WAVV 2004
Chattanooga, TN

XML and Web Services 201

WAVV 2004

Chuck Arney
illustro Systems International LLC

Handouts

- Download a copy of this presentation

  www.illustro.com/wavv2004

WAVV2004-2

Agenda

- What is a Web Service and who defines the specifications
- Technologies that are used by Web Services
- Live demonstrations
  - Creating a Web Service
  - Consuming a Web Service
- Other considerations
- Web Service Resources

WAVV2004-3

Definitions of a Web Service

- 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-process able 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.

Definitions of a Web Service

- Some kind of function you can use across some kind of network
  - Usually involve short interactions (connect, access code, disconnect)
  - Discoverable, maybe …
  - Described with XML

- A Web Service is the exposure of a business process over a network

Who Defines Web Services?

- Open System Standards Organizations
  - World Wide Web Consortium (W3C)
    www.w3c.org

  - Web Services Interoperability Organization (WS-I)
    www.ws-i.org

Who Defines Web Services?

- **Software Vendors**
  - IBM Corp.
    www.ibm.com/developerworks/webservices

  - Microsoft Corp.
    www.microsoft.com/webservices

  - Hewlett-Packard Company
    devresource.hp.com/drc/topics/web_services.jsp

Components of a Web Service

- **Service Provider**
  - Delivers services across the network
  - Publishes their services to a broker
- **Service Requestor**
  - Asks the broker for a service
  - Binds to the provider once its found
- **Service Broker**
  - Matchmaker between providers and requestors

Operations of a Web Service

- **Publish**
  - Service providers advertise (or not) their services with a service broker
- **Find**
  - Service requestors ask the broker for a service that meets certain criteria
- **Bind**
  - Service requestor bind to the service provider and transaction ensue

Behind the Technology

- XML
  - EVERYTHING is in XML
- SOAP
  - Simple Object Access Protocol
  - Protocol to make requests and receive result
- WSDL
  - Web Service Description Language
  - Describes the available services, inputs and outputs

Behind the Technology

- UDDI
  - Universal Description, Discovery and Integration
  - Data base of available services (gets you to the WSDL)
  - API to search and update the data base

Data Transports

- Web Service protocols do not define the physical (wire) or logical (application layer) protocol used to transmit requests and results
- Typically the TCP/IP protocols used are
  - SMTP
  - HTTP

Data Transports

- Simple Mail Transfer Protocol (SMTP)
  - Early TCP/IP protocol
  - Multipurpose Internet Mail Extensions (MIME) support allows non-textual data
- Problems with SMTP
  - Non-textual data must be encrypted
  - Security problems
  - Data size limitations

Data Transports

- Hypertext Transfer Protocol (HTTP)
  - Most used protocol for Web Services
  - Newer protocol (1990s)
  - Supports binary data
  - Request headers more extensible
  - Request/response model
- Other protocols are possible
  - Internet Inter-ORB Protocol (IIOP)
  - IBM MQSeries

eXtensible Markup Language XML

- Platform independent way to share data
- Markup language similar to SGML
- Tags describe the meaning and purpose of the data
- Data only, no presentation attributes
- Used for all documents, requests and responses in Web Services architecture
- Details at www.w3c.org

SOAP Design

- Vendor neutral
- Language neutral
- Platform neutral
- Transport neutral

SOAP Server

- Usually a Web Service lives behind a SOAP Server or Web Server that understands SOAP requests
  - Handles the transport protocol (HTTP)
  - Processes the SOAP request document
  - Calls processing program passing input data
  - Creates and returns result message

SOAP Messages

- There are request and response messages
  - A request invokes a method on a remote object
  - A response returns the result of running the method
- Requests can use either Remote Procedure Call or Document style messages

SOAP Messages

- Most Web Services use RPC style messages
  - SOAP specification defines how calls and returns are serialized in messages

- Document style messages can transport any XML document as long as the SOAP Server knows what to do with it

SOAP Envelopes

- A SOAP envelope contains the message itself
- The message is in an application specific vocabulary; namespaces distinguish the parts
- The envelope can contain a header, and MUST contain a body

SOAP Request

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="…"
    SOAP-ENV:encodingStyle="…">
    <SOAP-ENV:Body>
        <m:getLastTradePrice
            xmlns:m="my-ns">
            <symbol>IBM</symbol>
        </m:getLastTradePrice>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response

- **<SOAP-ENV:Envelope**
  xmlns:SOAP-ENV="..."
  SOAP-ENV:encodingStyle="...">
  <SOAP-ENV:Body>
    <m:getLastTradePriceResponse
        xmlns:m="my-ns">
        <price>91.77</price>
    </m:getLastTradePriceResponse>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

Live Demo

Web Services Description Language

- If we're going to find and use web services automatically, we have to have a way to describe them
  - WSDL is used to describe a web service when it is published
  - When a service is located, WSDL is used to invoke it

Web Services Description Language

- The WSDL file contains everything needed to make a SOAP request
  - The address of the machine that hosts the service
  - The name of the service
  - The name of the method
  - The name and types of the arguments to the method

WAVV2004-25

SOAP and WSDL

- To make a SOAP request we have to know the arguments
- Start out looking in the WSDL <portType>

```
<portType name="GreetingWSSOAP">
    <operation name="GetGreeting">
        <input
message="s0:GetGreetingSOAPIn" />
```

- This says use **GetGreetingSOAPIn** message as the input to the method

WAVV2004-26

SOAP and WSDL

- Now look at the **GetGreetingSOAPIn** message

```
<message name="GetGreetingSOAPIn">
 <part name="parameters"
   element="GetGreeting" />
</message>
```

- This message references the **GetGreeting** data type defined earlier

WAVV2004-27

Illustro
SYSTEMS INTERNATIONAL, LLC
See The Light.™

SOAP and WSDL

- Now look at the **GetGreeting** data type to see what goes in the method call

```
<s:element name="GetGreeting">
   <s:complexType>
        <s:element minOccurs="1"
        maxOccurs="1"
   name="greetingNumber"
   type="s:int" />
   </s:complexType>
</s:element>
```

SOAP and WSDL

- In reality, we will write our client code to a known programming interface
  - We will know the method name and number and type of arguments ahead of time

- But, this information in the WSDL will be used by our tools to automatically generate proxy classes for us

Universal Description, Discovery and Integration

- Web Service discovery
- Defines standards for distributed registry of Web Services
  - White pages   company information
  - Yellow pages   categories of services
  - Green pages   business rules

UDDI Registries

- IBM. Microsoft, SAP, NTT and others provide registries of Web Services
  - See www.uddi.org/find.html for access
- See www.ibm.com/services/uddi for information on IBM's registry
- You can register for no charge and publish your own Web Services

UDDI Registries

- Registration basic process
  - Split your WSDL file
  - Register the interface portion as a TModel
  - Describe your business with a BusinessEntity
  - Add a BusinessServices record to describe your service. Refers to BusinessEntity and TModel
  - Install the WSDL file on your web server

UDDI Process

- The UDDI discovery process
  - Client queries the UDDI registry for a service and receives the result
  - Client uses the results to request the WSDL file from the service provider's web server
  - Client invokes the service using the information from the WSDL file and receives the result from the service

UDDI4J

- IBM has released UDDI4J, a free, open source implementation of the client side of UDDI.
  - Simplifies the task of interacting with a UDDI registry
  - JAVA Class Library
  - See www-124.ibm.com/developerworks/projects/uddi4j

Recommendations

- When starting to use Web Services
  - Start with a pilot project to prove the value up front
    - Start small, grow fast
  - Do something representative of your larger goal
  - Goals can be achieved by a small team in 2-3 months
  - Define goals, expectations and measures of success before starting

Other Considerations

- Web Services is not the only method to achieve the result
  - Pluses
    - Open standards
    - Easy to use with supporting tools
    - Impressive technology
  - Minuses
    - More technologies to master
    - Serialization of passed values is overhead
    - Maybe more difficult to debug when there are problems

Other Considerations

- J2EE and .NET support direct XML handling
  - XML parser APIs (DOM and SAX)
  - Stream readers and writers (XMLTextReader/XMLTextWriter)
- They also support sockets and even HTTP
- XML documents can be sent/received and processed without SOAP overhead

Copyright © 2004 illustro Systems International, LLC   WAVV2004-37

Resources

- World Wide Web Consortium
  - WWW.W3C.org

- Web Services Interoperability Organization
  - WWW.WS-I.org

- OASIS
  - www.oasis-open.org
  - www.uddi.org

Copyright © 2004 illustro Systems International, LLC   WAVV2004-38

Resources

- IBM developerWorks
  - Ibm.com/developerworks/webservices
  - Ibm.com/developerworks/speakers/dtidwell
  - Ibm.com/developerworks/speakers/colan
- IBM Redbooks
  - redbooks.ibm.com
- IBM alphaWorks (WSTK)
  - www.alphaworks.Ibm.com/webservices

Copyright © 2004 illustro Systems International, LLC   WAVV2004-39

Resources

- Apache Web Services Project
  - ws.apache.org

- LOTS of books are available
  - Professional XML Web Services
    Wrox Press, ISBN 1-861005-09-1
  - Developing .NET Web Services with XML
    Syngress Publishing, ISBN 1-928994-81-4