Service-Oriented Architectures (SOA) and Semantic Web

Convenor: Hero Muhamad Sulaiman
MSc in Advanced Computer Science
Computer Science & IT Department
College of Science
Salahaddin University/Hawler
Email: hero.sulaiman@su.edu.krd
2023-2024

Syllabus

- Introduction, Motivation, and concepts of SOA
- XML
- DTD
- Introduction to UML
  - Use Case Diagrams
  - Sequence Diagrams
  - Component Diagrams
  - Class Diagrams
  - Object Diagrams

Syllabus

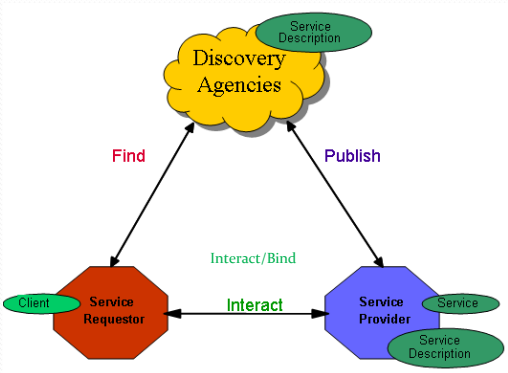
- Converting Class and Object diagrams to DTD
- XML based Messaging using SOAP
- Describing and Publishing Web Services using WSDL and UDDI
- Introduction to Web Services.
- Web Services and Semantic Web
- Introduction to Semantic Web
- Infrastructure of Semantic Web + Ontologies
- Ontology Engineering
- RDF & RDF Schema (RDFS)
- Turtle
- SPARQL

Syllabus

- Web Ontology Language : OWL
- Linked Data
- Light Weight Semantics (RDFa,Microdata/Microformat)

Service-Oriented Architecture

Lecture 1: Introduction to SOA



Outline

- Introduction
- Definitions of SOA
- Goals and Benefits of SOA
- SOA motivators
- Terminology

Introduction

"Things should be made as simple as possible, but no simpler." -- **Albert Einstein**

Loose Coupling or Loosely coupled: Services are connected to other services and clients using standard, dependency-reducing, decoupled and message-based methods such as XML document exchanges.

Introduction

- If you travel overseas on business, you need to bring power adapters along with you.
- **Real dependency**… (you need power)
- **Artificial dependency**… (your plug must fit into the local outlet)
- The aim is that, artificial dependencies *can not be removed*, however they *can be reduced* to the minimum.
- Reworking Einstein's principle: "Artificial dependencies should be reduced to the minimum but real dependencies should not be altered."
- So, Einstein was just talking about Loose Coupling…

SOA Definitions by different vendors

- "*A service-oriented architecture (SOA) is an application framework that takes everyday business applications and breaks them down into individual business functions and processes, called services. An SOA lets you build, deploy, and integrate these services independent of applications and the computing platforms on which they run.*" – **IBM**
- "*Service-Oriented Architecture is an approach to organizing information technology in which data, logic, and infrastructure resources are accessed by routing messages between network interfaces.*" – **Microsoft**
- *"An SOA is a set of components which can be invoked, and whose interface descriptions can be published and discovered."* – **W3C**.

So, What is SOA?

Essentially, SOA is a software architecture that starts with an interface definition and builds the entire application topology as a topology of interfaces, interface implementations and interface calls.

SOA Goals & Benefits

1) **Loose Coupling**: among interacting software agent.
2) **Quality of Service**: Security, Availability, Reliability, Scalability, Performance.
3) **Flexibility and Agility**: Services can be effectively designed and reconfigured in response to changing business requirements.
4) **Reusability**: SOA provide infrastructure that makes reuse possibilities in heterogeneous environment such as Java, .Net etc.

SOA Goals & Benefits

5. **Reduces IT Burden**: SOA allow IT to better support the organization by providing more value with less cost and less overall burden.
6. **Increased Return on Investment (ROI)**: Services are delivered as IT assets and provide repeated value with lower maintenance and integration cost.

SOA motivators

- Increasing nature of distributed systems.
- Heterogeneity of systems and computing environments
- Dynamics of operating environments
- Processes require multiple services
- Some other issues… (Next slide)

SOA motivators (Continued)

Problems:

Problem 1: Complexity

- Budget constraints
- Operating efficiencies

} Reuse systems, no replacing

- New business models
- Growth by merger and acquisition leads to integrating entire IT organizations, applications and infrastructures.

SOA motivators (Continued)

Problems:

Problem 2: Redundant and non-reusable programming

Mergers & acquisition

↓

Redundant apps & non-reusable functions of apps

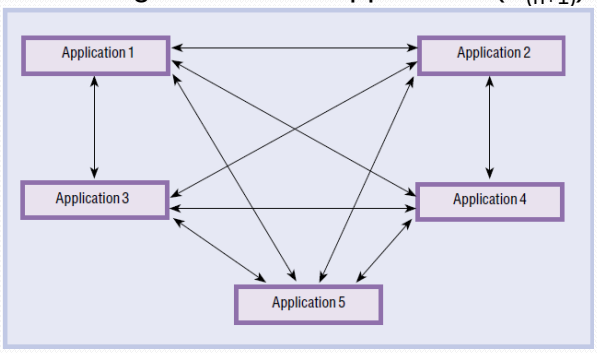
↓

Increasing both cost (using more resources) and time

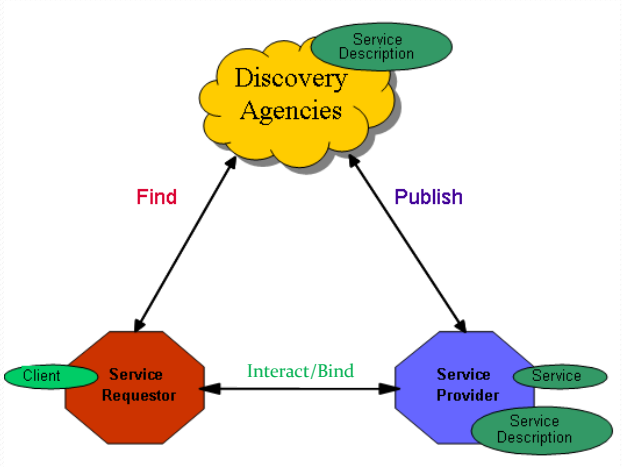
SOA motivators (Continued)

Problem 3: Multiple interfaces

- n(n-1) integration problem (n= No. off Appl.)
- Integrate another application ($A_{(n+1)}$)



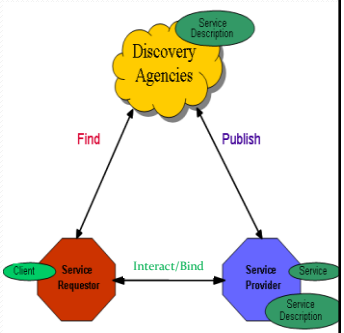
Terminology



Terminology (Continued)

Components

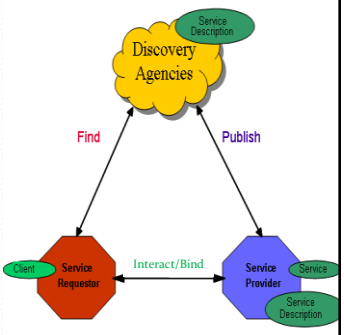
1. Service: A service is an implementation of a well-defined piece of business functionality, with a published interface that is discoverable and can be used by service consumers when building different applications and business processes.



Terminology (Continued)

Components

2. Service Description: The service description contains the details of the interface and implementation of the service. This includes its data types, operations, and binding information. It could also include categorization and other meta data to facilitate discovery and utilization by requestors.



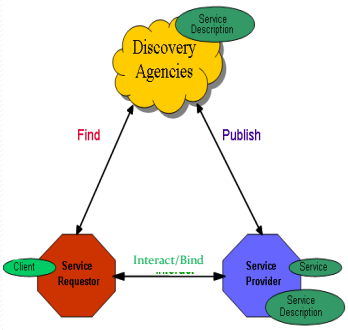
The complete description may be realized as a set of XML description documents.

The service description is usually published to a discovery agency.

Terminology (Continued)

Roles

1. Service requester (Client): The software entity that calls a service provider. Traditionally, this is termed a "client"; however, a service requester can be an end-user application or another service.

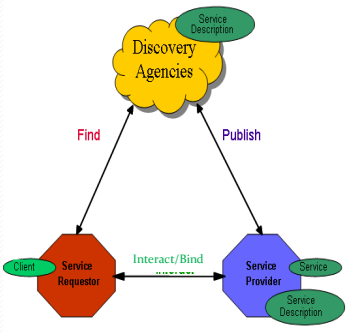


Terminology (Continued)

Roles

2. Service provider: The software entity that implements a service description, and publishes the service to a discovery agency.

3. Discovery agency (Registry): agency through which a Web service description is published and made discoverable.

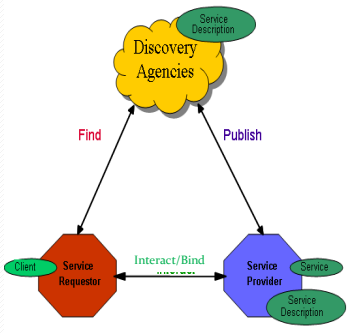


Terminology (Continued)

Operations

1. Publish: In order to be accessible, a service needs to publish its description such that the requestor can subsequently find it.

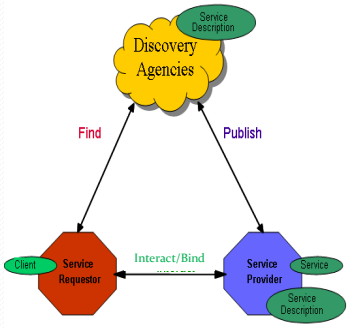
2. Find: The service requestor retrieves a service description directly or queries the registry for the type of service required.



Terminology (Continued)

Operations

3. Interact/Bind: The service requestor invokes or initiates an interaction with the service provider at runtime using the binding details in the service description to locate, contact, and invoke the service.



Terminology (Continued)

Other related terms and concepts:

Interface: Defines a set of public method signatures, logically grouped but providing no implementation. An interface defines a contract between the requestor and provider of a service. Any implementation of an interface must provide all methods.

Published interface: An interface that is uniquely identifiable and made available through a registry for clients to dynamically discover.

Terminology (Continued)

Asynchronous Messaging: Asynchronous means sender does not block when sends a request to the receiver.

Synchronous Messaging: Synchronous means sender blocks when sends a request to the receiver. i.e. The sender waits, after sending the request, until the receiver sends a response.

Summary

- ✓ Introduction
- ✓ Definitions of SOA
- ✓ Goals and Benefits of SOA
- ✓ SOA motivators
- ✓ Terminology

References

- Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, A. Brown et al, 2002.
- What is Service-Oriented Architecture, Hao He, 2003 (www.xml.com)
- Lecture Slides, Prof. Dr. Reiko Heckel
- Lecture Slides, M. All Babar
- http://www.w3.org/TR/2002/WD-ws-arch-20021114/

Questions?

Read about XML document…