WebServices

Bob Brown

transentia pty. ltd.

bob@transentia.com.au

http://www.transentia.com.au

What are WebServices?

- WebServices
  - *"Dymanic Services"*
    - *Oracle Technology Network (typo?)*
  - *"An XML-based grammar for specifying the properties of a web service"*
  - *"A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented. Web Services combine the best aspects of component-based development and the Web..."*
    - A way of specifying an addressable entity: what it does, where it is, how to invoke it, etc...

Quotes

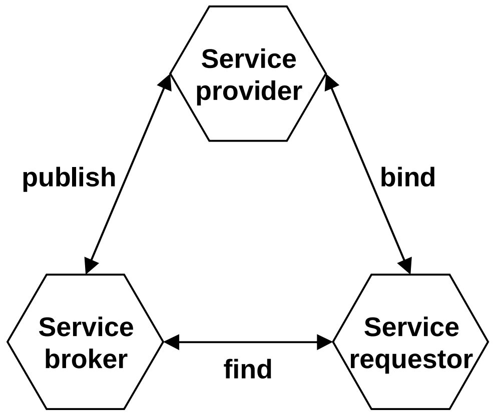
- *"Shrink-wrapped software is dead. Long live software as a service"*
  - *Web services: Few actually deliver*, Mary Jo Foley, CNET News.com
    - *".Net or .Nyet?"*
- *"Webservices are pretty much guaranteed to be at the heart of the next generation of distributed systems"*
  - Graham Glass, The Mind Electric
- *"The Internet is quickly evolving from today's Web sites that just deliver UI Pages to browsers to a next generation of programmable Web sites that directly link organizations, applications, services, and devices with one another. These programmable Web sites become more than passively accessed sites - they become reusable, intelligent Web Services."*
- *"The dot-com meltdown provides stark evidence that the Internet business model needs a reboot. Giving away services for free and making it up on volume just isn't a sustainable method for running a business."*
  - *Microsoft*

Quotes…

- *“Web services are a new model for creating dynamic distributed applications with common interfaces for efficient communication across the In- ternet. They're built around ubiquitous, open standards such as TCP/IP, HTTP, Java, HTML, and XML, as well as newer standard technologies such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI).”*
  – Jack Martin, *WebServices Journal*
- *“The cool thing about having a standard such as SOAP and WSDL that supersedes any specific technology standard (EJB, DCOM) is that an application can be built without regard to the underlying technology implementation.”*
  – Steve Benfield, *WebServices Journal*

IBM's Vision

- A *"Service Oriented Architecture"*
  - CORBA-like
    - Service providers *publish* services to a service broker. Service requestors *find* required services using a service broker and *bind* to them



- *"The next generation of e-business focusing on the integration and infrastructure complexities of B2B by leveraging the benefits of internet standards and common infrastructure to produce optimal efficiencies for intra- and inter-enterprise computing"*

IBM's Vision…

- *"One of the main ideas behind Web services is that applications of the future will be assembled from a collection of networked-enabled services. As long as two equivalent services are able to advertise themselves to the network in a standard and neutral way, an application could theoretically choose between alternative competing services based on criteria such as price or performance. In addition, some services could allow themselves to be copied between machines, thereby allowing an application running on a particular computer (or cluster) to improve its performance by copying useful services into its local storage."*

Microsoft's 'Hailstorm'

- *"HailStorm can be accessed from any device, service or application with an Internet connection, the ability to authenticate a user, and the ability to send and receive SOAP messages. All interactions are text-based SOAP messages, regardless of underlying platform, operating system, object model, programming language, application or online service. HailStorm is accessible from both clients and servers, and no Microsoft runtime is required on either the client or the server. "*
- Founded in .NET; associated with Passport
- Initial set of HailStorm services include:
  - myAddress - electronic and geographic address for an identity
  - myProfile - name, nickname, special dates, picture
  - myContacts – electronic relationships/address book
  - myLocation - electronic and geographical location and rendez-vous
  - myNotifications – notification subscription, management and routing
  - myInbox - inbox items like e-mail and voice mail, including existing mail systems
  - myCalendar – time and task management
  - myDocuments – raw document storage
  - myApplicationSettings - application settings
  - myFavoriteWebSites – favorite URLs and other Web identifiers
  - myWallet - receipts, payment instruments, coupons and other transaction records
  - myDevices – device settings, capabilities

Example WebServices

- a credit checking service that returns credit information, given a social security number
- A stock quote service that returns the stock price associated with a specified ticker symbol
- A purchasing service that allows computer systems to buy office supplies when given an item code and a quantity
- A travel booking service that orchestrates other car/plane/hotel room rental services
- Soda supplies getting low? No problem, the wireless-networked soda machine can contact the local supplier's WebService and order more of your favourite beverage…
- Etc.
  - Where once were Servlets/{J,A}SPs/Session EJBs, think about exposing WebServices

Features

- Interoperability
  - Thanks to SOAP, any WebService can interact with any other WebService
- Ubiquity
  - WebServices communicate using HTTP and XML. Any device which supports these technologies can both host and access WebServices
- Low barrier to Entry
  - Concepts are easy to understand; free toolkits allow developers to quickly create/deploy Web services. Some of these toolkits allow pre-existing COM/ JavaBeans components to be exposed as WebServices
- Industry Support
  - All major vendors are supporting SOAP/WebServices. It should be easy for components written in Visual Basic to be deployed as WebServices, and consumed by services written using IBM VisualAge, and vice-versa

Issues

- Not everything is worked out yet but solutions do exist in other spheres, so the task should be manageable and should progress quickly
- Some challenges:
  - Discovery/advertising
    - WSDL (Web Services Definition Language) and UDDI (Universal Description, Discovery and Integration)
  - Reliability/failover/scalability
    - Does there need to be a new kind of “Web service” application server?
  - Security
    - HTTP over SSL will provide basic security, but individual services will need a higher level of granularity
    - How does a Web service authenticate users?
    - Do services need to be able to provide security at the method level?

Issues…

– Transactions
  - Two-phase commit works fine in a closed environment where transactions are short-lived, but doesn't work well in an open environment where transactions can span hours or even days
    – Microsoft's "compensating transactions"/standards like XAML may help but there is currently enough activity to muddy the waters

– Manageability
  - Say goodbye to the days of the central piece of iron, or even the centrally managed network

– Profitability model
  - Outright $$$, subscription-based, or pay-as-you-go?

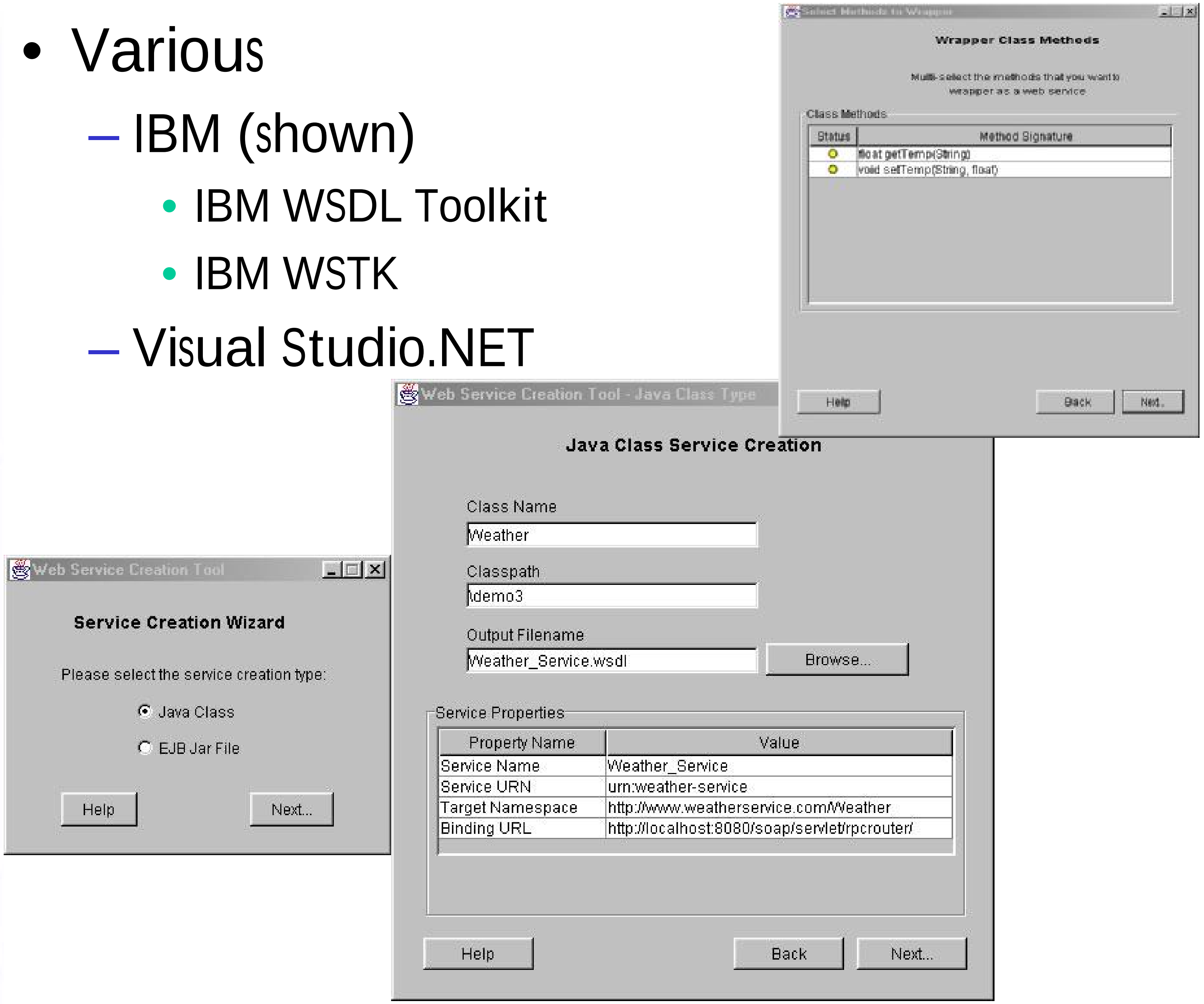
– Testing
  - When a system is comprised of many WebServices whose location and qualities are potentially dynamic, testing and debugging takes on a whole new dimension!
    – How do you achieve predictable response times? How do you debug WebServices that can come from different vendors, hosted in different environments and on different operating systems?
    – *"A distributed system is one where a machine on a remote network that you have never heard of can crash your box!"*

Introducing WSDL

- WebServices Definition Language
  - The XML equivalent of a resumé
    - Describes what a service can do, where it is, and how to invoke it
  - *"WSDL can be seen as a complement to SOAP, as it facilitates interoperability between Web services. Like IDL (Interface Definition Language), which acts as a service describer with CORBA, WSDL (Web Service Description Language) is an XML syntax to describe Web services. The specifications for WSDL come from a joint initiative by Microsoft, IBM and Ariba. More and more SOAP implementations support this description language; with WSDL, applications that use SOAP can self-configure exchanges between Web services, while hiding most of the low-level technical details."*
    - ALAIN LEFEBVRE, Vice President, Groupe SQLI (TechMetrix)

WSDL Toolkits

- Various
  - IBM (shown)
    - IBM WSDL Toolkit
    - IBM WSTK
  - Visual Studio.NET



Select Methods to Wrapper

Wrapper Class Methods

Multi-select the methods that you want to wrapper as a web service

Class Methods

Status	Method Signature
●	float getTemp(String)
●	void setTemp(String, float)

Web Service Creation Tool

Service Creation Wizard

Please select the service creation type:

(● Java Class) ○ EJB Jar File

Web Service Creation Tool - Java Class Type

Java Class Service Creation

Class Name: Weather

Classpath: \demo3

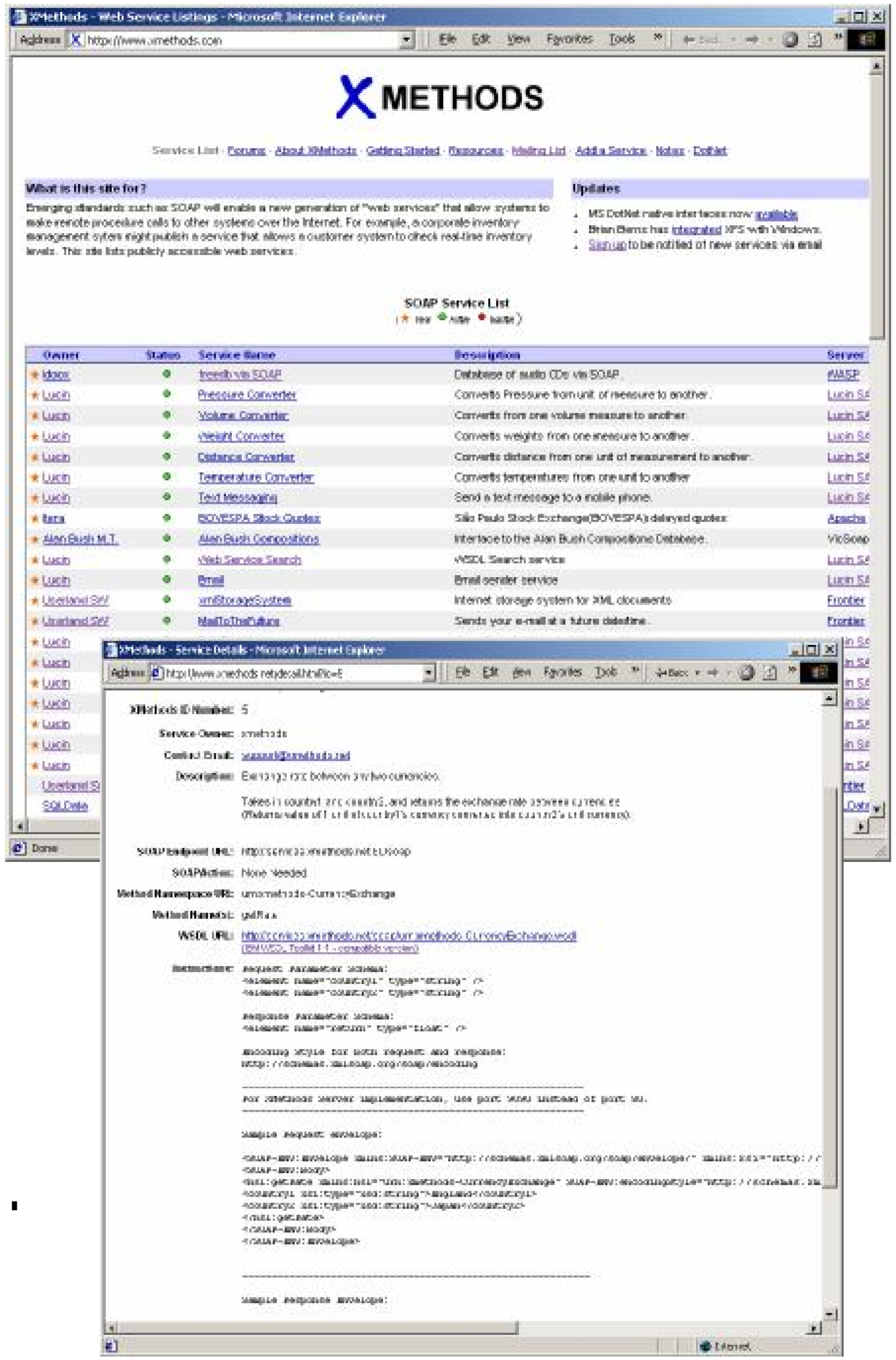
Output Filename: Weather_Service.wsdl

Service Properties

Property Name	Value
Service Name	Weather_Service
Service URN	urn:weather-service
Target Namespace	http://www.weatherservice.com/Weather
Binding URL	http://localhost:8080/soap/servlet/rpcrouter/

WebServices Brokers

- Brokers
  - Xmethods
  - Biztalk
  - ebXML
  - RosettaNET
  - Etc.
  - *“the nice thing about standards is that there are so many to choose from…”*
- More under UDDI…


XMethods - Web Service Listings - Microsoft Internet Explorer

Address http://www.xmethods.com

Service List · Forums · About XMethods · Getting Started · Resources · Mailing List · Add a Service · Notes · DotNet

What is this site for?

Emerging standards such as SOAP will enable a new generation of "web services" that allow systems to make remote procedure calls to other systems over the Internet. For example, a corporate inventory management sytem might publish a service that allows a customer system to check real-time inventory levels. This site lists publicly accessible web services.

Updates

- MS DotNet native interfaces now available
- Brian Berns has integrated XPS with Windows.
- Sign up to be notified of new services via email

SOAP Service List
(★ new ● active ● inactive)

Owner	Status	Service Name	Description	Server
★ Idoox	●	freedb via SOAP	Database of audio CDs via SOAP.	WASP
★ Lucin	●	Pressure Converter	Converts Pressure from unit of measure to another.	Lucin SA
★ Lucin	●	Volume Converter	Converts from one volume measure to another.	Lucin SA
★ Lucin	●	Weight Converter	Converts weights from one measure to another.	Lucin SA
★ Lucin	●	Distance Converter	Converts distance from one unit of measurement to another.	Lucin SA
★ Lucin	●	Temperature Converter	Converts temperatures from one unit to another	Lucin SA
★ Lucin	●	Text Messaging	Send a text message to a mobile phone.	Lucin SA
★ Ilera	●	BOVESPA Stock Quotes	São Paulo Stock Exchange(BOVESPA) delayed quotes	Apache
★ Alan Bush M.T.	●	Alan Bush Compositions	Interface to the Alan Bush Compositions Database.	VicSoap
★ Lucin	●	Web Service Search	WSDL Search service	Lucin SA
★ Lucin	●	Email	Email sender service	Lucin SA
★ Userland SW	●	xmlStorageSystem	Internet storage system for XML documents	Frontier
★ Userland SW	●	MailToTheFuture	Sends your e-mail at a future date/time.	Frontier
★ Lucin				Lucin SA
★ Lucin				Lucin SA
★ Lucin				Lucin SA
★ Lucin				Lucin SA
★ Lucin				Lucin SA
★ Lucin				Lucin SA
★ Lucin				Lucin SA
Userland S				Frontier
SQLData				SQLData

Done

XMethods - Service Details - Microsoft Internet Explorer

Address http://www.xmethods.net/detail.html?id=5

XMethods ID Number: 5

Service Owner: xmethods

Contact Email: support@xmethods.net

Description: Exchange rate between any two currencies.

Takes in country1 and country2, and returns the exchange rate between currencies (Returns value of 1 unit of country1's currency converted into country2's unit currency).

SOAP Endpoint URL: http://services.xmethods.net:80/soap

SOAPAction: None Needed

Method Namespace URI: urn:xmethods-CurrencyExchange

Method Name(s): getRate

WSDL URL: http://services.xmethods.net/soap/urn:xmethods-CurrencyExchange.wsdl
(IBM WSDL Toolkit 1.1 - compatible version)

Instructions:

```
request parameter schema:
<element name="country1" type="string" />
<element name="country2" type="string" />

response parameter schema:
<element name="return" type="float" />

encoding style for both request and response:
http://schemas.xmlsoap.org/soap/encoding
------------------------------------------------------------
For XMethods server implementation, use port 9090 instead of port 80.
------------------------------------------------------------

sample request envelope:

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://
<SOAP-ENV:Body>
<ns1:getRate xmlns:ns1="urn:xmethods-CurrencyExchange" SOAP-ENV:encodingStyle="http://schemas.xm
<country1 xsi:type="xsd:string">England</country1>
<country2 xsi:type="xsd:string">Japan</country2>
</ns1:getRate>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

------------------------------------------------------------

sample response envelope:
```

Internet

Example: Java Service

- Simplicity itself!
  - J

```
public class Exchange
  {
  public float getRate (String country1, String country2)
    {
    return 144.52F; // always return the same value for now
    }
  }
```

Example: Java Client

```
import java.net.*;
import java.util.*;
import org.apache.soap.*; // Body, Envelope, Fault, Header
import org.apache.soap.rpc.*; // Call, Parameter, Response

public class Client
  {
  public static void main (String [] args) throws Exception
    {
    URL url = new URL ("http://localhost:8080/soap/servlet/rpcrouter");
    String urn = "urn:demo1:exchange";

    Call call = new Call (); // prepare the service invocation
    call.setTargetObjectURI (urn);
    call.setMethodName ("getRate");
    call.setEncodingStyleURI (Constants.NS_URI_SOAP_ENC);
    Vector params = new Vector ();
    params.addElement (new Parameter ("country1", String.class, "USA", null));
    params.addElement (new Parameter ("country2", String.class, "japan", null));
    call.setParams (params);
    try
      {
      System.out.println ("invoke service\n" + "  URL= " + url + "\n  URN =" + urn);
      Response response = call.invoke (url, ""); // invoke the service
      if ( ! response.generatedFault ())
        {
        Parameter result = response.getReturnValue (); // response was OK
        System.out.println ("Result= " + result.getValue ());
        }
      else
        {
        Fault f = response.getFault (); // an error occurred
        System.err.println ("Fault= " + f.getFaultCode () + ", " + f.getFaultString ());
        }
      }
    catch (SOAPException e) // call could not be sent properly
      {
      System.err.println ("SOAPException= " + e.getFaultCode () + ", " + e.getMessage ());
      }
    }
  }
```

Example: WSDL File

```
<?xml version = "1.0"?>
<definitions name = "CurrencyExchangeService"
             targetNamespace = "http://www.xmethods.net/sd/CurrencyExchangeService.wsdl"
             xmlns:xsd = "http://www.w3.org/1999/XMLSchema"
             xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns = "http://schemas.xmlsoap.org/wsdl/">
  <message name = "getRateRequest">
    <part name = "country1" type = "xsd:string"/>
    <part name = "country2" type = "xsd:string"/>
  </message>
  <message name = "getRateResponse">
    <part name = "return" type = "xsd:float"/>
  </message>
  <portType name = "CurrencyExchangePortType">
    <operation name = "getRate">
      <input message = "getRateRequest" name = "getRate"/>
      <output message = "getRateResponse" name = "getRateResponse"/>
    </operation>
  </portType>
  <binding name = "CurrencyExchangeBinding" type = "CurrencyExchangePortType">
    <soap:binding style = "rpc" transport = "http://schemas.xmlsoap.org/soap/http"/>
    <operation name = "getRate">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use = "encoded" namespace = "urn:xmethods-CurrencyExchange"
                   encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"/>
      </input>
      <output>
        <soap:body use = "encoded" namespace = "urn:xmethods-CurrencyExchange"
                   encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>
  </binding>
  <service name = "CurrencyExchangeService">
    <documentation>Returns the exchange rate between the two currencies</documentation>
    <port name = "CurrencyExchangePort" binding = "CurrencyExchangeBinding">
      <soap:address location = "http://services.xmethods.net:80/soap"/>
    </port>
  </service>
</definitions>
```

Example: WSDL File…

- Contains various sections. Salient points:
  - definitions
    - contains the definition of one or more services
  - message
    - a single piece of information moving between the invoker and the service. A regular round trip remote method call is modelled as two messages, one for the request and one for the response
  - portType
    - a set of one or more input/output message sequences
  - binding
    - a <portType> implemented using a particular protocol such as SOAP or CORBA
  - service
    - a collection of a particular binding at a specific end-point
- Since WSDL fully defines a service, it can be used to auto-generate RPC stub/marshalling/proxying code
  - Most toolkits allow for this, actually

Wrapup

- *"Even though the WebServices actually available at the moment are few and far between and rather basic in nature, professional offerings with more complete solutions are beginning to emerge…"*
- *"Four years ago, we all saw the rush to develop Web sites on the Internet. Today, we will witness the rush to set up Web Services over the Internet."*
- *"WebServices are becoming the programmatic backbone for electronic commerce"*
- *"So far, IBM has yet to deliver Web services themselves, and it's unclear if the company intends to do so. Stay tuned in 2001."*
- *'Microsoft needs more of a cross-application strategy if it's serous about proving that it's a Web services, not just a "Windows services," company…'*
- *'HP…was one of the first companies to outline a comprehensive vision for Web services with its "E-speak" application. Recently, however, the company has focused less on the grandiose vision and more on the concrete delivery details. The best evidence of that was a late-year decision to join backers of the UDDI standard, a proposed Yellow Pages for companies that want to do business electronically.'*

Resources

- The True Nature of Web Services
  - http://www.techmetrix.com/trendmarkers/tmk0501/tmk0501-2.php3
- Web services: Few actually deliver
  - http://news.cnet.com/news/0-1003-201-4298495-0.html
- The Web services (r)evolution (4 parts)
  - ftp://www6.software.ibm.com/software/developer/library/ws-peer{1,2,3,4}.pdf
- Visual Studio .NET: Build Web Applications Faster and Easier Using Web Services and XML
  - http://msdn.microsoft.com/msdnmag/issues/0900/vsnet/vsnet.asp
- Web Services: Building Reusable Web Components with SOAP and ASP.NET
  - http://msdn.microsoft.com/msdnmag/issues/01/02/webcomp/webcomp.asp
- Your first C# Web Service
  - http://www.codeproject.com/webservices/myservice.asp
- Web Services Interoperability and SOAP
  - http://msdn.microsoft.com/xml/general/soapinteropbkgnd.asp