Service Oriented Architecture (SOA)

CSCI 5828 — Spring 2010
Foundations of Software Engineering

- Arpit Sud
Agenda

What is it?
Why to use it?
When to use it?
How to implement it?
Where not to apply it?
What?

Service oriented Architecture
What is (Software) Architecture?

- Software architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

  - IEEE 1471-2000
What is Service?

• A mechanism to enable access to one or more capabilities
  • using a prescribed interface
  • consistent with constraints and policies as specified by the service description.

• Windows service: RPC Locator, Event Log
• Software Service: Distribution, Alert, Log
So SOA is:

- A business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services. - IBM
- Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.
Components of SOA

Ref: Element of SOA

- Front End
- Service
  - Contract
  - Implementation
    - Business Logic
  - Service Repository
  - Interface
    - Data
- Service Bus
Architectural Principles

- **Loose coupling** - Services maintain a relationship that minimizes dependencies and only maintain an awareness of each other
- **Service contract** - Services adhere to a communications agreement as defined collectively by one or more service description documents
- **Service abstraction** - Beyond what is described in the service contract, services hide logic from the outside world
- **Service reusability** - Logic is divided into services with the intention of promoting reuse
Architectural Principles...

- **Service composability** - Collections of services can be coordinated and assembled to form composite services.
- **Service autonomy** – Services have control over the logic they encapsulate.
- **Service optimization** – All else equal, high-quality services are generally considered preferable to low-quality ones.
- **Service discoverability** - Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms.
Why?
Service oriented Architecture
Business Benefits

- Decreased cost:
  - Add value to core investments by leveraging existing assets
  - New systems can be built faster for less money because
  - Reducing integration expense
  - Built for flexibility
  - Long term value of interoperability
Business Benefits...

- Increased employee productivity:
  - Built on existing skills
  - Consolidate Duplicate Functionality

- Built for partnerships:
  - Standards based
  - Business relationships expressed via service interactions
  - Integration is driven by what is needed, not what is technically possible
Business Benefits...

• **Agility - Built for change**
  • Helps applications evolve over time and last
  • Abstract the backend and replace over time
  • Focusing on core-competencies
  • Incremental implementation approach is supported.
  • Service Outsourcing – new business model!
Technical Benefits

- **Services Scale**
  - Build scalable, evolvable systems
  - Scale down to mobile devices
  - Scale up to for large systems or across organizations

- **Manage complex systems**
  - Does not require centralized services
  - Empowers users with high end communication

- **Platform independent Use**

- **Loose Coupling allows flexibility**
WHEN?
Service Oriented Architecture
Drivers

- Large scale Enterprise systems
- Internet scale provisioning of services
- Reduce the cost of doing business
- Implementation abstraction
- Business process reuse (multiple use cases for same process)
How?
Service oriented Architecture
Getting to SOA

- Think services not objects
- Select which services to expose
- Choose a communication mechanism
- Consider the Security mechanism
- Consider the re-usability
Comprises of

- **Services**: a function or business processing that is well-defined, self-contained, and does not depend on the context or state of other services
  - Loan Processing Service, Weather Forecast Service
- **Connections**: The link connecting these self-contained distributed services with each other.
  - For web services: HTTP and SOAP.
Service Request - Response

Service Directory/Repository

Register The Service

Service Provider

Query For Service

Service Consumer

Invoke Service (via Proxy)
<%@ WebService Language="VBScript" Class="TempConvert" %>

Imports System
Imports System.Web.Services

Public Class TempConvert :Inherits WebService
    <WebMethod()>
    Public Function FahrenheitToCelsius (ByVal Fahrenheit As String) As String
        dim fahr
        fahr=trim(replace(Fahrenheit","","."))
        if fahr="" or IsNumeric(fahr)=false then return "Error"
        return ((((fahr) - 32) / 9) * 5)
    end function

    <WebMethod()> 
    Public Function CelsiusToFahrenheit (ByVal Celsius As String) As String
        dim cel
        cel=trim(replace(Celsius","","."))
        if cel="" or IsNumeric(cel)=false then return "Error"
        return (((cel) * 9) / 5) + 32
    end function

end class
Web Service Example Using ASP.NET

- **Test the Web service:**
  - *Celsius To Fahrenheit*
  - *Fahrenheit To Celsius*

- **Remember:**
  - SOA != Web service.
  - Web services can implement a service-oriented architecture.
Challenges

- Managing services
  - Insufficient attention to governance of services can cause performance and reliability issues
- Providing proper security for roles
- Assuring inter-operability of Services
Where NOT?
Service Oriented Architecture
Where *not* to use SOA?

- When you have homogenous IT environment
- When real time performance is critical
- When tight coupling is a pro not a con
- When things don’t change
Service
Question - Answer

Presenter
Question/Request
Answer/Response
Audience
References

- Elements of SOA, by Dirk Krafzig, Karl Banke, and Dirk Slama

- CIO Magazine - ABC: An Introduction to Service-oriented Architecture (SOA):
  [http://www.cio.com/article/40941/ABC_An_Introduction_to_Service_oriented_Architecture_SOA](http://www.cio.com/article/40941/ABC_An_Introduction_to_Service_oriented_Architecture_SOA)

- Microsoft – Service Oriented Architecture:

- Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools:

- Migrating to a Service-Oriented Architecture:

- [http://www.w3schools.com](http://www.w3schools.com)
Thank You!