Service Oriented Architecture

Course Code: SOA0010
Price:
Duration: 5 Days
DATE:

Who should attend:

Ideally suited to high-level enterprise application architects, developers and managers.

Prerequisite Skills:

An understanding of XML / Web Services and/or software architecture would be useful.

Course overview

Service-Oriented Architecture (SOA) expresses an architectural concept which defines the use of services to meet the requirements of software users. An SOA environment will consist of nodes on a network which make resources available to other participants in the network as independent services (for instance Web Service) which are accessed in a standardised way.

Due to the highly interoperable nature of SOA services, technologies such as Java and .NET can co-exist. Software components tend to be reusable, so for instance a C#.NET service may be used by a Java application, and/or any other programming language which can access this service, as the interface can be defined in a standards-compliant manner which 'hides' any vendor- or language-specific implementation from the calling service. With so many "SOA badged" tools available it can be hard to know where to start looking, or even to know what you should be looking for. Our instructors' impart their real-world experience of implementing SOA in large commercial, retail, banking & financial organisations in a series of case studies aimed to help you make the right decisions about your SOA strategy. This course is non vendor-specific so you'll get good, imparital advice.

What you will learn

- What is a Service Oriented Architecture?
- Advantages of SOA
- Defining a Service in WSDL
- Simple Object Access Protocol (SOAP)
- Service Oriented Analysis & Design (SOAD)
- Enterprise Service Bus (ESB)
- Business Process Implementation
- Layered Architecture
- Process Driven Services
- Transaction Management
- Overview of popular SOA platforms

A Service Oriented Methodology

- Introduction to a SOA adoption roadmap
- Service lifecycle
- Three analysis approaches
- Service oriented analysis
- Service oriented design
- Introduction to service oriented patterns

Advantages of SOA

- Traditional EAI Approach
- Problems With Traditional EAI Approach
- Enter Service Oriented Architecture (SOA)
- Build the Services
- We Can Easily Change the Process
- Change Flow Using Legacy Approach
- Replacing an Application
- Other Advantages
- Business Advantages
- Adoption Stages

SOA Past and Present

- From XML to Web Service to SOA
- How SOA was done before
- Emerging standards for SOA
- Compare SOA with other architectures

A Service Oriented Methodology

- What is SOA?
- Creating a common understanding of SOA
- The evolution of SOA
- Introduce the concepts of services and SOA
- Design principles of SOA
- The relationship between SOA and web services
- The advantages and risks of SOA

Introduction to modelling and UML

- Why use models with SOA.
- Difference between model and methodology.
- Why use the Unified Modeling Language?
- Introducing UML, the notation.
- Identifying business processes.
- Notation, Patterns and Methodology.
- Which Methodology to choose?

Fundamental Concepts

- Building from components.
- Modeling concepts.
- What is an object?
- Containment.
- Messages and methods.
- Object interaction.
- Exercise: testing some basic concepts.

Introduction to Business Process

- How a collection of services perform a task.
- Simple request response interaction
- Complex interaction involving many services.
- Need for a coordinator service emerges.
- Birth of orchestration or business process.
- Composing processes using processes.
- Business Process Execution Language (BPEL)
- Example business processes

Service Enablement

- Basic web services elements
- Core web services standards stack
- The Importance of WSDL
- The design of SOAP
- The use of registries via UDDI
- The basic concepts of service orientation

Distributing Services Across a Network

- Aligning functional and nonfunctional requirements
- The role of Intermediaries In Service Networks
- Introductions to WS-* Extensions
- SOA Tenets

Modeling SOA building blocks

- Using UML to analyse and design interfaces
- Generating a domain model
- Implementing and realising
- Use Cases Showing web service collaboration
- Usage of communication diagrams

Enterprise Service Bus (ESB)

- Objectives
- Service Invocation
- Legacy System Integration
- Web Services to the Rescue
- The role of ESB in SOA
- Security and ESB

Process Driven Services

- Service layer abstraction
- Introduction to business process layer
- Process patterns
- Orchestration and choreography
- The use WS-BPEL for process automation

A Service Oriented Reference Model

- Reference models and reference architectures
- The IMPACT SOA reference model and architecture
- SOA vendors and their relationship with SOA
- SOA support in .NET and J2EE platforms

Layered Architecture

- The layers pattern.
- Classic three-their architecture.
- Connecting to the domain layer.
- Linking to the User interface.
- Using packages to decompose a system.
- Avoiding mutual dependencies.
- What is layering and why we need them?
- Application service layer
- Business service layer
- Orchestration service layer

Conclusions

- The benefits of employing SOA
- Review of common business goals
- The risks associated with the SOA approach
- Evaluating tradeoff strategies