COIN-078B
Internet Programming with XML

Web Services: SOAP, WSDL, UDDI, REST

Module Objectives

- What is Web Services
- Benefits and limitations of Web Services
- The basic components: XML, SOAP, WSDL & UDDI
- The differences between SOAP and REST
- Current state of Web Services

Reading Assignments:

- **Textbook:** chapters 19 and 22
- **Teach Yourself Web Services in 24 Hours:** Part I & II

What is Web Services

W3C definition: "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."

Web Services, in its simple definition, "applications that interact with each other using internet standards" such as HTTP, XML, SOAP, WSDL and UDDI will enable system-to-system communication that is easier and cheaper than ever before. Web Services run through industry standard protocols and offer the potential of eliminating the need for proprietary hardware, software, and network protocols. By breaking down monolithic applications into software components that interface through SOAP, Web services can drive down the cost of software development, integration and maintenance. Companies will be able to lower their investment costs greatly in terms of increased ROI by implementing Web Services.

The Role of Humans

Although one of the main purposes of Web services is to automate processes that might otherwise be performed manually, humans still play a role in their architecture and use, notably in two ways

Web Services Technologies

Standard	Function
TCP/IP	Network
HTTP	Transport
XML	Data
SOAP	Messaging
WSDL	Description
UDDI	Discovery

- **TCP/IP**

  The universal networking protocol. Everything from pagers to mobile phones to laptops to mainframes can communicate through TCP/IP. Though the network can be anything else like LAN, VPN...

- **HTTP**

  The transport protocol of the Web. Other transport mechanisms such as FTP, SMTP, MQ and JMS can be used to perform this same function.

- **XML**

  All the Web Services languages are built on XML specifications. It is a great technology for moving structured data across the Web. If Web services are responsible for manipulating data in a reliable, automated way, HTML documents just don't fit the bill. If data is delivered as XML, Web services can process that data in a variety of useful ways. XML's *separation of content and presentation* is ideal.

- **SOAP**

  SOAP, originally stood for Simple Object Access Protocol, uses XML messages to invoke remote methods. SOAP is an XML document format (an envelope) that describes how to send information between endpoints in a network. The endpoint can be any platform (mainframe, PC, mobile phone). A SOAP client builds an XML document (the envelope), sends it to a host, then waits for the response. The response is another XML document that contains the results of the requested operation. A Web service could interact with remote machines through HTTP's `post` and `get` methods, but SOAP is much more robust and flexible. The SOAP envelope has two parts: the `header` that carries processing instructions and a `body` that contains the payload.

  There are two types of SOAP messages:
  - **Documents** - its payload is any XML document that can be moved from one system to another
  - **Remote Procedure Calls (RPC)** - is a method call that is intended to be executed in the Web Services provider's server. The RPC message performs the same function as an ordinary method calls in an ordinary programming language. The difference is that this call can take place over the internet.

  SOAP message example

- **WSDL** - Web Services Description Language is a specification that tells us how to describe a piece of software in terms of the method calls that it responds to. These methods are described in an abstract way that is independent of what programming language the actual service is written in or what computer and operating system it runs on. If I'm providing a service, I need to be able to describe it to the world, and if I want to use a service, I need to be able to describe what I'm looking for. WSDL was designed with this in mind.

  A WSDL document defines **services** as collections of network endpoints, or **ports**. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: **messages**, which are abstract descriptions of the data being exchanged, and **port types** which are abstract collections of **operations**. The concrete protocol and data format specifications for a particular port type constitutes a reusable **binding**. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Hence, a WSDL document uses the following elements in the definition of network services:

  - **Types**– a container for data type definitions using some type system (such as XSD).

- **Message**– an abstract, typed definition of the data being communicated.
- **Operation**– an abstract description of an action supported by the service.
- **Port Type**–an abstract set of operations supported by one or more endpoints.
- **Binding**– a concrete protocol and data format specification for a particular port type.
- **Port**– a single endpoint defined as a combination of a binding and a network address.
- **Service**– a collection of related endpoints.

See

Here's an excerpt from a WSDL document that describes a Web service:

```
<binding name="StockQuoteServiceBinding"
         type="StockQuoteServiceType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getQuote">
    <soap:operation soapAction="http://www.getquote.com/GetQuote"/>
    <input>
      <soap:body type="InMessageRequest"
                 namespace="urn:live-stock-quotes"
                 encoding="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body type="OutMessageResponse"
                 encoding="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>
</binding>
```

This is part of the definition of a stock quote service. It defines a method called `getQuote`, with the associated SOAP information that enables a piece of code to find the service, invoke a method, and process the response.

WSDL document example

- UDDI - Universal Description, Discovery and Integration is a specification for information registries of Web services. Web services themselves are discovered; a UDDI-based registry is where that discovery takes place. UDDI's approach to discovery is to have a registry of services distributed across the Web. In that distributed registry, businesses and services are described in a common XML format. The structured data in those XML documents is easily searched, analyzed, and manipulated.
  This is a specification of OASIS not W3C, and not widely implemented, therefore it will not be emphasized in this class.

Web Services SOA Operations

There are three Web services operations:

- **Publish/Unpublish**

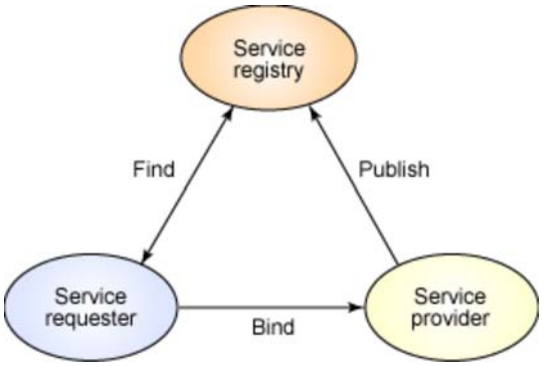
  Publishing and unpublishing involves advertising services to a registry (publishing) or removing those entries (unpublishing). The service provider contacts the service broker to publish or unpublish a service.

- **Find**

  The find operation is performed by service requestors and service brokers together. The service requestors describe the kinds of services they're looking for, and the service brokers deliver the results that best match the request.

- **Bind**

  The bind operation takes place between the service requestor and the service provider. The two parties negotiate as appropriate so the requestor can access and invoke services of the provider.



What is REST

Representational State Transfer (REST) is a term coined by Roy Fielding, one of the principal authors of the HTTP specification, in his doctoral dissertation in 2000 to describe an architecture style of networked systems such as the World Wide Web.

REST is not a standard. It strictly refers to a collection of network architecture principles that outline how resources are defined and addressed. The term is often used in a looser sense to describe any simple interface that transmits domain-specific data over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies.

An important concept in REST is the existence of resources (sources of specific information), each of which can be referred to using a global identifier (a URI). In order to manipulate these resources, components of the network (clients and servers) communicate via a standardized interface (e.g. HTTP) and exchange representations of these resources (the actual documents conveying the information). Thus an application can interact with a resource by knowing two things: the identifier of the resource, and the action required — it does not need to know whether there are caches, proxies, gateways, firewalls, tunnels, or anything else between it and the server actually holding the information. The application does, however, need to understand the format of the information (representation) returned, which is typically an HTML or XML document of some kind, although it may be an image or any other content. The representation places the client application in a state. The result of the client traversing a hyperlink in one resource is another resource being accessed. The new representation places the client application into yet another state. Thus, the client application changes (transfers) state with each resource representation, hence the term "Representational State Transfer"!

RESTful example: the World Wide Web

The World Wide Web is the key example of a RESTful design. Much of it conforms to the REST principles. The Web consists of the HTTP, content types including the HTML, and other Internet technologies such as the Domain Name System (DNS). HTML can include JavaScript and applets to support code on demand, and has implicit support for hyperlinks. HTTP has a uniform interface for accessing resources, which consists of URIs, methods, status codes, headers, and content distinguished by MIME type. The most important HTTP methods - POST, GET, PUT and DELETE are often compared with the CREATE, READ, UPDATE, DELETE (CRUD) operations associated with database technologies

REST versus RPC Web Services

	REST	RPC
Commands	defined in simple terms: resources to be retrieved, stored / get, set — difficult to do many joins	defined in methods with varying complexity depending on "standard" - easier to hide complex

		things behind a method
Exchange	Nouns — exchanging resources and concepts	Verbs — exchanging methods
Example	http://example.com /users/user	getUser()

Specifications:

- W3C: Web Services Architecture
- W3C: SOAP Version 1.2 Primer
- W3C: Web Services Description Language (WSDL) Version 1.1
- OASIS: UDDI Version 2 Specifications
- WS-I: Basic Profile Version 1.0a

General Resources:

- A Web Services Primer
- W3Schools: SOAP Tutorial
- W3Schools: WSDL Tutorial
- Web Services/PHP Tutorial
- An introduction to SOAP basics
- Yahoo! Directory on SOAP
- MSDN Webcast: Building and Consuming Web Services
- Building Web Services the REST Way
- XMethods Webservices Directory

Vendors Resources

- Java Technology and Web Services
- IBM Developer Works: Web Services
- Microsoft Web Services Developer Center
- Apache Web Services - Axis

Hands-on Assignment:

- **Final Project** due **Week 12**

© Copyright 1999-2010 Newton Chan. All Rights Reserved.