DB2 & Web Services

Principal sources:
Dirk Wollscheid, IBM Silicon Valley Lab.
Arthur Ryman, IBM Toronto Lab.

Weak Integration

Corporate Travel Server

HTTP/HTML

Copy/Paste

Corporate Travel GUI

2

Airline Flight Web Server

HTTP/HTML

Airline Flight GUI
Improved Integration

"The objectives of a business organization are only partially known at any given time, and are always subject to change."

Source: 'Enterprise Application Integration & Information Flow', by John Mann, The Yankee Group

This constantly changing and often unpredictable environment creates a major challenge for those seeking to justify long-term investment, particularly in I.T...

...A more realistic approach is to build flexibility into I.T. systems to ensure that they can be configured, augmented or replaced when change inevitably occurs.

Source: 'Information Integration', by Spike Cavell & Co.
Distributed Architectures

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Tightly Coupled</th>
<th>Loosely Coupled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Development</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Response Time</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>Re-use</td>
<td>software components</td>
<td>data</td>
</tr>
<tr>
<td>Business to Business</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Cross-Platform</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Availability</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Scale-ability</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Service-Oriented Architecture

Service

Transport

Description
Conceptually, Web Services represent a model in which discrete tasks within e-business processes are distributed widely throughout a value net.

Specifically, Web Services are a stack of emerging standards that describe a service-oriented, component-based application architecture. 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.

Gartner define Web Services as software components that employ one or more of SOAP, WSDL, or UDDI to perform distributed computing. Using these technologies constitutes a Web service; using all of them is not required.

Web Services Protocol Stack

- Base Technologies:
  - Processes: Discovery, Aggregation, Choreography, ...
  - Descriptions: Web Services Descriptions (WSDL)
  - Messages: SOAP Extensions (Correlation, Reliability, Transactions, ...), SOAP
  - Schema
  - Communications: HTTP, SMTP, FTP, JMS, IIOP, ...

http://www.w3.org/TR/ws-arch/#whatis

www.w3.org/TR/ws-arch/
XML

- Extensible Markup Language
- Self-defining data structure protocol
- Text-only -- ideal for cross-platform data

For example:

```xml
<?xml version='1.0'>
<CustomerBalance>
  <CustomerNumber>123456</CustomerNumber>
  <AccountBalance>$5,000</AccountBalance>
</CustomerBalance>
```

- **DTD:** defines the data model for an XML document
- **Schema:** similar to DTD, with some advantages, e.g.
  - data type of each element and attribute
  - has a name, aka 'schema name space'

[www.w3.org/XML](http://www.w3.org/XML)

SOAP Specification

- (originally Simple Object Access Protocol, now only a name)

- **Message Format:**
  - defines what is in a message

- **Message Elements:**
  - how to parse the message

- **Encoding rules:**
  - serialization mechanisms for encoding application-defined data types

- **Representation:**
  - how to represent remote procedure calls and responses

[www.w3.org/TR/SOAP](http://www.w3.org/TR/SOAP)
SOAP Message Types

- **Request**
  - Contains invoked service, service operations and input parameters

- **Response**
  - Returns output parameters returned from a service invocation

- **Fault**
  - Used to communicate errors

SOAP Message Structure

- **Envelope**
  - Root element of SOAP message

- **Header**
  - Used to include additional information
  - Must be immediate child of envelope
  - Must occur before the body element

- **Body**
  - Contains the method call, and the call arguments

- **Fault**
  - Status information contained within the SOAP body
    - `<faultcode>`: Specify error codes
    - `<faultstring>`: Textual description of fault
    - `<faultactor>`: The entity that generated the fault
    - `<detail>`: Application specific data about the fault
Example message

```xml
<SOAP:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <SOAP:Header>
    <MyHeader xmlns="http://www.acme.com">
      <Username>MyUsername</Username>
    </MyHeader>
  </SOAP:Header>

  <SOAP:Body>
    <ns1:DoSomething xmlns:ns1="urn:MyService">
      <in0 xsi:type="xsd:int">300</in0>
      <in1 xsi:type="xsd:boolean">false</in1>
    </ns1:DoSomething>
  </SOAP:Body>

</SOAP:Envelope>
```

Namespaces

```xml
<SOAP:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <SOAP:Header>
    <MyHeader xmlns="http://www.acme.com">
      <Username>MyUsername</Username>
    </MyHeader>
  </SOAP:Header>

  <SOAP:Body>
    <ns1:DoSomething xmlns:ns1="urn:MyService">
      <in0 xsi:type="xsd:int">300</in0>
      <in1 xsi:type="xsd:boolean">false</in1>
    </ns1:DoSomething>
  </SOAP:Body>

</SOAP:Envelope>
```
Encoding

```xml
<SOAP:Envelope
   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
   SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

   <SOAP:Header>
     <MyHeader xmlns="http://www.acme.com">
       <Username>MyUsername</Username>
     </MyHeader>
   </SOAP:Header>

   <SOAP:Body>
     <ns1:DoSomething xmlns:ns1="urn:MyService">
       <in0 xsi:type="xsd:int">300</in0>
       <in1 xsi:type="xsd:boolean">false</in1>
     </ns1:DoSomething>
   </SOAP:Body>

</SOAP:Envelope>
```

WSDL

- **Web Services Description Language**
  - An XML language for describing Web services

- Describes:
  - What the Web service does
  - How to invoke it
  - Where it is located

...an IDL for Web services

[www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)
WSDL elements

- **Service**
- **Port**
  - **Endpoint (address for binding)**
- **Binding**
  - **Protocol & data encoding**
- **Port Type**
  - **Set of Operations**
- **Operation**
  - **Method**
- **Message**
  - **Input, Output, Fault**
- **Parameters**
  - **Part**
    - **Type**
      - **Data Type**

UDDI

- **Universal Description, Discovery, and Integration**

  - **Registry for:**
    - Service types
    - Businesses Information
      - Service details

  - **Description:**
    - Host common definitions for service interfaces (promote standardization and interoperability)

  - **Discovery:**
    - Find service implementations, and the businesses that provide them

  - **Integration:**
    - Facilitate aggregation of services in business workflows (promotes B2B integration)

  - [www.uddi.org](http://www.uddi.org)
Example Applications

- **Private**
  - ‘open’ alternative to existing interfaces, such as:
    - client/server
    - RPC
    - ...

- **Business to Business**
  - ‘open’ versions of proprietary interfaces
    - for example, alternatives to SAP’s IDOC, PeopleSoft’s and J. D. Edwards APIs...

- **Public**
  - hundreds of examples already available...

---

Real Web Services

- Query web documents directly from other computer applications:
  google.com/apis

- Locate entertainment, restaurants, and lodging in a given area:
  zagat.com

- Find ATM locations in a given zip code:
  serviceobjects.com/products/dots_atm_demo.asp

- Track the current location and status of a FedEx package:
  fedex.com/us/tracking

- Translate text between two languages:
  babelfish.altavista.com
DB2 II Web Services Overview

Types of DB2 Web Services operations

- Any DB2 Stored Procedures
  - output parameters and result sets are returned
  - results are tagged using default XML tagging

- SQL requests
  - results are tagged using default XML tagging
  - Parameters to SQL statement

- XML Extender DAD for:
  - DAD (Document Access Definition) describes XML to table mapping
  - Advanced tagging of XML results from SQL
  - XML shredding into relational data
  - Retrieving of XML data constructed from relational data
Development Scenario

- Users list database operations to be exposed as Web Services
  - <DADX>
    - <operation name="listMeetings">
      - <documentation>
        List meetings on calendar on a certain date.
      - </documentation>
      - <query>
        - <SQL_query>
          SELECT * FROM CALENDAR WHERE DATE = :date
        - </SQL_query>
        - <parameter name="date" type="xsd:date"/>
      - </query>
    - </operation>
  - </DADX>

- Provider Runtime helps:
  - Deploy Web Service
  - Generate WSDL
  - Provide HTML test environment to run Web Service
  - Automatic regeneration if Web Service definition changes

- Platforms:
  - WebSphere (distributed, zSeries, and iSeries)
# Building Web Services

<table>
<thead>
<tr>
<th>Provider</th>
<th>Tools</th>
<th>SOAP</th>
<th>WSDL</th>
<th>UDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>WebSphere Studio ETTK (was WSTK)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BEA</td>
<td>WebLogic Workshop</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Oracle</td>
<td>Portal Development Kit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Visual Studio .net .net SDK</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sun</td>
<td>Forte for Java WSDP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HP</td>
<td>e-Speak Tools</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

# WebSphere Studio

- Tooling enables end-to-end development of DB-oriented Web Services
  - Create SQL queries with SQL editor or XML Extender DADs
  - Create Web Service definition files (DADX)
  - Deploying of Web Services to WebSphere and Apache Tomcat
  - Publishing of WSDL in UDDI
  - Testing of Web Service
  - Generation of Java Web Service clients
  - Generation of Web Service UDFs
Example: DB Schema

Example: Query
Example: DADX group

Set DADX Group Properties

Example: DADX file

Generate DADX file
Example: DADX content...

```xml
<?xml version="1.0" encoding="UTF-8"?>
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://schemas.ibm.com/db2/dxs/dadx dadx.xsd"
    xmlns:/documentation="http://www.w3.org/1999/xhtml">

    My calendar web service

    </dadx:documentation>
    <dadx:operation name="selectAllMeetings">
        <dadx:documentation xmlns="http://www.w3.org/1999/xhtml">
            Show all meetings
        </dadx:documentation>
        <dadx:query>
            <dadx:SQL_query>
                <![CDATA[
                    SELECT * FROM DEMO.CALENDAR
                ]]> 
            </dadx:SQL_query>
        </dadx:query>
    </dadx:operation>
</dadx:DADX>
```

Example: Deployment
Test 'selectAllMeetings' operation

...DADX content
Test 'createMeeting' operation

```xml
<n:createMeetingResponse xmlns:n='http://tempuri.org/calendar/calendar.dada'
xmlns:s:XMLSchema='http://www.w3.org/2001/XMLSchema'
<returns>1</returns>
</n:createMeetingResponse>
```

Security

- **WebSphere security can be used for:**
  - Encryption of messages being exchanged
  - **Authentication/authorization for Web Service endpoints**
    - Only authenticated users can access Web Services
    - Specify which user can access which Web Service

- **Database user information:**
  - Can be specified in WebSphere or in the Web application itself
  - One database user per group of DADX web services

- **No support for WS-Security — yet!**
Authentication & Authorization

Roles & Constraints
Futures

- Dynamic Query Web Service
  - Allows execution of SQL statements without DADXes

- Support for more SOAP engines

- WS-Security
  - Message level security

- N:M Mapping of WebSphere/Web Service users to database users
  - Leverages permissions in Database

Integration Advantages with Web Services

- Flexibility
  - Independence of programming languages, operating systems or hardware platforms used on client or server

- Standards
  - Stack of Web Service specifications
    - Connection layer: SOAP, WSDL, UDDI
    - Security and reliability layer: WS-Security, WS-ReliableMessaging, ...
    - Enterprise: WS-Transaction, Workflow, Systems Management, ...
  - Packaged applications with Web service interfaces
    - WebSphere Business Integrator, SAP, Siebel, MS Office, ...

- Cost saving
  - Tooling for code generation of clients
  - Development cost of provider is low because of tooling
  - Enable existing technology for Web Services
  - Replaces manual or paper-based processes
DB2 Web Services

- Ideal for integration of data

- Provider technology
  - allows easy exposing of DB2 data or logic as Web Service

- Consumer technology
  - allows integration of dynamic data

- Development cost is low
  - Quick reaction to new requirements
  - Tooling eliminates need for coding and Web Service skills

Lightyear Consulting Ltd.
Palo Alto - Austin - Scottsdale - Laguna Beach - Calgary
www.lightyr.com

Next steps towards Web Services...

1. Tutorials:
     - Access DB2 with Web services
     - Building Web services with WORF and DB2 XML Extender
     - Invoking Web Services from SQL

2. Tools & code examples:

3. Ask Lightyear:
   - On-site technical seminars & education classes
   - Consulting services
   - Java Development
   - ...