost:8080/axis/HelloWorld.jws"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
{
  <wsdl:types>
  </wsdl:types>
```

```
{
  <wsdl:message name="mainRequest">
    <wsdl:part name="args" type="impl:ArrayOf_xsd_string"/>
  </wsdl:message>
```

```
  <wsdl:message name="mainResponse">
  </wsdl:message>
```

```
  <wsdl:message name="getMessageResponse">
    <wsdl:part name="getMessageReturn" type="xsd:string"/>
  </wsdl:message>
```

```
  <wsdl:message name="getMessageRequest">
  </wsdl:message>
```

```
{
  <wsdl:portType name="HelloWorld">
  </wsdl:portType>
```

```
{
  <wsdl:binding name="HelloWorldSoapBinding" type="impl:HelloWorld">
  </wsdl:binding>
```

```
{
  <wsdl:service name="HelloWorldService">
  </wsdl:service>
```

```
</wsdl:definitions>
```

Hello World (WSDL Style)

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:impl="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:intf="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:apacheSOAP="http://xml.apache.org/xml-soap"
  xmlns:wsdlSOAP="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:SOAPENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:XSD="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://localhost:8080/axis/HelloWorld.jws">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/">
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="SOAPENC:Array">
            <attribute ref="SOAPENC:ArrayType" wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
      <element name="ArrayOf_xsd_string" nillable="true" type="impl:ArrayOf_xsd_string"/>
    </schema>
  </wsdl:types>
  <wsdl:message name="mainRequest">
    <wsdl:part name="args" type="impl:ArrayOf_xsd_string"/>
  </wsdl:message>
  <wsdl:message name="getMessageResponse">
    <wsdl:part name="getMessageReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="getMessageRequest">
  </wsdl:message>
  <wsdl:message name="mainResponse">
  </wsdl:message>
  <wsdl:portType name="HelloWorld">
    <wsdl:operation name="main" parameterOrder="args">
      <wsdl:input name="mainRequest" message="impl:mainRequest"/>
      <wsdl:output name="mainResponse" message="impl:mainResponse"/>
    </wsdl:operation>
    <wsdl:operation name="getMessage">
      <wsdl:input name="getMessageRequest" message="impl:getMessageRequest"/>
      <wsdl:output name="getMessageResponse" message="impl:getMessageResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="HelloWorldSoapBinding" type="impl:HelloWorld">
    <wsdl:soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="main">
      <wsdl:soap:operation soapAction="" />
      <wsdl:input name="mainRequest">
        <wsdl:soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
      </wsdl:input>
      <wsdl:output name="mainResponse">
        <wsdl:soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getMessage">
      <wsdl:soap:operation soapAction="" />
      <wsdl:input name="getMessageRequest">
        <wsdl:soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
      </wsdl:input>
      <wsdl:output name="getMessageResponse">
        <wsdl:soap:body use="encoded"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="HelloWorldService">
    <wsdl:port name="HelloWorld" binding="impl:HelloWorldSoapBinding">
      <wsdl:soap:address location="http://localhost:8080/axis/HelloWorld.jws"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

WSDL Definitions Element

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:impl="http://localhost:8080/axis/HelloWorld.jws"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
...
</wsdl:definitions>
```



WSDL Document Elements

- `<wsdl:types>`
The datatypes used by the Web Service
- `<wsdl:message>`
The abstract definition of the data being transmitted
- `<wsdl:portType>`
The abstract operations that constitute the Web service
- `<wsdl:binding>`
The concrete protocol and data format used by the Web service
- `<wsdl:port>`
The address for a single communication endpoint
- `<wsdl:service>`
An aggregation of related ports



WSDL Types

- Define the datatypes used as arguments to the Web service as well as the return values from a Web service
- Preferably XML Schema
 - XSD namespace
- Must Handle nillable (Java Wrapper Classes)
- SOAP

WSDL Types

■ Map WSDL (XSD) to Language (e.g., Java)

xsd:boolean	boolean
xsd:byte	byte
xsd:double	double
xsd:float	float
xsd:int	int
xsd:long	long
xsd:short	short
xsd:dateTime	java.util.Calendar
xsd:decimal	java.math.BigDecimal
xsd:hexBinary	byte[]
xsd:base64Binary	byte[]
xsd:QName	javax.xml.namespace.QName
xsd:integer	java.math.BigInteger
xsd:string	java.lang.String



WSDL Types

- Recommended approach
 - Use Elements not Attributes
 - Only define types that refer to abstract content of messages (not protocols)
 - Array types should extend the SOAP Array type
 - Name scheme: ArrayOfXXX
 - Xsd:anyType used to represent any type.

<wsdl:types>

```
<wsdl:types>
```

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://localhost:8080/axis/HelloWorld.jws">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType name="ArrayOf_xsd_string">
    <complexContent>
      <restriction base="soapenc:Array">
        <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]" />
      </restriction>
    </complexContent>
  </complexType>
  <element name="ArrayOf_xsd_string" nillable="true"
    type="impl:ArrayOf_xsd_string" />
</schema>
</wsdl:types>
```

```
public class HelloWorld {

    public java.lang.String getMessage() {
        return "Hello World!";
    }

    public static void main(String[] args) {
        HelloWorld hw = new HelloWorld();
        System.out.print(hw.getMessage());
    }
}
```



Web service Messages

- Interactions between Web service client and server are called messages
- Message element describes the messages that can be exchanged
- Logical definition of a type of message that may be used by operations listed in portType element
 - Input
 - Output
 - Fault Message
- Components
 - Message must have a local name



Web service Messages

- Components (wsdl:message element)
 - Message must have a local name
 - Use WSDL Namespace
 - Zero or more Part descriptions
 - part name
 - part type
 - Arguments or return parameters.
 - Should follow XML Schema

- Message element Future?



<wsdl:message>

```
<wsdl:message name="mainRequest">
```

```
  <wsdl:part name="args" type="impl:ArrayOf_xsd_string"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="getMessageResponse">
```

```
  <wsdl:part name="getMessageReturn" type="xsd:string"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="getMessageRequest">
```

```
</wsdl:message>
```

```
<wsdl:message name="mainResponse">
```

```
</wsdl:message>
```

```
public class HelloWorld {  
  
    public java.lang.String getMessage() {  
        return "Hello World!";  
    }  
  
    public static void main(String[] args) {  
        HelloWorld hw = new HelloWorld();  
        System.out.print(hw.getMessage());  
    }  
}
```

WSDL Port Types

WSDL defines four transmission primitives (or operations) that an endpoint can support

- **One-way (input element)**
 - The endpoint receives a request, but does not send a response.
- **Request-response (input then output element)**
 - The endpoint receives a request, and sends a **correlated** response.
- **Solicit-response (output then input element)**
 - The endpoint sends a response, and receives a **correlated** response.
- **Notification (output element)**
 - The endpoint sends a response, but does not receive a request.





WSDL portType

- A portType element defines the interfaces that a Web service exposes.
- Similar to a
 - Class
 - Module
 - or Function Library
- The operations are the class/module/library methods.

<wsdl:portType>

```
<wsdl:portType name="HelloWorld">
```

```
  <wsdl:operation name="main" parameterOrder="args">
```

```
    <wsdl:input name="mainRequest" message="impl:mainRequest"/>
```

```
    <wsdl:output name="mainResponse" message="impl:mainResponse"/>
```

```
  </wsdl:operation>
```

```
  <wsdl:operation name="getMessage">
```

```
    <wsdl:input name="getMessageRequest" message="impl:getMessageRequest"/>
```

```
    <wsdl:output name="getMessageResponse"
      message="impl:getMessageResponse"/>
```

```
  </wsdl:operation>
```

```
</wsdl:portType>
```

```
public class HelloWorld {
    public java.lang.String getMessage() {
        return "Hello World!";
    }

    public static void main(String[] args) {
        HelloWorld hw = new HelloWorld();
        System.out.print(hw.getMessage());
    }
}
```



WSDL Binding

- Defines message format

- For a given portType, defines protocol
 - for operations
 - for messages

- Requires unique name attribute

- Type attribute is portType QName

<wsdl:binding>

```
<wsdl:binding name="HelloWorldSoapBinding" type="impl:HelloWorld">
```

```
  <wsdlsoap:binding style="rpc"
```

```
    transport="http://schemas.xmlsoap.org/soap/http"/>
```

...

```
  <wsdl:operation name="getMessage">
```

```
    <wsdlsoap:operation soapAction=""/>
```

```
      <wsdl:input name="getMessageRequest">
```

```
        <wsdlsoap:body use="encoded"
```

```
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
```

```
      </wsdl:input>
```

```
      <wsdl:output name="getMessageResponse">
```

```
        <wsdlsoap:body use="encoded"
```

```
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```
          namespace="http://localhost:8080/axis/HelloWorld.jws"/>
```

```
      </wsdl:output>
```

```
    </wsdl:operation>
```

```
</wsdl:binding>
```



WSDL Services

- A port defines a single endpoint
- The port can then be used for binding
- Multiple ports can reference the same address with different protocols
- A Service consists of one or more ports
- A service defines a single serviceType



<wsdl:service> & <wsdl:port>

```
<wsdl:service name="HelloWorldService">  
  <wsdl:port name="HelloWorld" binding="impl:HelloWorldSoapBinding">  
    <wsdlsoap:address  
      location="http://localhost:8080/axis/HelloWorld.jws"/>  
  </wsdl:port>  
</wsdl:service>
```



Invoking a Service

- Use SOAP to communicate messages
 - SOAP Sender to SOAP Receiver
 - Potential SOAP Intermediaries
- Essentially a one-way communication between SOAP nodes.
 - RPC style
 - Document style



SOAP Basics

- Message is wrapped in the Envelope

- Envelope consists of
 - Header (Optional) used by intermediaries
 - Body contains the actual message
 - Document
 - Service Call

- Fault Handling
 - Child element of body
 - Contains Reason and Code elements

SOAP Basics

- Fault Handling (V1.2)
 - Fault Element is a child element of body
 - No other elements in the body
 - Contains
 - Reason element (Mandatory)
 - Code element (Mandatory)
 - Standard List
 - Detail element (Optional)
 - Node element (Optional)
 - Role element (Optional)



SOAP Request (HelloWorld)

```
POST /axis/HelloWorld.jws HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.0
Host: localhost
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 407
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getMessage
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://localhost:8080/axis/HelloWorld.jws"/>
    </soapenv:Body>
  </soapenv:Envelope>
```



SOAP Response (HelloWorld)

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Connection: close

Date: Wed, 09 Oct 2002 21:34:47 GMT

Server: Apache Tomcat/4.0.6 (HTTP/1.1 Connector)

Set-Cookie: JSESSIONID=8A6802F3136B882A53BC0E8E1E30F8CC;Path=/axis

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getMessageResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://localhost:8080/axis/HelloWorld.jws">
      <getMessageReturn xsi:type="xsd:string">Hello World!</getMessageReturn>
    </ns1:getMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



Web Service Registries

- UDDI Currently Dominant
 - Public Registries
 - IBM, MS, SAP (all have been taken offline!).
 - Private Registries
 - UDDI Functions
 - Describe services
 - Discover businesses
 - Integrate business services

- The MetaData Problem

UDDI Registry

The screenshot shows the 'seekda!' website interface. At the top, there's a navigation bar with icons for 'login', 'register', 'home', 'help', and 'contact'. Below this is a main menu with tabs for 'Seek Services', 'News', 'Consumers', 'Providers', and 'About'. The central area features a 'Web Services Search Engine' with a search input field and a 'Search' button. To the left, there's a 'What is seekda?' section explaining the search engine's purpose and a 'seekda! connect' section. To the right, there's a 'Get in Touch' section and a 'Service Tags' section listing various categories like 'business science USA', 'development', 'free onsale', 'unknown unknow', 'bioinformatics commercial', 'company free internal', 'offline research restricted STS test', and 'tourism university'. A 'Featured' section is also visible at the bottom right.

UDDI Registry

The screenshot shows a web browser window displaying the 'MovieInformation' service details on the seekda! UDDI Registry. The browser's address bar shows the URL: <http://seekda.com/providers/ignyte.com/MovieInformation>. The website has a navigation menu with 'Seek Services', 'News', 'Consumers', 'Providers', and 'About'. The main content area is titled 'Web Service Details: MovieInformation' and includes a 'Bookmark' button. Below the title are tabs for 'Overview', 'Use Now', 'Availability', 'Comments', and 'Wiki History'. The 'Overview' tab is active, displaying the following service details:

- Country: United States
- Provider: ignyte.com
- WSDL File: <http://www.ignyte.com/webser...vice/moviefunctions.asmx?wsdl>
- WSDL Cache: [XML or HTML](#)
- Monitored since: Jul 14, 2006
- Server: Microsoft-IIS/6.0
- Availability: (98.50% since Jul 2006) [\(view details\)](#)
- Documentation: good (within WSDL)
- Description:
- User Rating: (cast your vote by clicking the corresponding star)

Below the details is a section for 'Service Description by seekda Users' with an 'edit' button. The description reads: 'This service retrieves a list of current movies. Works for e.g. New York (zip 10027) and 1 mile ...'. On the left side of the page, there is a sidebar with navigation options: 'Advanced Search', 'Services Tag Cloud', 'Providers by Countries', 'Most Used Services', 'My Bookmarks', and 'Recently Found Services'. Below these is a 'Search Services' section with a search input field and a 'Search' button. At the bottom of the sidebar, there is a 'Recently Viewed' section showing the 'MovieInformat...' service.

On the right side of the page, there is a 'Get in Touch' section with contact information and an 'Ads' section featuring 'Ads by Google' for 'Cloud Computing Platforms' and 'Restore PC Performance'.



UDDI Private Registry

- Some development tools or products provide private UDDI registry server
 - Java WS Developer pack.
 - Oracle JDeveloper
 - IBM WS toolkit
 - MS VS .NET

- Greater control, no registration!



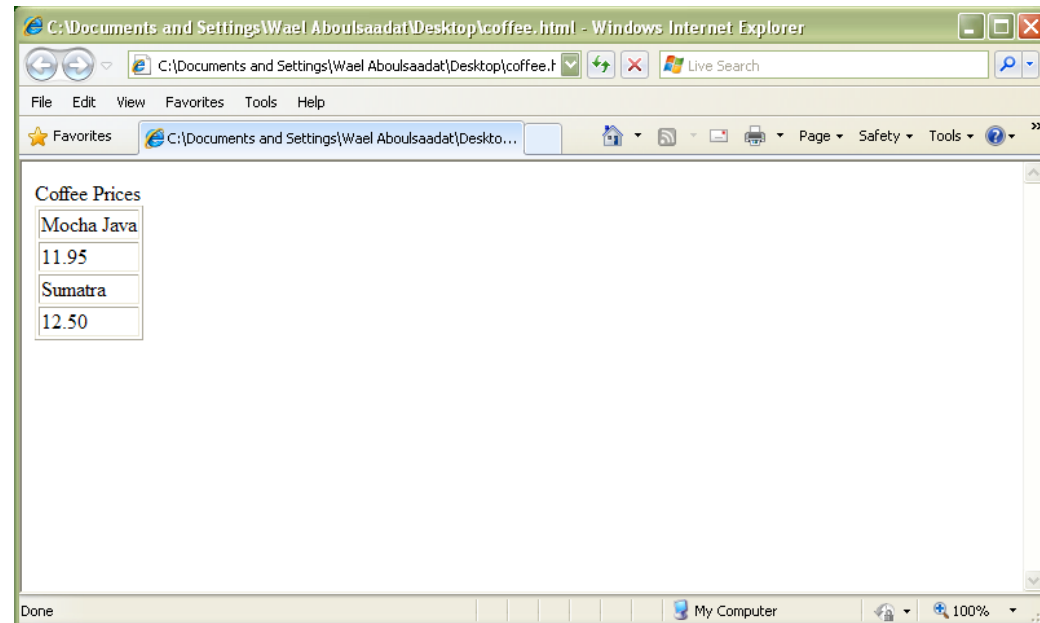
Web Services & the future!

- Web 3.0...



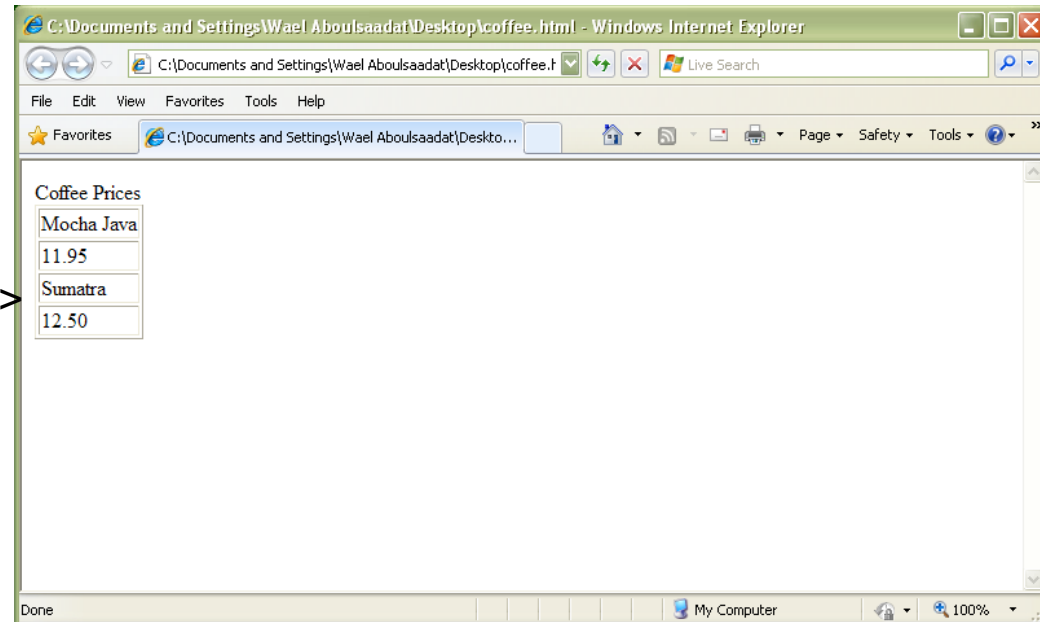
XSL && XSLT

```
<priceList>
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
  <coffee>
    <name>Sumatra</name>
    <price>12.50</price>
  </coffee>
</priceList>
```



```
<priceList>
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
  <coffee>
    <name>Sumatra</name>
    <price>12.50</price>
  </coffee>
</priceList>
```

```
<html>
<head>Coffee Prices</head>
<body>
  <table border="1">
    <tr><td>Mocha Java</td></tr>
    <tr><td>11.95</td></tr>
    <tr><td>Sumatra</td></tr>
    <tr><td>12.50</td></tr>
  </table>
</body>
</html>
```



```
<priceList>
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
  <coffee>
    <name>Sumatra</name>
    <price>12.50</price>
  </coffee>
</priceList>
```

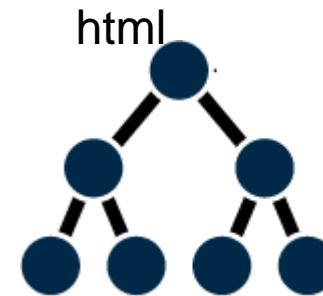
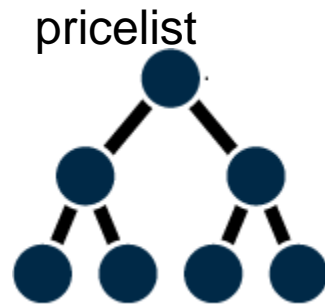


```
<html>
<head>Coffee Prices</head>
<body>
  <table border="1">
    <tr><td>Mocha Java</td></tr>
    <tr><td>11.95</td></tr>
    <tr><td>Sumatra</td></tr>
    <tr><td>12.50</td></tr>
  </table>
</body>
</html>
```



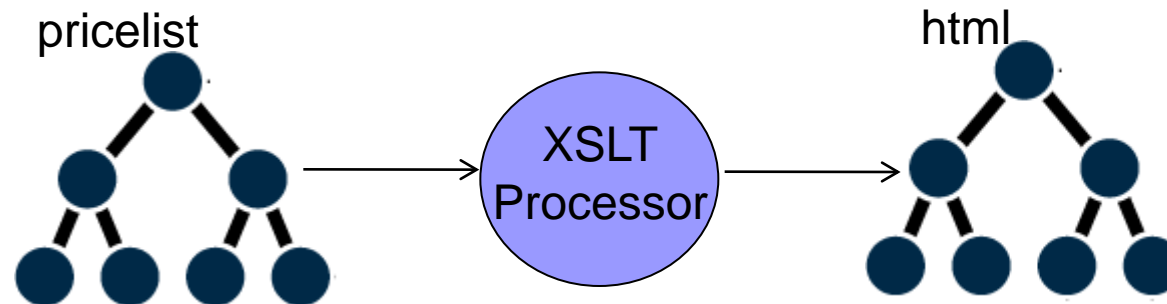
XML Transformations

- How can we transform one xml tree to another XML tree?



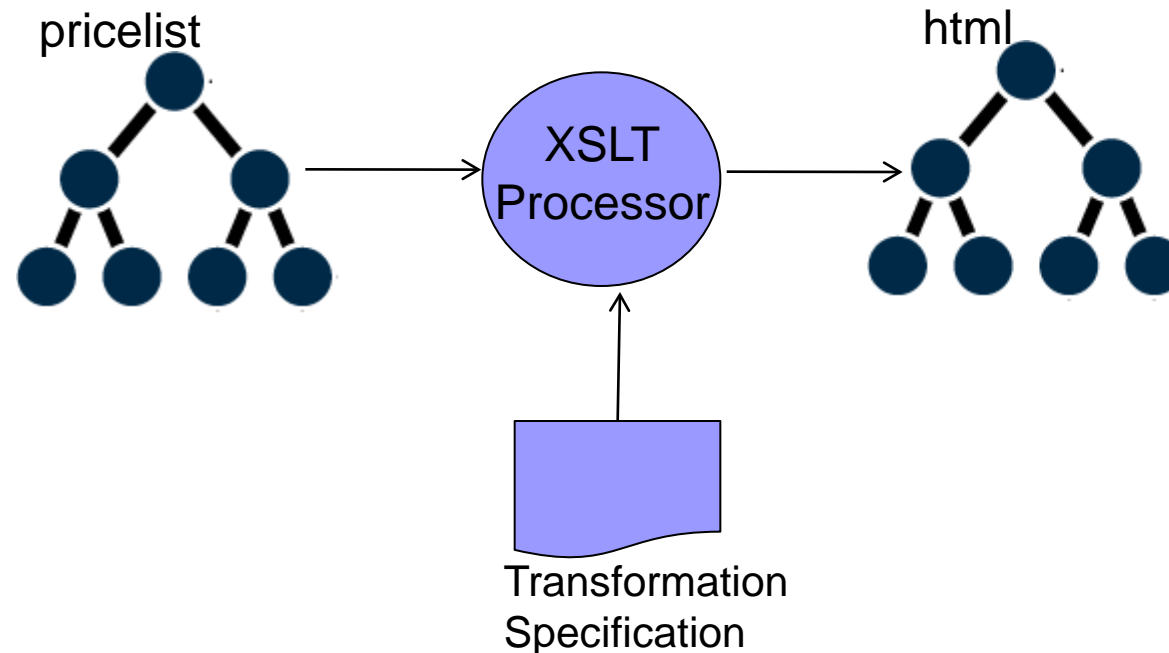
XML Transformations

- How can we transform one xml tree to another XML tree?



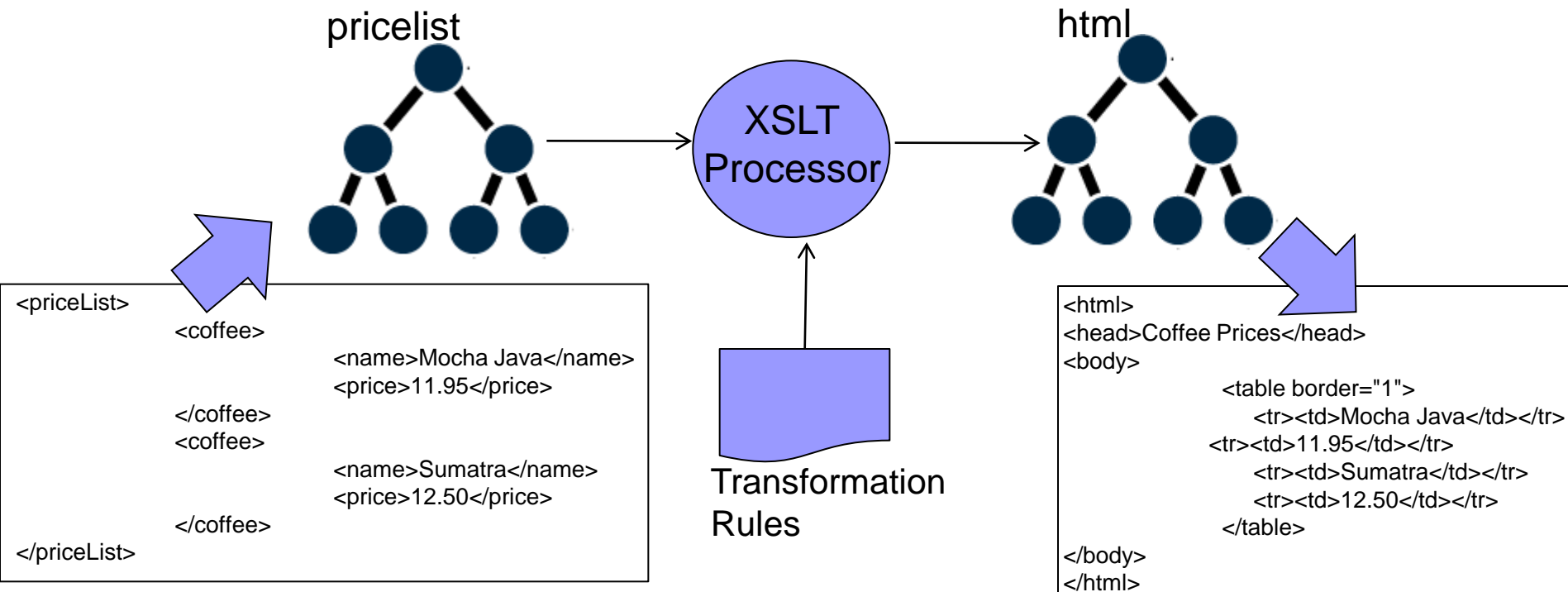
XML Transformations

- How can we transform one xml tree to another XML tree?



XML Transformations

- How can we transform one xml tree to another XML tree?





How do you apply XSLT Rules?

- In Browser

include in xml

```
<?xml-stylesheet href="URI"  
  type=""  
  title=""  
  media=""  
  charset=""  
  alternate="yes"  ?>
```

- On Server:

Use an XSL processor/engine – e.g. XT, Xalou, Saxon

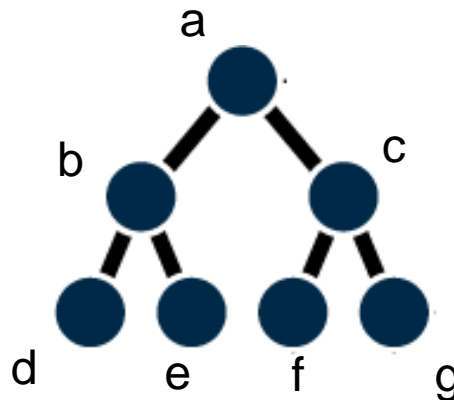


XSLT Demos

XSL Transformations (XSLT)

■ Basic idea:

1. Match on a path along the XML tree
2. Specify what you want to do after the parser finds a match



Possible interesting matches:

/a
/a/b
/a/c
/a/b/d
/a/c/g

Not interesting:

/a/b/g
/c/g



XSL Transformations (XSLT)

- Full-featured procedural programming language
 - Flow control (e.g., looping & conditional branching)
 - Variables, input parameters supported
 - Basic string manipulation, numeric calculations
 - Modularity, calls to external functions, recursion
- External functions execute within constraints of transforming XML parser being used
- Single XML document as output (1.0)

XSL Transformations (XSLT)

- XSLT documents are themselves XML

```
<xsl:template match='//author'>
  <xsl:element name='dc:creator'>
    <xsl:value-of select='lastname' />
    <xsl:text>, </xsl:text>
    <xsl:value-of select='firstname' />
  </xsl:element >
</xsl:template>
```



How do you specify a path in an XML Tree?

- XPath: Language for addressing parts of an XML file
 - Data model for XML
 - Includes tests, pattern matching
- Used in XSLT to access DOM elements



XPath

- Here's a simple XML document:

```
<?xml version="1.0"?>
<library>
  <book>
    <title>XML</title>
    <author>Gregory Brill</author>
  </book>
  <book>
    <title>Java and XML</title>
    <author>Brett McLaughlin</author>
  </book>
</library >
```

- XPath expressions look a lot like paths in a computer file system
 - / means the document itself (but no specific elements)
 - /library selects the root element
 - /library/book selects every book element
 - //author selects every author element, wherever it occurs



XPath filters

- path to element or attribute may be optionally followed by filter

[element=value]

for example

book[author='Jones']

allowed operators are: = != < >

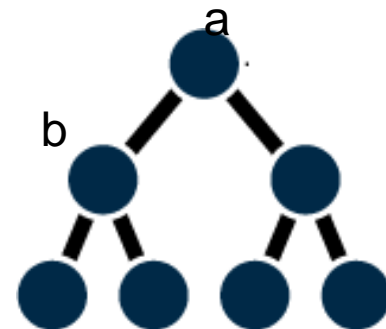


What's inside an XSLT document?

- Any number of “templates”
- A template uses Xpath to match nodes
- Highest priority matching template selected
- Then the template takes over and generates:
 - Literal output XML
 - Computational results (of XSLT functions)
 - Results of further template applications
 - Results of queries on the document
- Many options

An XSLT Stylesheet HelloWorld

```
<xsl:stylesheet version="1.1">  
  
  <xsl:template match="/">  
    <html><head>Hello World</head>  
    <body>  
      <xsl:apply-templates />  
    </body>  
  </html>  
</xsl:template>  
  
  <xsl:template match="a/b">  
  
  </xsl:template>  
  
  ...  
</xsl:stylesheet>
```





XSLT Template Element

```
<template  
  match = xpath-expression  
  name = name  
  priority = number  
  mode = name>
```



Applying Templates

```
<apply-templates  
  select = expression  
  mode = name>
```

```
<call-template name = name>
```

Example 1

```
<?xml version="1.0"?>
<?xml-stylesheet href="style1.xsl" type="text/xsl"?>

<items>

  <item partNum="123-AB">
    <productName>Porsche</productName>
    <quantity>1</quantity>
  </item>

  <item>
    <productName>Ferrari</productName>
    <quantity>2</quantity>
  </item>

</items>
```

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match="/">
  <html>
    <head></head>
    <body>
      <ol>
        <li>Root</li>
        <xsl:apply-templates/>
      </ol>
    </body>
  </html>
</xsl:template>
```

```
<xsl:template match="items">
  <li>Items</li>
  <xsl:apply-templates/>
</xsl:template>
```

```
<xsl:template match="item">
  <li>Item: <xsl:value-of select="productName"/></li>
</xsl:template>
</xsl:stylesheet>
```



Example 2

```
<?xml version="1.0"?>
<?xml-stylesheet href="style1.xsl" type="text/xsl"?>

<items>

  <item partNum="123-AB">
    <productName>Porsche</productName>
    <quantity>1</quantity>
  </item>

  <item>
    <productName>Ferrari</productName>
    <quantity>2</quantity>
  </item>

</items>
```

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/TR/WD-xsl">

  <xsl:template match="/">
    <html>
      <head></head>
      <body>
        <ol>
          <xsl:apply-templates/>
          <li>Root</li>
        </ol>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="items">
    <xsl:apply-templates/>
    <li>Items</li>
  </xsl:template>

  <xsl:template match="item">
    <li>Item: <xsl:value-of select="productName"/></li>
  </xsl:template>

</xsl:stylesheet>
```



Repetition

```
<for-each  
  select = "item">  
  Do something here ...  
</for-each>
```



Example 3

```
<?xml version="1.0"?>
<?xml-stylesheet href="style1.xsl" type="text/xsl"?>

<items>

  <item partNum="123-AB">
    <productName>Porsche</productName>
    <quantity>1</quantity>
  </item>

  <item>
    <productName>Ferrari</productName>
    <quantity>2</quantity>
  </item>

</items>
```

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<xsl:template match="/">
  <html>
    <head></head>
    <body>
      <ol>
        <li>Root</li>
        <xsl:apply-templates/>
      </ol>
    </body>
  </html>
</xsl:template>

<xsl:template match="items">
  <li>Items</li>
  <xsl:for-each select="item">
    <li>Item:
      <xsl:value-of select="productName"/> </li>
  </xsl:for-each>
</xsl:template>

</xsl:stylesheet>
```