



ICT336 Internet Systems Programming

Introduction to Web Services (Week 10 Lecture 1)



Learning Objectives

- Learn about the concept of Web Services why it is an important development in Internet technologies.
- Learn about the basic set-up for Web Services.



Learning Objectives

- In the scheme of what we are doing in this unit:
 - We are studying how to use XML as an important set of Internet technologies to use as solutions in different areas.
 - It is likely that any XML work in the industry will involve the use of Web Services. This lecture is to start you in the direction of learning about the concepts.



Lecture Outline

- What are Web Services?
- Components of Web Services
- The Role of XML



Prologue

- In this lecture, I will follow the online book required reading (see last page) quite closely.
 - Please read the required reading after this lecture.
 - Those listening to this lecture on iLecture, the diagrams I draw in the lectures are the same as the ones in the required reading
 - You may choose to have the online book up while listening to the lecture - may help you understand what I am saying.



From Manual to Automated

- *"Today, the principal use of the World Wide Web is for interactive access to documents and applications. In almost all cases, such access is by human users, typically working through Web browsers, audio players, or other interactive front-end systems. The Web can grow significantly in power and scope if it is extended to support communication between applications, from one program to another."*

W3C XML Protocol Working Group Charter.



What is a "Web Service"

- "A web service is any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any one operating system or programming language."
- Options for "messaging system":
 - XML-RPC
 - SOAP



Difference from "Normal" Web

- What is the difference between this type of service and "normal" web applications we are used?
 - Eg. dynamic PHP web-sites, shopping carts, etc.
- Answer: "normal" web applications are NOT
 - Self-Describing
 - Discoverable



Difference from "Normal" Web

- Eg.
 - I can't get my web client to go and find available online applications that does a particular thing (eg. return weather data in London, convert between currencies, etc)
 - When I have a new application, no existing programs will be able to use it unless I tell the programmers explicitly about it.
 - When an application changes it's result format, other programs making use of it cannot automatically detect what those changes are.
 - etc.



Towards the Semantic Web

- From
 - Human-centric...
 - To Application-centric...
 - To Automated Web.

- Concept of the "Semantic Web".



The Semantic Web

- The ***Semantic Web*** is W3C's vision of the web where:
 - "... it becomes a place where data can be shared and processed by automated tools as well as by people."
 - "... tomorrow's programs must be able to share and process data even when these programs have been designed totally independently."
 - "The ***Semantic Web*** is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications."
 - All Quotes from <http://www.w3.org/2001/sw/>



Distributed Object Technologies

- There have been past efforts in providing program-level object exchange:
 - OMG's CORBA/IIOP (www.corba.org/)
 - Microsoft's DCOM (www.microsoft.com/com/tech/dcom.asp)
 - Java RMI (java.sun.com/rmi)

- But all of these technologies suffer from:
 - Complex set-up and object invocation.
 - Platform and language dependence.
 - lack of universal acceptance.
 - Lack of extensibility to different problem areas.



Components of A Web Service

- The Web Service infrastructure consist of
 - Providers
 - Requesters
 - Registries



Components of A Web Service

- In a crude way, the current web already have the 3 components in the form of (respectively)
 - Clients
 - Servers
 - Search Engines

- Problems is that **CONTENT** between clients, servers, and search engines are not precisely defined - not good enough for automated web.



Web Service Protocol Stack

Discovery

UDDI

Description

WSDL

Messaging

XML-RPC, SOAP, etc

Transport

HTTP, SMTP, etc



Support for Web Services

- Tools for programming and supporting Web Services are appearing under all platforms and programming languages.
- Currently,
 - Currently, SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) is in wide-spread use.
 - A lot of developments have moved away from XML-RPC (XML-Remote Procedure Calls) towards SOAP.
 - The next important set of developments in the bring UDDI (Universal Description, Discovery and Integration) into the picture.



Importance of XML

- The use of XML is an essential part of building Web Services.
- The concepts of self-description and Discovery will not as easily implementable if not for an existing wide-spread support for XML.

Example WSDL for a Weather Service

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="WeatherService"
  targetNamespace="http://www.ecerami.com/wsdl/WeatherService.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.ecerami.com/wsdl/WeatherService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <message name="getWeatherRequest">
    <part name="zipcode" type="xsd:string"/>
  </message>
  <message name="getWeatherResponse">
    <part name="temperature" type="xsd:int"/>
  </message>
  ...
  <service name="Weather_Service">
    <documentation>WSDL File for Weather Service</documentation>
    <port binding="tns:Weather_Binding" name="Weather_Port">
      <soap:address
        location="http://localhost:8080/soap/servlet/rpcrouter"/>
    </port>
  </service>
</definitions>
```

Example from section 1.4 in
Web Service Essentials by Ethan
Cerami, O'Reilly, 2002



Example SOAP Request

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeather
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <zipcode xsi:type="xsd:string">10016</zipcode>
    </ns1:getWeather>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Asking for weather data...

... for zipcode 10016.

Example from section 1.3.2 in
Web Service Essentials by Ethan Cerami, O'Reilly, 2002



Example SOAP Response

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getWeatherResponse
      xmlns:ns1="urn:examples:weatherservice"
      SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
      <return xsi:type="xsd:int">65</return>
    </ns1:getWeatherResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Returning the response...

... with the whether data.

Example from section 1.3.2 in
Web Service Essentials by Ethan Cerami, O'Reilly, 2002



Advantages of XML

- Think about what XML offers in the previous example:
 - Clients and servers know unambiguously where to retrieve the needed data (in elements `<zipcode>`, `<return>`, etc)
 - Clients and servers can determine if they are communicating the same information by looking at what namespace they are using.
 - If we determine unambiguously where the location of the service is.
 - What goes into requests and responses can be extended in the future, and those changes can be detected automatically by looking at the XML and Schemas
 - eg. we can change the data type of the response value, we can change the `<zipcode>` element, etc.
 - These are so many existing support for processing XML documents in different platforms and programming languages.
 - I can integrate the content of the messages with other XML content from other applications or domains.
 - etc - use your imagination.



Security Problems

- Security for Web Services still needs a lot of attention and development.
 - Eg. most SOAP services binds to (ie. uses) HTTP for transport. That means a whole set of behaviours can be invoked that currently firewalls cannot distinguish from normal web server-client traffic, and block appropriately.



Summary

- In the following 3 weeks, we will go into the details of SOAP, WSDL and UDDI.
- The concept of the communication is:
 - The service provider creates a WSDL description for a new SOAP service.
 - The service provider publishes the location of the WSDL description in a UDDI registry.
 - The service requestors finds the service in the registry.
 - The service requestor gets the WSDL description from the provider to discover the service details.
 - The requestor connects to the provider service using SOAP and uses the service.



Reference

- Required Reading:
 - Web Service Essentials by Ethan Cerami, O'Reilly, 2002, Chapter 1 - online book available from University library, linked from *Unit Information>Readings* page.
- Overview of technical developments and activities:
 - <http://www.w3.org/2002/ws/>