WebServices

Pieter Bruegel's The Tower of Babel

Bob Brown

transentia

bob@transentia.com.au

http://www.transentia.com.au

What are WebServices?

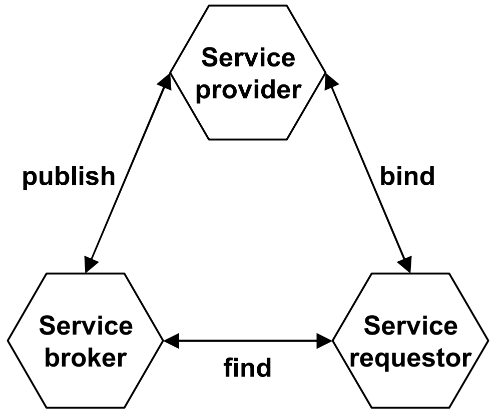
- WebServices
  - *"Dymanic Services"*
    - *Oracle Technology Network (typo?)*
  - *"An XML-based grammar for specifying the properties of a web service"*
  - *"A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous 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…

"There are two things that make Web services different: the protocol stack is now ubiquitous… and Microsoft is finally at the table."

Quotes

- *"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 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*
- *"Web services are a new model for creating dynamic distributed applications with common interfaces for efficient communication across the Internet. 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*

- 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' Idea

- *"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
- Now defunct
  - will reappear in some guise or other
    - from many vendors

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

- A frightening thought but…
  - This example defines an ASP+ WebService
    - Defines a method ADDME

```
<%@ webservice language="COBOL" %>
CLASS-ID. FOO.
FACTORY.
PROCEDURE DIVISION.
METHOD-ID. ADDME.
DATA DIVISION.
LINKAGE SECTION.
01 OPND-1 PIC S9(9) COMP-5.
01 OPND-2 PIC S9(9) COMP-5.
01 RET PIC S9(9) COMP-5.
PROCEDURE DIVISION USING BY VALUE OPND-1 OPND-2 RETURNING RET.
COMPUTE RET = OPND-1 + OPND-2.
END METHOD ADDME.
END FACTORY.
END CLASS FOO.
```

Features

Feature	RPC	WebServices
Programming language independent	No	Yes
Binding to different transports	No	Yes
Firewall friendly	No	Yes
Loosely coupled contract	No	Yes
Asynchronous support	No	Yes
Space/time efficiency	Yes	No

- 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?

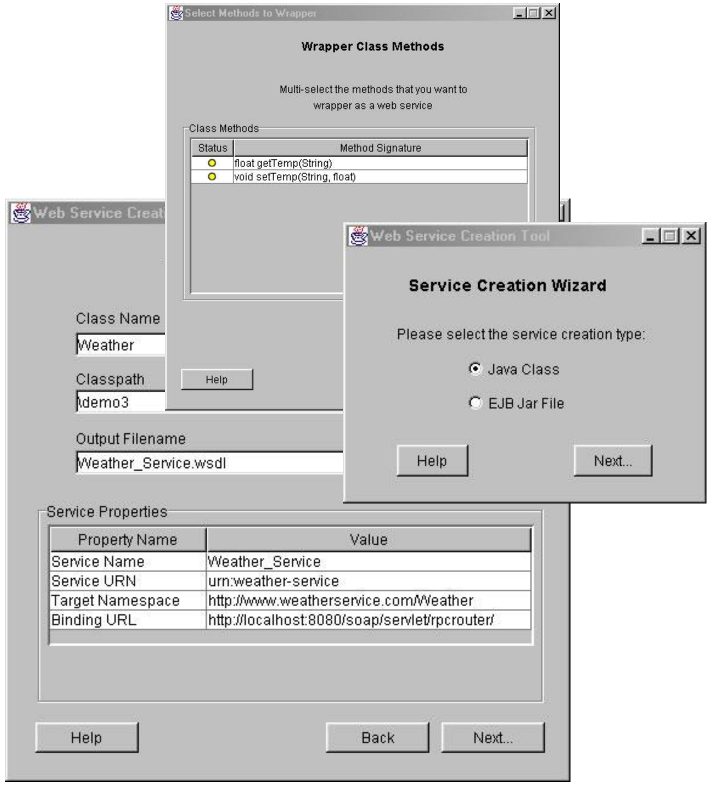
Issues…

- 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!"*

- 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)

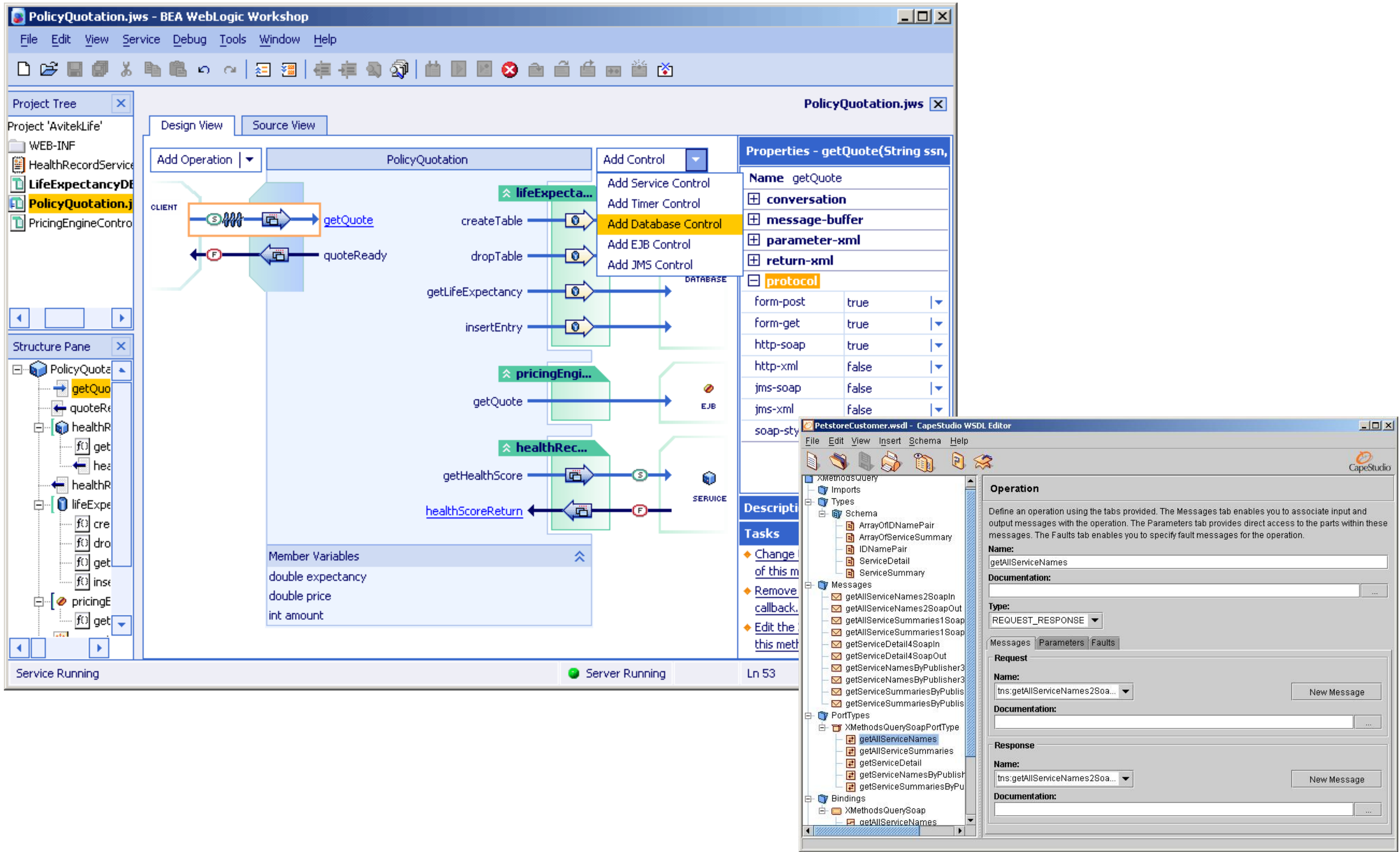
Tools

- Various, including:
  - IBM (shown)
    - IBM WSDL Toolkit
    - IBM WSTK
  - Visual Studio.NET
  - JDeveloper 9i
  - XML Spy
  - MS SOAP Toolkit 2.0
  - BEA WLS 7
    - Provides a number of ant tasks to enable automation
      - http://edocs.bea.com/wls/docs70/webserv/assemble.html
  - Cape Clear WSDL Editor
    - *"is to Web Services development what WYSIWYG HTML editors were to Web page development"*



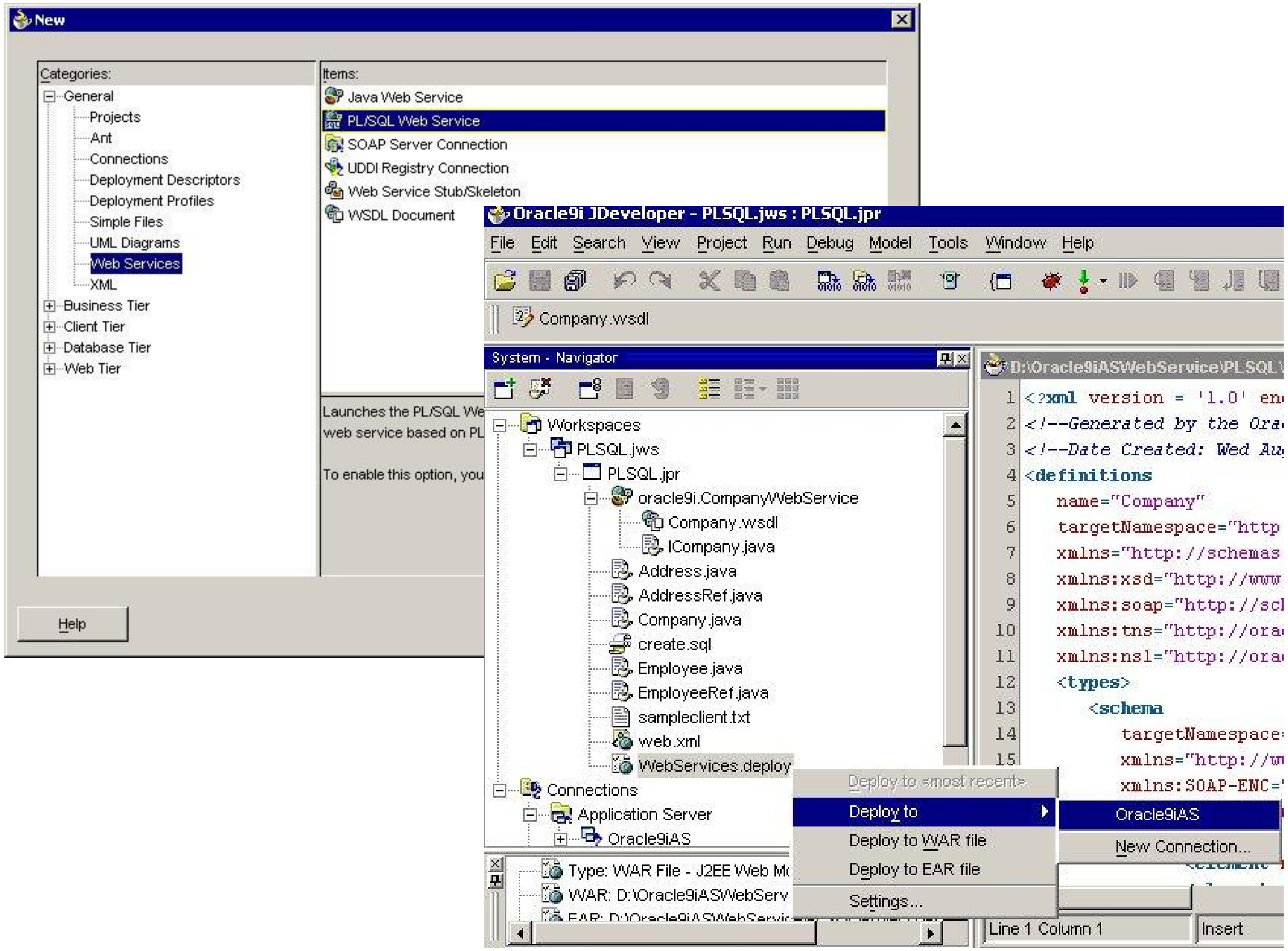
Tools…

- Capeclear's 'CapeStudio'
- BEA Weblogic Workshop
  - With WLS 7.0



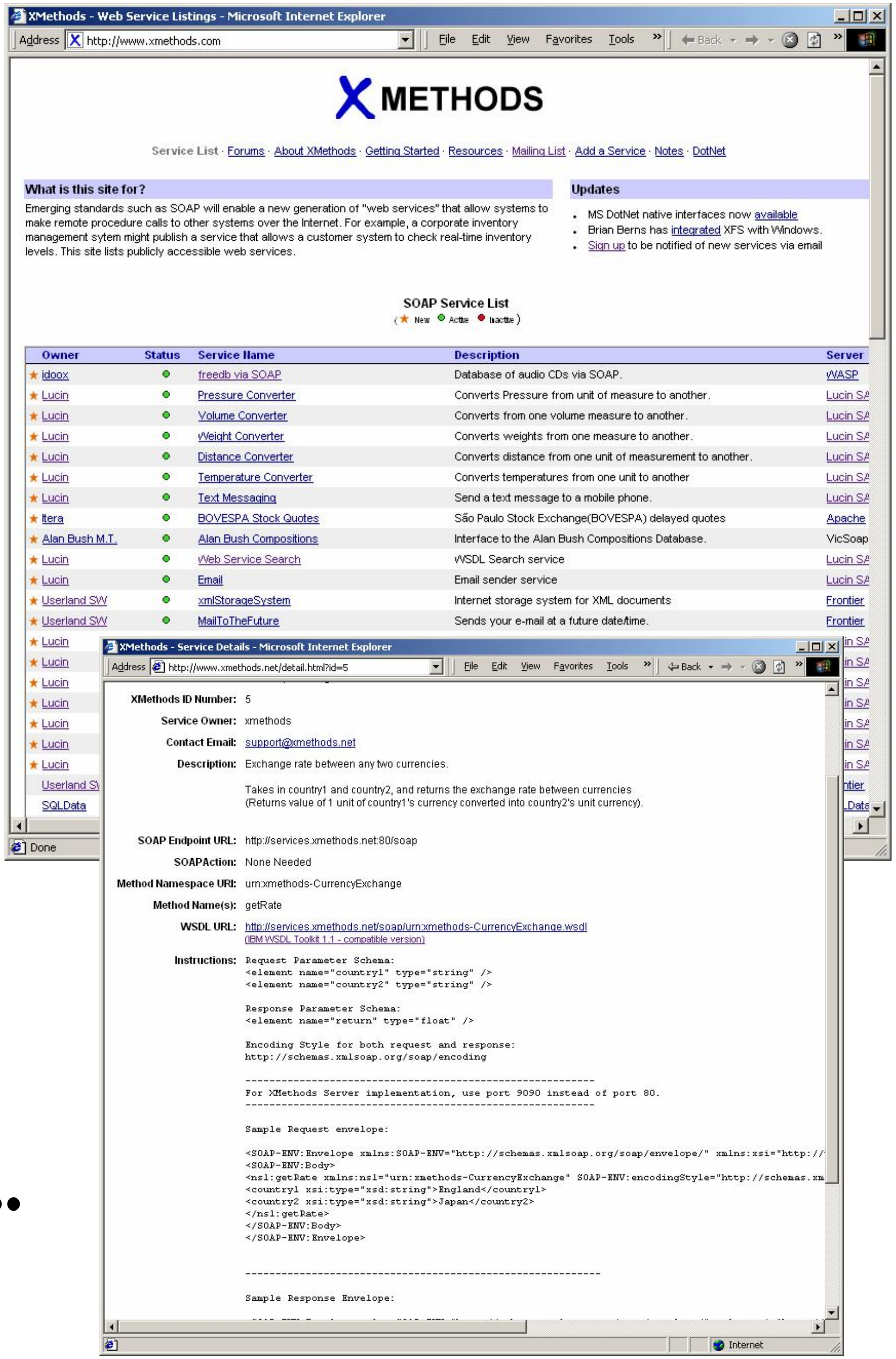
Tools...

- JDeveloper 9i
  - Provides extensive support for WebServices
    - Almost anything can be published...



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...



- Simplicity itself!
  - ☺

```
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 ());
      }
    }
  }
```

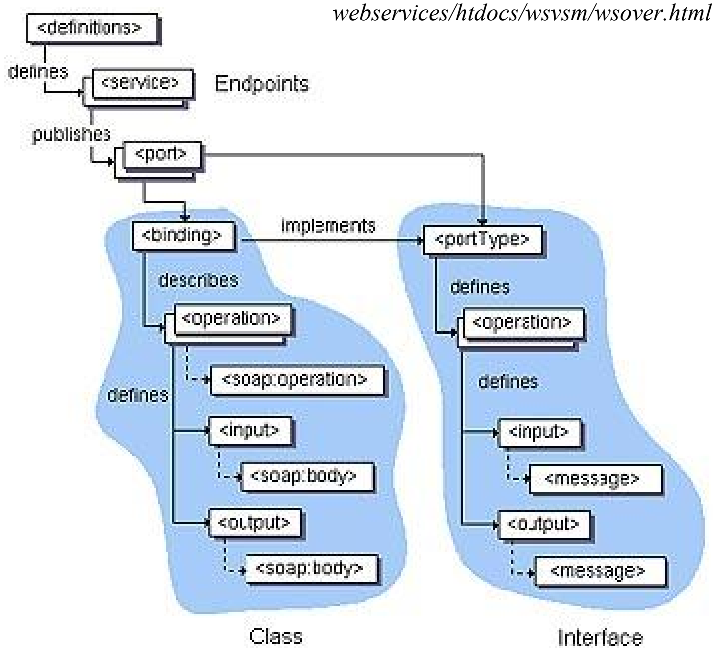
Example: WSDL File

```xml
<?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
  - Key: self-configuring interactions

http://otn.oracle.com/tech/ webservices/htdocs/wsvsm/wsover.html



WSDL 1.2

- Working draft 2003-06-11
- Update for SOAP 1.2
- Easier and more flexible for developers
  - Modularised
  - Adopts
    - XML Schemas
    - XML Information Set
- Clarification of IP ownership
- Getting a bit bogged down
  - Discussions on how to handle binary data, working *"to synthesize the different approaches"* of various issues *"as they pertain to the diagra"*
  - Developments are slowing

WS-I

- An open, industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages
  - IBM, Microsoft, IETF, OAGI, OASIS, OMG, UDDI, W3C, etc.
    - Sun conspicuously absent at start
      - (Probably) at Microsoft's behest
      - Admitted to board October 17, 2002
- Basic Profile Version 1.0
  - July 22, 2003
    - http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.htm
    - http://www.ws-i.org/docs/WhitePapers/200308zapthink-ws-i.pdf
  - Dictates how a selected set of specified Web services technologies should be used together in an interoperable manner:
    - Messaging—the exchange of protocol elements, usually over a network, to effect a Web service
    - Description—the enumeration of the messages associated with a Web service, along with implementation details
    - Discovery—metadata that enables the advertisement of a Web service's capabilities
    - Security—mechanisms that provide integrity, privacy, authentication and authorization functions
  - Incorporates SOAP 1.1, WSDL 1.1, UDDI 2.0, XML 1.0, and XML Schema

"WS-I's role is not to develop new standards and specifications. WS-I is really focused on profiling and establishing best practices for specs that are already widely adopted in the marketplace. WS-I is not, for example, getting into the whole orchestration area right now…"

WS-I WEB SERVICES INTEROPERABILITY ORGANIZATION

- *Lots* of "stuff" going on...
  - Hard to know what is useful, what is not...
    - Lots of activity, much of it quite 'juvenile'
      - *"A people with no past have no future"*
  - Microsoft/IBM seem to be taking the lead in generating 'standards'
    - Sun definitely behind the 8-ball
      - Preoccupied with Java
        - » A mistake?
        - » "it's the data, dummies!"

'Sun is getting left behind on Web services as it tries to come up with a non-Microsoft approach to the technology, according to one industry analyst.

"Sun has been fighting a losing rearguard action on this thing from day one. This is a battle that Sun has been battered and bloodied on"'

  - Microsoft is promoting its "Global XML Web Services Architecture (GXA) platform"
    - a framework that augments the basic Web service with generic higher-level services like security, reliability and transactions
    - Includes a lot of work
      - Looked at next...

WS-Security

- Adds "baseline security" to WebServices
  - defines a set of SOAP headers that can be used to implement security measures for Web services
  - specification describes how to add encryption and digital signatures to Web services
    - in each case supporting a W3C working group effort: XML-Encryption and XML-Signatures
  - also defines a general mechanism for passing around arbitrary security tokens

"...not everyone is using SOAP, simply because it's early in the deployment phase. But soon SOAP stacks will be as prevalent and standard as TCP/IP stacks are today.

And it looks like a WS-Security-based security stack may become just as ubiquitous."

- Offered by IBM, Microsoft, and VeriSign
  - supported by Sun
- Promoted by OASIS
  - seems to be taking over stewardship for WebServices from W3C

'Clearly, there's been a shift in power among standards bodies. Companies are becoming increasingly frustrated by the bureaucracy of the W3C, as well as its emphasis on esoteric concepts like "the semantic Web." What were once quiet whispers have now become on-the-record realities.'

WS-Security…

- Spec has six parts, in two layers:
  - Policy
    - **WS-Policy**, a general purpose spec describing how to express enterprise security policies
    - **WS-Trust**, defining brokered trust relationships a Web services environment
    - **WS-Privacy**, defining a way for Web services to state and implement privacy preferences and practices
  - Federation
    - **WS-Secure Conversation**, defining a general method for managing and authenticating message exchanges between parties
    - **WS-Federation**, describing how to manage and broker trust relationships in a heterogeneous federated environment, including support for federated identities
    - **WS-Authorization**, defining how Web services manage authorization data and policies

SAML 1.0

- Security Assertion Markup Language
  - OASIS standard; ratified 6th November, 2002
- An XML-based framework for Web services that allows the exchange of authentication and authorization information among business partners.
- Enables Web-based security interoperability functions, such as single sign-on, across sites hosted by multiple companies.
- Most vendors of Web access management solutions have committed to SAML 1.0 and are currently implementing the specification in their products
  - Netegrity have the first 'real' implementation
    - http://www.netegrity.com/products/index.cfm?leveltwo=JSAML

SAML 1.0

```
<saml:assertion Issuer="example.com"…>
    <saml:Conditions NotBefore=… NotAfter=…/>
    <saml:AuthenticationStatement
      AuthenticationMethod=… AuthenticationInstant=… >
      <saml:subject …>John Doe</saml:subject>
    </saml:AuthenticationStatement>
    <saml:AttributeStatement>
      <saml:subject …>John Doe</saml:subject>
      <saml:Attribute AttributeName="Title" …>
        <saml:AttributeValue>Manager</AttributeValue>
      </saml:Attribute>
     <saml:Attribute AttributeName="SpendLimit" …>
       <saml:AttributeValue>10000.00</AttributeValue>
     </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
```

John Doe is a manager with authority for purchases up to $10,000

- W3C has a working group looking at this
- MS WebServices Enhancements: WSE
  - Coming up to version 2.0
- IBM's XML Security Suite for Java: XSS4J
  - Still a bit rough

```
<PurchaseOrderRequest>
  <Order>
    <Item>
      <Code>42</Code>
        <Description>Some good stuff</Description>
    </Item>
    <Quantity>1</Quantity>
  </Order>
  <Payment>
    <CreditCard>
      <BrandId>Brand X</BrandId>
      <Number>123</Number>
      <ExpiryDate>20030408</ExpiryDate>
    </CreditCard>
    <PurchaseAmount>
      <Amount>12.36</Amount>
      <Currency>A$</Currency>
    </PurchaseAmount>
  </Payment>
</PurchaseOrderRequest>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<PurchaseOrderRequest>
  <Order>
    <Item>
      <EncryptedData
       xmlns="http://www.w3.org/2001/04/xmlenc#
        <CipherData>
          <CipherValue>BNjiv…dnZBrg=</CipherValue>
        </CipherData>
      </EncryptedData>
    </Item>
    <Quantity>1</Quantity>
  </Order>
  <Payment>
    <EncryptedData
     xmlns="http://www.w3.org/2001/04/xmlenc#">
      <CipherData>
        <CipherValue>ci3q/7Bq…TxA==</CipherValue>
      </CipherData>
    </EncryptedData>
    <PurchaseAmount>
      <Amount>12.36</Amount>
      <Currency>A$</Currency>
    </PurchaseAmount>
  </Payment>
</PurchaseOrderRequest>
```

XML Encryption...

```
XMLPlainText xmlPlainText = new XMLPlainText(new FileInputStream("payment.xml"));
KeyStore ks = KeyStore.getInstance("jceks");
ks.load(new FileInputStream("jwstore"), "jwpassword".toCharArray());

KeyStoreKeyInfoResolver kskiResolver = new KeyStoreKeyInfoResolver(ks);
kskiResolver.putAliasAndPassword("merchant", "merchantpassword".toCharArray());
kskiResolver.putAliasAndPassword("bank", "bankpassword".toCharArray());

EncryptedData ed = new EncryptedData();
ed.setCipherData(prepareCipherData());

EncryptionMethod rsa_1_5 = prepareEncryptionMethod(EncryptionMethod.RSA_1_5);

KeyInfo merchant = prepareNameOnlyKeyInfo("merchant");
KeyInfo bank = prepareNameOnlyKeyInfo("bank");

xmlPlainText.encrypt("//Item", kskiResolver, ed, EncryptedData.CONTENT, rsa_1_5, merchant);
xmlPlainText.encrypt("//CreditCard", kskiResolver, ed, EncryptedData.ELEMENT, rsa_1_5, bank);

PrintWriter pw = new PrintWriter(new FileWriter("encryptedPayment_rsa_1.xml"));
pw.write(xmlPlainText.toString());
pw.close();
```

```
<PurchaseOrderRequest>
  <Order>
    <Item>
      <Code>42</Code>
      <Description>Some good stuff</Description>
    </Item>
    <Quantity>1</Quantity>
  </Order>
  <Payment>
    <CreditCard>
      <BrandId>Brand X</BrandId>
      <Number>123</Number>
      <ExpiryDate>20030408</ExpiryDate>
    </CreditCard>
    <PurchaseAmount>
      <Amount>12.36</Amount>
      <Currency>A$</Currency>
    </PurchaseAmount>
  </Payment>
</PurchaseOrderRequest>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<PurchaseOrderRequest>
  <Order>
    <Item>
      <EncryptedData
       xmlns="http://www.w3.org/2001/04/xmlenc#
        <CipherData>
          <CipherValue>BNjiv…dnZBrg=</CipherValue>
        </CipherData>
      </EncryptedData>
    </Item>
    <Quantity>1</Quantity>
  </Order>
  <Payment>
    <EncryptedData
     xmlns="http://www.w3.org/2001/04/xmlenc#">
      <CipherData>
        <CipherValue>ci3q/7Bq…TxA==</CipherValue>
      </CipherData>
    </EncryptedData>
    <PurchaseAmount>
      <Amount>12.36</Amount>
      <Currency>A$</Currency>
    </PurchaseAmount>
  </Payment>
</PurchaseOrderRequest>
```

BPEL4WS

- IBM, Microsoft and BEA originated; now V1.1
  - Supported by webMethods and others
  - We're not in Kansas anymore
    - adds much complexity
      - what happened to the "Simple" word?
  - *(Sun supports an alternative: WS-Choreography Interface, WSCI)*
- XML-based flow language that defines how business processes interact
  - Can describe complex processes that can span multiple companies, such as order processing, lead management and claims handling
  - Helps enable business processes to interoperate within and between companies that use different underlying technologies
  - Accounts for various issues, such as:
    - state
    - coordination
    - data-dependent behaviour
    - timing
    - exceptional behaviour
    - nested interactions
  - Serves as the basis for "automated protocol enforcement engines" that can track the state of protocol instances and detect protocol errors in message flows
- Works with WS-Coordination and WS-Transaction
  - The three coordinate the various activities that occur within a business process, in order and at the right time for completion

BPEL4WS...

- Sits “on top of” WSDL
  - WSDL services define the basic units of interaction
- Provides a simple declarative flow language
  - <receive>
  - <reply>
  - <invoke>
  - <assign>
  - <throw>
  - <terminate>
  - <wait>
  - <empty>
  - <sequence>
  - <switch>
  - <while>
  - <pick>
  - <flow>
  - <scope>
  - <compensate>

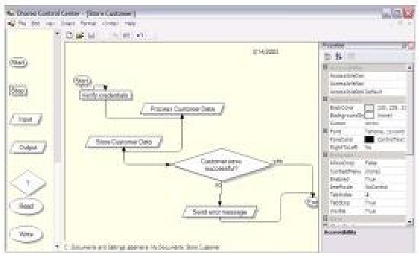
```
 <invoke partner="Seller" portType="SP:Purchasing"
         operation="SyncPurchase"
         inputContainer="sendPO"
         outputContainer="getResponse">
   <compensationHandler>
       <invoke partner="Seller" portType="SP:Purchasing"
               operation="CancelPurchase"
               inputContainer="getResponse"
               outputContainer="getConfirmation">
   </compensationHandler>
 </invoke>
```

```
<pick>
  <onMessage partner="buyer"
             portType="orderEntry"
             operation="inputLineItem"
             container="lineItem">
      <!-- activity to add line item to order -->
  </onMessage>
  <onMessage partner="buyer"
             portType="orderEntry"
             operation="orderComplete"
             container="completionDetail">
      <!-- activity to perform order completion -->
   </onMessage>

   <!-- set an alarm to go after 3 days and 10 hours -->
   <onAlarm for="P3DT10H">
      <!-- handle timeout for order completion  -->
   </onAlarm>
</pick>
```

```
<sequence>
  <wait until="2002-12-24T18:00+01:00"/>

  <invoke partner="CallServer" portType="AutomaticPhoneCall"
          operation="TextToSpeech"
          inputContainer="seasonalGreeting">
  </invoke>
</sequence>
```

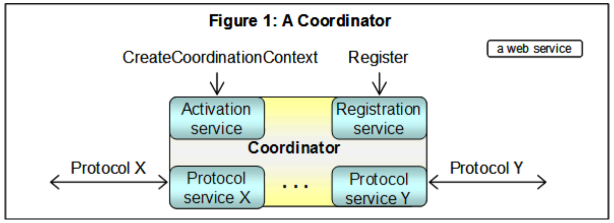
- Also possible to use BPEL4WS to *"define an executable business process… the language effectively defines a portable execution format for business processes that rely exclusively on Web Service resources and XML data"*
  - just an idea at the moment…but written into the specification: *"This is arguably the most attractive prospect for the use of BPEL4WS from the viewpoint of unlocking the potential of Web Services because it allows the development of tools and other technologies that greatly increase the level of automation and thereby lower the cost in establishing cross-enterprise automated business processes."*
- IBM's BPWS4J engine provides a toolkit for WebSphere and Tomcat
  - Plus a plug-in for the Eclipse IDE
- Third party support for .Net
  - Collaxa, http://www.collaxa.com
  - ChoreoServer, http://www.openstorm.com

ChoreoServer Control Center
(Coming Soon)



WS-Coordination

- *"enables an application service to create a context needed to propagate an activity to other services."*
- *'...is all about setting up contexts. WS-Coordination lets you create a "context" out of the blue that you can propagate/flow to anyone you want to talk to in "the context of what you are currently doing". That context can be used to hang any number of services on it, like, for instance, transactions.'*



- Defines an extensible framework for defining coordination types
  - specification provides for two coordination types
    - atomic transaction (AT)
      - used to coordinate “all or nothing” activities having a short duration and executed within limited trust domains
      - enables existing transaction processing systems to wrap their proprietary protocols and interoperate across different hardware and software vendors.
    - business activity (BA)
      - used to coordinate activities that are long in duration
        - » can’t use locking to make actions tentative and hidden from other applications. Instead, actions are applied immediately and are permanent.
      - compensations are used for error handling
  - A WebService application can include/use both AT and BA types

- *"WS-Coordination, WS-Transaction and BPEL4WS Describe How to Reliably Define, Create and Connect Multiple Business Processes in a Web Services Environment. ...BPEL4WS allows companies to describe business processes that include multiple Web services and standardize message exchange internally and between partners. WS-Coordination and WS-Transaction provide companies with a reliable and durable way of handling multiple Web services interactions, regardless of the underlying computing infrastructure. In addition, they outline how partners can interact with a collection of Web services and coordinate the outcome of those corresponding activities."*

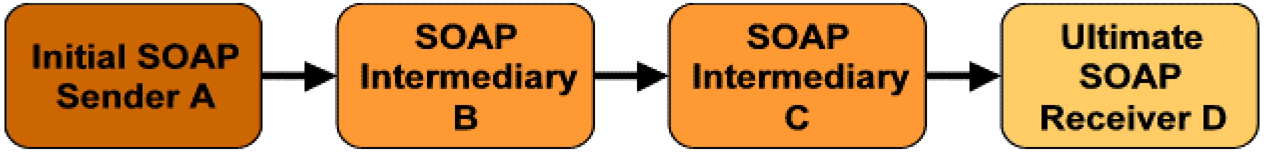
http://www.prdomain.com/companies/i/ibm/newreleases/200208aug/pr_ibm_nr_20020809.htm

WS-Routing

- A simple, stateless, SOAP-based protocol for routing SOAP messages in an asynchronous manner over a variety of transports like TCP, UDP, and HTTP
- Can describe the entire message path for a SOAP message (as well as its return path) within the SOAP envelope
- Supports one-way messaging, two-way messaging such as request/response and peer-to-peer conversations, and long running dialogs

WS-Routing...

```
<SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://www.w3.org/2001/06/soap-envelope">
   <SOAP-ENV:Header>
      <wsrp:path xmlns:wsrp="http://schemas.xmlsoap.org/rp/">
         <wsrp:action>http://www.im.org/chat</wsrp:action>
         <wsrp:to>soap://D.com/some/endpoint</wsrp:to>
         <wsrp:fwd>
            <wsrp:via>soap://B.com</wsrp:via>
            <wsrp:via>soap://C.com</wsrp:via>
         </wsrp:fwd>
         <wsrp:from>soap://A.com/some/endpoint</wsrp:from>
         <wsrp:id>uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</wsrp:id>
      </wsrp:path>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      ...
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



- Enables the routing strategies used by SOAP nodes in a message path to be dynamically configured. SOAP itself provides a distributed processing model where SOAP messages can have content destined for specific processing nodes. WS-Routing adds to SOAP the capability of describing the actual message path
- WS-Referral provides a way to configure how SOAP routers will build a message path, whereas WS-Routing provides a mechanism for describing an actual path of a message

WS-Referral...

```
<r:ref xmlns:r="http://schemas.xmlsoap.org/ws/2001/10/referral">
  <r:for>
    <r:exact>soap://example.org/some.doc</r:exact>
    <r:prefix>soap://example.org/topics/icebergs</r:prefix>
  </r:for>
  <r:if>
    <r:ttl>43200000</r:ttl>
  </r:if>
  <r:go>
    <r:via>soap://example.com/mirror</r:via>
  </r:go>
  <r:refId>uuid:09233523-345b-4351-b623-5dsf35sgs5d6</r:refId>
  <r:desc>

<r:refAddr>http://example.com/references/2001/10/1234.xml</r:refAddr>
  </r:desc>
</r:ref>
```

for any SOAP actor name **matching** the set of SOAP actors listed in the **for** element

if the set of **conditions** listed in the **if** element is met and hence the statement is **satisfied**

then go via one of the SOAP routers listed in the **go** element

Go to http://example.com/references/2001/10/1234.xml, which may give you the object with uuid:09233523-345b-4351-b623-5dsf35sgs5d6. Then:

For any SOAP actor name matching the SOAP actor "soap://example.org/some.doc" or SOAP actors starting with "soap://example.org/topics/icebergs", if this referral is less than 12 hours (43,200,000 milliseconds) old then go via "soap://example.com/mirror".

WS-Inspection

- An XML format for assisting in the inspection of a site for available services
  - "closely resembles business cards"
- Provides a completely distributed model for providing service-related information
  - Contrast with UDDI:
    - service descriptions may be stored at any location
    - requests to retrieve the information are generally made directly to the entities which are offering the services
    - Searching/discovery harder than for UDDI
  - An admission of failure for UDDI?

WS-Inspection….

```
<?xml version="1.0"?>
<inspection xmlns="http://schemas.xmlsoap.org/wsil/"
            xmlns:wsiluddi="http://schemas.xmlsoap.org/wsil/uddi/">
  <service>
    <abstract>A stock quote service with two descriptions</abstract>
    <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
                 location="http://example.com/stockquote.wsdl"/>
    <description referencedNamespace="urn:uddi-org:api">
       <wsiluddi:serviceDescription
                 location="http://www.example.com/uddi/inquiryapi">
         <wsiluddi:serviceKey>
                 4FA28580-5C39-11D5-9FCF-BB3200333F79
         </wsiluddi:serviceKey>
       </wsiluddi:serviceDescription>
    </description>
  </service>
  <service>
    <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
            location="ftp://anotherexample.com/tools/calculator.wsdl"/>
  </service>
  <link referencedNamespace="http://schemas.xmlsoap.org/wsil/"
        location="http://example.com/moreservices.wsil"/>
</inspection>
```

- Defines a generic model and syntax for describing and communicating the policies of a Web service
  - *"...to successfully integrate with a nontrivial Web service, one must fully understand the service's XML contract along with any additional requirements, capabilities, and preferences (also referred to as policies). For example, just knowing that a service supports WS-Security is not enough information to enable successful integration. The client needs to know if the service actually requires WS-Security. If so, it also needs to know what security tokens it's capable of processing (such as UsernameToken, Kerberos tickets, or certificates), and which one it prefers..."*

Microsoft Tools

- http://msdn.microsoft.com/webservices/
- Web Services Enhancements 2.0
  - http://msdn.microsoft.com/library/en-us/dnwebsrv/html/programwse2.asp
- SOAP Toolkit
- Office XP Web Services Toolkit
- SQLXML
  - Web Services support for MSSQL Server

JSR 109

- WebServices for J2EE
  - Went final on 15 Nov, 2002
  - Covers a number of important areas:
    - How J2EE applications can access Web services as if they were "traditional" remote objects
    - How Web services can be implemented as servlets and stateless session EJBs and their lifecycle
    - Deployment into a compliant application server
  - Largely an attempt to standardise what is already happening in the J2EE world
    - BEA, Oracle, etc. support WebServices at the "flick of a switch"

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
- Sun Dot-Com Builder
  - http://dcb.sun.com/practices/webservices
- O'Reilly
  - http://www.onjava.com/onjava/web_services