SOA
Service Oriented Architecture

Dr. F. Shams
A. Mahjoorian

CSICC2007 Workshop
Table of Content

- What is SOA?
- SOA Items
- SOA Implementation
- Conclusion
What is SOA?

- Overview
- SOA Definition
- SOA Benefits
- EA Future
- What is Service?
- What is Web Service?
- Effect of SOA on current systems
What is Service?

- A unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software components.

A service in SOA is an exposed piece of functionality with three major properties:

- The interface contract to the service is platform-independent.
- The service can be dynamically located and invoked.
- The service is self-contained. That is, the service maintains its own state.
What is a Web service?

- any Service that is callable by another program across the **Web** in a way that is:
  - platform-independent,
  - language-independent &
  - object model-independent

- So **Web Service** = web-base Service

- Uses “next generation” infrastructure standards:
  - XML, SOAP, WSDL, UDDI
Next-generation architecture strategies — the road to SOA
SOA is …

- Service-oriented architecture (SOA) is an approach to loosely coupled, protocol independent, standards-based distributed computing where software resources available on the network are considered as Services.

- SOA is believed to become the future enterprise technology solution that promises the agility and flexibility the business users have been looking for by leveraging the integration process through composition of the services spanning multiple enterprises.

- Communication infrastructure used within an SOA should be designed to be independent of the underlying protocol layer.
SOA Layers
What is SOA?

- SOA should be **business oriented**
- SOA is a **way of thinking**
- SOA is **not** Web Services
- **Loosely coupled** architecture that uses **messaging**
- **Enriched** by creating composite Apps
- Move from batch to **real-time**
- Build reusable **business components** (Services)
- Provide **agility and flexibility**
Difference between “Service”, “Web Service” and “Web Application”

**Component** = Reusable, Portable, Well-Defined, reliable, extendable Modules

**Service** = Business-Driven Platform-Independent **Component**

**Web Service** = Web-Base **Service**

*Web Application* = application used by a **user**

*Web Service* = application used by another **application**
As-Is Application: Integration Problem!

Enterprise SOA

- ERP
- Database
- Mainframe
- Financials
- CRM
- UNIX®
- Java
- .Net

Business process changes

Every change requires a change in the integration technology
- Slows down business
- Increases operational cost

Application implementation changes
IT Strategy in 2004

Top strategic software platform project over the next year

% of respondents

- Application integration: 35%
- e-business: 33%
- CRM: 30%
- SCM/Logistics: 24%
- HR: 23%
- Database upgrade: 21%
- Intranet improvements: 19%
- Financial (Accounting): 16%
- Marketing apps on Web site: 15%
- Commerce server: 13%
- e-procurement Web site: 12%
- Sys. mgmt infrastructure: 12%
- Building Internet company: 8%
- Engineering software: 7%
- Manufacturing software: 5%
- Other: 4%
- Deregulation: 3%
Companies Are Moving Forward Now With SOA

Q: What Stage Is Your Company Currently In With Respect to SOA?

Enterprise-wide SOA up 200%
Department-wide SOA up 300%,

- **Enterprise-wide SOA**
  - 2005: 8%
  - 2006: 16%

- **Department-wide SOA**
  - 2005: 4%
  - 2006: 12%

- **Pilot Projects**
  - 2005: 13%
  - 2006: 28%

- **Evaluation**
  - 2005: 20%
  - 2006: 25%

- **Not Planning to Deploy**
  - 2005: 7%
  - 2006: 21%

- **Don't Know**
  - 2005: 12%
  - 2006: 32%
SOA Benefit (IBM)

Business Benefits
- Business flexibility provided by increased granularity of processes
- Ability to quickly create business processes and composite applications to respond to changes in the marketplace
- Improved customer service, without having to worry about the underlying IT infrastructure

IT Benefits
- Becoming a more responsive IT organization
- Decreasing development and deployment cycle times through the use of pre built, reusable services building blocks
- Reducing complexity and maintenance costs with common services
- Enhancing existing IT systems rather than replacing them
SOA is well-suited to reuse current applications

Application -> Service
SOA is well-suited to built new applications
Service -> Application

Use / reuse

Create new applications
SOA Change work ...

Turn this...

...into this
SOA + Enterprise Service Bus (ESB)

Turn this…

…into this
SOA, Legacy, and the ESB

Enterprise Service Bus

Add new services faster

Change services with minimal impact to existing services

New Check Traveler Service

Travel Reservation Process

Book Flight Service

Check Credit Service

Hotel Availability Service

Flight Availability Service

Book Hotel Service

Book Car Service

New Flight Availability Service

Old Flight Availability Service

Travel Reservation Process
Why shift to Service Orientation

From
- Connections = cost
- Function oriented
- Build to last
- Prolonged development
- Application silos
- Vendor silos
- Tightly coupled
- Object oriented

To
- Connections = value
- Process oriented
- Build for change
- Incrementally deployed
- Orchestrated solutions
- Multi-vendor
- Loosely coupled
- Message oriented
SOA Items

- XML
- WSDL
- BPEL
- SOAP
- UDDI
SOA Server, Users and UDDI

Enterprise SOA
What is XML?

- Standard data types and structures, independent of any programming language, development environment, or software system.

```xml
<book>
  <authors>
    <author>
      <title>Joliet</title>
      <first>Jake</first>
      <last>Blues</last>
    </author>
  </authors>
  <title>Shades of Blue</title>
  <publisher>Free Forest Publishing</publisher>
</book>
```
What is WSDL?

A WSDL file, in XML structure, contains details about provider (server) services.

For each service the provider supports, WSDL describes the set of operations.

For each operation, WSDL describes formats of messages exchanged between the requesting consumer (client) and the provider (server).
What Is BPEL?

- **Business Process Execution Language**, platform independent, XML-based
- A language to specify the behavior of business processes *between* Web services *and as* Web services
- Contains process flow constructs for conditional branching, parallel processes, nested sub-processes, process joins, etc.
- Uses WSDL to describe process interfaces so that they can be more easily integrated into other processes or applications.
- Provided as an open standard under royalty free terms.
- Platform independence – XML based language (Java, .Net implementations available)
Price Procurement Flow

start

Accounting Dept.

credit check

invoke

Handle negative credit exception

Amazon

invoke

product data request

invoke

product data request

eBay

receive

product info

receive

product info

?

Select lowest price

end
BPEL Structure Overview

<process>
  <partners> … </partners>
  <variables> … </variables>
  <correlationSets> … </correlationSets>
  <faultHandler> … </faultHandler>
  <compensationHandler> … </compensationHandler>
  <eventHandler> … </eventHandler>
  (activities)*
</process>

activities = <receive>, <reply>, <invoke>, <assign>, <throw>,
             <terminate>, <wait>, <empty>, <sequence>, <switch>,
             <while>, <pick>, <flow>, <scope>, <compensation>
What is SOAP?

- Shortcut for Simple Object Access Protocol (SOAP)
- SOAP is an XML notation for describing how messages are assembled and transmitted over HTTP between service consumers (clients) and service providers (servers)
- SOAP = XML + HTTP
What is UDDI?

UDDI is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet.

- UDDI stands for Universal Description, Discovery and Integration
- UDDI is a directory for storing information about web services
- UDDI is a directory of web service interfaces described by WSDL
- UDDI communicates via SOAP
UDDI specification help to:

- Making it possible to **discover the right business** from the millions currently online
- Defining how to enable commerce once the preferred business is discovered
- Reaching new customers and increasing access to current customers
- Expanding offerings and extending market reach
- Solving customer-driven need to remove barriers to allow for rapid participation in the global Internet economy
- Describing services and business processes programmatically in a single, open, and secure environment
Flexibility vs. Interoperability

- Flat file + FTP
- Web services
- CORBA
SOA Implementation

- Sample of Web Service
- BPEL Implementation
- SOAP
Example: Using UDDI with SOAP and WSDL

User

UDDI Request

UDDI Directory

WSDL

SOAP Request

SOAP Response

SAP
ASP.NET Web Service example

<%@ WebService Language="VB" Class="TempConvert" %>
Imports System
Imports System.Web.Services

Public Class TempConvert :Inherits WebService
  <WebMethod()> Public Function FahrenheitToCelsius (ByVal Fahrenheit As Int16) As Int16
    Dim celsius As Int16
    celsius = ((((Fahrenheit) - 32) / 9) * 5)
    Return celsius
  End Function

  <WebMethod()> Public Function CelsiusToFahrenheit (ByVal Celsius As Int16) As Int16
    Dim fahrenheit As Int16
    fahrenheit = (((Celsius) * 9) / 5) + 32
    Return fahrenheit
  End Function
End Class
ASP.NET Web Service example

- This Code is a .asmx file. This is the ASP.NET file extension for XML Web Services.

- The only difference from a normal application is that this function is defined as a "WebMethod".

- Use "WebMethod" to mark the functions in your application that you would like to make into web services.

- The ASP.NET has automatically created a WSDL and SOAP request.
Web Browser vs. Web Service

Web Browser

HTTP

Service Requester

SOAP

ASP.NET Webpage

Internal Service Call

.asmx file type
(ASP.NET Service)

Service Provider

Web Server
Find a Web Service example

Welcome to XMethods.
Emerging standards such as SOAP, WSDL and UDDI will enable a new generation of "web services" that allow systems to communicate with other systems over the open protocols of the Internet. For example, a corporate inventory management system might publish a web service that lets a customer's system check real-time inventory levels. This site lists publicly available web services.

Updates
- 2002-04-03 Introducing XSpace, a utility service [Read]
- 2002-03-04 WSDL Analyzer can now browse any WSDL L
- 2002-02-13 New feature: RPC Profiler [Read]
- 2002-02-06 New programmatic interfaces to XMethods [R
- 2002-01-14 Sign up to be notified of new services. [Read]

NEW! XMethods Utility Services
XMethods introduces its "utility services" - simple, useful, highly-interoperable building-block services that can be used as the foundation for business services applications. XSpace is our first service in this category, click on the link below to learn more about it.

XSpace Multi-user shared database "space" that stores key/value pairs

Service List

<table>
<thead>
<tr>
<th>Publisher</th>
<th>Style</th>
<th>Service Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SalCentral</td>
<td>DOC</td>
<td>UDDI Weather Report Web Service</td>
<td>Reports communication accuracy of UDDI links</td>
</tr>
<tr>
<td>SalCentral</td>
<td>DOC</td>
<td>Communications and Email Web Service</td>
<td>Specialist email facility, allows bulk email and newsletter sending.</td>
</tr>
<tr>
<td>ManojSrivastava</td>
<td>RPC</td>
<td>SecureXML</td>
<td>W3C Compliant XML Signature Verification Service</td>
</tr>
</tbody>
</table>

Also accessible via XML Interfaces: DISCO, WS-Inspection, RSS

See the interfaces section for more
WSDL: Analyze WSDL | View RPC Profile | http://www.alethea.net/webservices/ZipCode.asmx?WSDL

| XMethods ID   | 7534 |
| Service Owner: | jcono |
| Contact Email: | jcon0@ripeedev.com |
| Service Home Page: | http://www.alethea.net/webservices/ZipCode.asmx |
| Description: | Info about a location from zip code, area code, city, or state |
| SOAP Implementation: | MS .NET |

Contributed Clients for this Service  What is this?  Add / Edit / Delete Client

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Publisher</th>
<th>Toolkit</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB.Net Client</td>
<td>Application</td>
<td>jcon0</td>
<td>MS .NET</td>
<td>Visual Basic</td>
</tr>
</tbody>
</table>

Detailed Description

Retrieves pairs of city+state, zip code, area code, or time zone for a given zip code, city, or city+state.
SOAP Request Format

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /webservices/ZipCode.asmx HTTP/1.1
Host: www.alethea.net
Content-Type: text/xml; charset=utf-8
Content-Length: **length**
SOAPAction: "http://www.alethea.net/webservices/ZipCodeToCityState"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
>
  <soap:Body>
    <ZipCodeToCityState xmlns="http://www.alethea.net/webservices">
      <ZipCode>**string**</ZipCode>
    </ZipCodeToCityState>
  </soap:Body>
</soap:Envelope>
```
SOA Response Format

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfString xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.alethea.net/webservices">
    <string>Herndon, VA</string>
</ArrayOfString>
```
Elements of a BPEL Process

- **PartnerLinks**: placeholders for process callers and service providers.
- **FaultHandlers**: enclose activities that are performed in cases of error.
- **Activities**: subtasks of the process.
- **Variables**: hold data used in the business process.
- **CorrelationSets**: support process instance identification.
- **Control Links**: define the process' control flow.
BPEL Models

- `<sequence>`

- `<flow>`

- `<pick>`

Serialized `<send>`, `<receive>`, `<invoke>`, `<…>`

parallel `<send>`, `<receive>`, `<invoke>`, `<…>`

Wait for one or more `<receive>` or an alarm condition
BPEL Models

- `<switch>`
  `<case />`
  `</switch>`

- While
Executing BPEL

You’re approved!

Loan Service 1

Loan Service 2

Credit Validation Service

Loan Service
(Orchestration)

Loan Service (Orchestration)

Loan Service (Orchestration)
BPEL for Loan Service

```
<process>
  <sequence>
    ...
    <invoke>
    <receive>
    ....
    <while>
      <invoke>
      <receive>
    </While>
    ....
  </sequence>
</process>
```
**WSDL Sample**

```xml
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```
Web services and SOAP messaging

1. The Web service client creates and sends a SOAP message.
2. The SOAP server listens for SOAP messages. SOAP server is an application running in an Application Server.
3. The SOAP server processes the SOAP messages and passes the request to a Web service.
4. The Web service runs the specified method and returns the result to the SOAP server.
5. The SOAP server forwards the result back to the Web service client.
Conclusion

- SOA Concept …
- SOA Benefits
- SOA vs. others Issues
So SOA is …

- SOA and Web services are the tools that enable IT to accommodate business requests for flexibility while optimizing enterprise application investments.

- An Application Architecture that is designed to achieve loose coupling among interacting software applications. SOA provides greater flexibility in developing, integrating, and managing Enterprise Applications

- A set of components which can be invoked, and whose interface descriptions can be published and discovered (W3C)

- A methodology for achieving application interoperability and reuse of IT assets that features:
  
  - The concept of SOA isn’t new—what is new is the ability to mix and match execution environments, clearly separating the service interface from the execution technology
SOA is all that we need
SOA benefits

- **Reuse**—The ability to create services that are reusable in multiple applications.

- **Efficiency**—The ability to quickly and easily create new services and new applications using a combination of new and old services, along with the ability to focus on the data to be shared rather than the implementation underneath.

- **Loose technology coupling**—The ability to model services independently of their execution environment and create messages that can be sent to any service.

- **Division of responsibility**—The ability to more easily allow business people to concentrate on business issues, technical people to concentrate on technology issues, and for both groups to collaborate using the service contract.
Enterprise without SOA:
Not Integrated (Spaghetti-Like) Architecture
The original goals of CORBA are very similar to the goals of Web services. CORBA didn’t succeed widely because of vendor politics, and there’s some truth to that. However, CORBA also hurt itself in its early days by not defining a standard for interoperability.

**SOA isn’t new**

What is new is the ability to mix and match execution environments, clearly separating the service interface from the execution technology.
F_Shams@sbus.ac.ir
Mahjoorian@ieaf.ir