

SOA

Revised 11/10/2011

/training/etc

The Art of Knowledge.

This Page Intentionally Left Blank

Table of Contents

Introduction to SOA.....	1
Understanding SOA: A Technical Overview.....	2
Service Oriented Architecture (SOA) Analysis & Design.....	3
Mastering SOA and Web Services Developer's Workshop.....	4
Service Oriented Architecture – SOA Analysis.....	5
SOA, Web Services, and XML Security.....	6

This Page Intentionally Left Blank

Course Description: This course provides an introduction to the concepts of Service Oriented Architecture. This course discusses some of the key issues facing many organizations, especially dealing with integration among disparate systems. Participants will learn how SOA addresses these issues and its many other benefits. The course also discusses related technologies such as Enterprise Service Bus and its relationship with SOA. The participants then look at the various SOA patterns that can be used as a basis for developing SOA applications. Participants also go through case studies to better visualize the role of SOA.

Who Should Attend: This course is for J2EE developers, architects, project leaders, and project managers who would like to gain an understanding of SOA and its advantages.

Prerequisites: No specialized technical prerequisites are required for this course. However, a basic knowledge in IT systems and distributed computing is expected.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Describe the fundamentals of SOA.
- Understand the business advantages of SOA.
- Explain the SOA runtime elements.
- Describe the concept of an Enterprise Service Bus.
- Understand the role of Business Process Execution Language (BPEL) in SOA.
- Identify features of typical SOA Framework Products.
- Patterns Map to SOA.

Course Outline:

SOA Fundamentals

Objectives
 Defining Service Oriented Architecture (SOA)
 Putting SOA in Context
 SOA as an Alignment Strategy
 The SOA Umbrella
 What's a Service?
 Service Actors
 Serving Up SOA
 Business Process Management
 BPM & Workflow
 SOA Governance
 SOA Governance Model
 SOA Job Role Impact
 Services Please!
 SOA Re-Organization
 What Makes a Good Design?
 Is This a New Concept?
 Service Orienting the Enterprise
 Service Oriented Thinking
 SOA Is Perfect NOT!
 Service Characteristics
 When Do I Need a Service?
 About Services in SOA
 Contract-Driven Software
 SOA Standards
 Summary

Layers of Services

Objectives
 What Is Layering?
 SOA Layers
 Common Layers
 Auxiliary Layers
 Digesting the Layers
 The Application Service Layer
 The Business Service Layer
 The Orchestration Layer
 Layering Rules of Thumb
 Scope of a Service
 SOA User Interface
 Portal Site's Context Awareness
 Web 2.0 Data Aggregation
 Summary

SOA Value Proposition

Objectives
 The SOA Value Proposition
 Reducing Integration Expense
 Integration Costs Illustration
 Ripple effect of changes
 The Value of SOA Layering
 SOA Reduces Integration Costs

Increasing Asset Reuse
 SOA Economics/ROI
 Asset Reuse Illustration
 Service Reuse v. Object Reuse
 Increasing Business Agility
 Business Agility Illustration
 Traditional EAI Approach
 Problems with Traditional EAI Approach
 Change Flow Using Legacy Approach
 SOA Agility
 Build the Services
 Build the Process
 We Can Easily Change the Process
 Reducing Business Risk
 Risk reduction illustration
 SOA Eases Compliance Risk
 Other Advantages
 Business Advantages
 Hasn't This Been Said Before?
 ROI Quantification Hurdles
 Real World SOA Example 1
 Real World SOA Example 2
 Real World SOA Example 3
 Real World SOA Example 4
 Summary

Enterprise Service Bus (ESB)

Objectives
 SOA and the ESB Pattern
 Loose Coupling
 Service Invocation
 Business Process
 Data Integration
 Enterprise Service Bus (ESB)
 Legacy System Integration
 Unsupported Protocol
 The Role of ESB in SOA
 ESB: Software Artifacts
 Business Process
 Business Process: Example
 Minimum ESB Capabilities
 Minimum ESB Capabilities: Integration
 Minimum ESB Capabilities: Communication
 Minimum ESB Capabilities: Service Interaction
 Minimum ESB Capabilities: Management
 Security and ESB
 Summary

SOA Infrastructure and Tools

SOA Infrastructure and Tools
 Network Appliances
 Enterprise Service Bus
 Service Container

Business Rules Engine
 Business Activity Monitoring
 Service & Policy Management
 Business Process Engine
 Service Registry
 Service Repository
 Development Tools
 Tool Combinations
 What is the minimum?

Governance Fundamentals

Objectives
 The Strategic Role of IT
 Governing IT
 IT Infrastructure Library (ITIL)
 ITIL: Managing IT Activities
 ICT Infrastructure Management (ICTIM)
 ICTIM Processes
 Control Objectives for Information and related Technology (COBIT)
 Need for SOA Governance
 SOA Governance
 SOA Governance Adoption
 Key Governance Relationships
 SOA Governance Elements
 SOA Governance Procedures
 Recommended Procedures
 More Recommended Procedure
 SOA Governance Policies
 Enterprise Policies
 Business Policies
 Service Policies
 SOA Governance Metrics
 More SOA Governance Metrics
 Some Best Practices
 Relevant Standards Work
 OASIS SOA-RM
 Reference Model in Context
 OASIS SOA-RA
 Summary

Appendix A. Glossary

Appendix B. Introduction to Web Services

Objectives
 A Conceptual Look at Services
 Defining Services
 Service Communication Analogy
 Three Key Service Questions
 Connecting the Dots
 SOA: Runtime Implementation
 What Is a Web Service?
 Enterprise Assets as Services
 Typical Development Workflow

Course Description: Understanding Service-Oriented Architectures is a one-day, dynamic seminar geared for managers and project stakeholders who need to understand the impact of SOA, what is different about SOA and what is an incremental step from IT practices and approached of the past.

Who Should Attend: This an overview level SOA training course, designed for people who need to understand and manage existing or upcoming SOA projects. Experience with managing and working with enterprise applications will be helpful. We will explore the terminology, the specification, the processes and technologies specific to SOA.

Prerequisites: Attendees should have a minimum of two years working knowledge in the IT industry. A basic understanding of software development and web-based applications is necessary. Actual development working knowledge is helpful but not necessary.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Explain the concepts behind a SOA.
- Discuss how a common framework is embodied in both a technical infrastructure and an organizational entity in the form of governance.
- Understand the history of services-oriented architecture and what design processes led up to SOA.
- Discuss the challenges to adopting SOA in the enterprise.
- Understand the various standards, conventions, and best practices in implementing and supporting SOA.
- Explain how Enterprise Application Integration affects the reuse of existing applications.
- List the various roles involved in Service-oriented Analysis and Design (SOAD).
- Understand the importance of business process modeling.
- Relate a SOA maturity model and adoptance process to where an organization currently is and where they are trying to go.
- Discuss business process analysis and its relation to BPEL.
- Understand the difference between OO analysis and design and SOA analysis and design.
- Compare SOA best practices.
- Understand the responsibilities crucial to governance.

Course Outline:

SOA Overview

Introduction to SOA
Business impact/ROI
History
Myths/Reality
SOA and client/server
SOA and web services
Adoption issues

SOA: the Business Proposition

Leveraging business processes
Challenge to adoption
The SOAD Process
Roles and Skills
Web services standards
ESBs

Service-oriented Architecture

Principles
Business Process-driven development
Roles
Messaging
Transactions
Security
Business modeling
Design issues
EAI
ESB
Web services
RPC vs. REST
Web services, SOAP, and WSDL
Integrating legacy applications
Extending the lifetime of legacy apps
External integration
Governance
What needs to be governed and what is already governed?
Governing IT vs. SOA
Continuous improvement
Strategies

Defining Business Processes Using BPEL

BPEL Overview
Top-down Process Design
Bottom-up Process Design
Using WSDL
Process Elements and Properties

Select Expression Language
Partner Links
Variables
Fault Handler
Compensation Handlers
BPEL Process as a Service

Service-oriented Analysis and Design

OOAD vs. SOAD
Analysis
Design
Implementation
Process

SOA Best Practices

Planning
Standardizing
Designing
Managing
Implementing

Common Framework: Governance

Governance Overview
Importance
Responsibilities
Implementation

Common Framework: Infrastructure

Overview
Role in SOA
Security Issues
Scenarios and Analysis
ESB Issues

Web Services

W3C standards
WSDL
UDDI
Tools
Implementation technologies

Course Description: Service-Oriented Analysis and Design is a five day in-depth course geared for software architects and designers, this course explores what services and service-oriented architectures are and what best practices and design patterns to use in designing SOA-based applications. This course presents a strong perspective on services as an essential and important part of enterprise systems as well as how to identify, design, and develop of complex services using sound analysis and design techniques and best programming practices.

Who Should Attend: This an advanced level SOA training course, designed for architects and analysts who need to identify, design, and lead the implementation of SOA projects. We will explore and apply the terminology, the specification, the processes and technologies specific to SOA.

Prerequisites: Attendees should have an extensive working knowledge of developing enterprise applications. Designing and analysis working knowledge is also extremely beneficial.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Explain the business impact of SOA.
- Understand the history of services-oriented architecture and what design processes led up to SOA.
- Discuss the challenges to adopting SOA in the enterprise.
- Apply the concepts and principles of SOA to on-going and future projects.
- Understand the various web service standards available to support SOA.
- Explain how Enterprise Application Integration affects the reuse of existing applications.
- Relate what SOA means from architectural and development perspectives.
- Discuss business process analysis and its relation to BPEL.
- Understand the difference between OO analysis and design and SOA analysis and design.
- List the various roles involved in Service-oriented Analysis and Design (SOAD).
- Perform SOA Analysis to identify useful and manageable services.
- Perform SOA Design to craft architectures that support the required data and communication dynamics for identified services.
- Understand the importance of business process modeling.
- Understand the difference between RPC's and REST as a means of requesting resources.
- List the advantages of web services as a distributed systems technology.

Course Outline:

SOA Overview

Introduction to SOA
Business impact/ROI
History
Myths/Reality
SOA and client/server
SOA and web services
Adoption issues

SOA: the Business Proposition

Web services standards
ESBs
Leveraging business processes
Challenge to adoption
The SOAD Process
Roles and Skills

SOA: an Architectural Perspective

SOA Design Principles
RPC vs. REST
Web services, SOAP, and WSDL
Architectural Issues
SOA Maturity Model

SOA: a Development Perspective

The SOAD Process
Architecture
OO/UML
Services, operations and data
Policies
W3C standards
Web services standards
Testing

Service-Oriented Architecture

Principles

Business Process-driven development

Roles
Messaging
Transactions
Security
Business modeling
Design issues
EAI
ESB
Web services
RPC vs. REST
Web services, SOAP, and WSDL
Integrating legacy applications
Extending the lifetime of legacy apps
External integration
Governance
What needs to be governed and what is already governed?
Governing IT vs. SOA
Continuous improvement
Strategies

Defining Business Processes Using BPEL

BPEL Overview
Top-down Process Design
Bottom-up Process Design
Using WSDL
Process Elements and Properties
Select Expression Language
Partner Links
Variables
Fault Handler
Compensation Handlers
BPEL Process as a Service

Service-oriented Analysis and Design

OOAD vs. SOAD
Analysis
Design
Implementation
Process

SOA Analysis

Use cases
Identifying services
Operations and data formats
Error conditions
Service reusability
Identifying processes
Best practices

SOA Design

Business process modeling
XML and XML Schema
Asynchronous services
Callbacks
Messaging
Non-blocking calls
Best practices

SOA Best Practices

Planning
Standardizing
Designing
Managing
Implementing

SOA Patterns

Direct Connections
Broker Interactions
Serial Process Flows
Serial and Parallel Processes

SOA Anti-Patterns

SOA Adoption antipatterns
Service identification & design antipatterns
Service realization antipatterns

SOA Governance

Governance Overview
Importance
Responsibilities
Implementation

Enterprise Service Bus

Overview
Role in SOA
Security Issues
Scenarios and Analysis
ESB Issues

SCA and SDO

Overview
Service Components
Service Data Objects

Web Services Overview

W3C standards
WSDL
UDDI
Tools
Implementation technologies

Introduction to WSDL

Overview
Documents
Ports
Bindings
UDDI
Syntax

Introduction to SOAP

Overview

Syntax
Envelope
Header
Body
Fault
HTTP Binding

Messaging in SOA

Overview
Reliability
Delivery
Addressing
Security
Notification

Course Description: SOA and Web Services represent a groundbreaking evolution in distributed computing. The concepts are not altogether new, but the application of them, and the unanimous acceptance of core standards like HTTP, XML, SOAP, WSDL, and UDDI, has paved the way for XML Web Services and service-oriented architecture. Developers are introduced to the core standards that enable Web Services, and provide them hands-on experience with implementations of the Java XML and Web Service APIs including JAXP, JAXB, SAAJ, JAX-WS/RPC, WSEE, and XWSS.

Who Should Attend: This is an intermediate and beyond level web services training course, designed for J2EE developers and architects who need to identify, design, and implement web services.

Prerequisites: Students should have 1-2 years of working knowledge with Servlets and JSPs, and should be familiar with XML, Namespaces, and XML Schema.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Understand and work with Service-Oriented Architectures.
- Apply SOA Analysis and Design to the identification and implementation of web services.
- Understand and intelligently discuss Web Services and the core technologies involved.
- Develop and deploy real-world J2EE Web Services.
- Expose existing Java components as Web Services.
- Write Java components that access remote Web Services hosted by a third party.
- Read and understand a WSDL document.
- Parse, process, and respond to a SOAP message.
- Effectively use the tools and APIs provided by the JWSDP.
- Understand the concepts behind REST and implement a REST-based web service.
- Understand and work with the standards and technologies encompassed by WS-Security.
- Understand and utilize web service-related design patterns.

Course Outline:

Session: Introduction to SOA

SOA Overview
Thinking Services
Enabling Infrastructure
Analysis of Case Study

Session: Working with SOA

Layering of Services
Legacy Functions and Resources
Defining and Creating New Services
Implementing SOA
Web Services and SOA
Designing Services

Session: Path to Useful Web Services

SOA via the Web
Web Services Overview
Web Services in Action
Web Services, Java, and J2EE
Web Services Quickstart
Implementing a Web Service
Debugging Web Services

Session: Foundation for Web Services

XML: Basics, Namespaces and Schemas
Namespaces and Schemas
XML in Java: JAXP and JAXB
Using JAXB

Session: Binding: SOAP

SOAP Overview
SOAP in Action
SOAP in Detail
SAAJ
Creating a SOAP Message
Accessing a Service with a Servlet

Session: Description: WSDL

WSDL Overview
WSDL in Action

Session: Web Services in Java: JAX-WS/RPC

JAX-WS/RPC Overview
Working with JAX-WS/RPC
WSDL-generated service
WSDL-generated client

Session: Web Services in J2EE - WSEE

Implementing Enterprise Web Services

Session: REST in Java

REST Overview
Principles and Best Practices with REST Web Services
Publishing and Consuming REST Web Services in Java

Session: Finding Web Services

UDDI
Discovery in Action

Session: Security - WS-Security and Defenses

Securing Untrusted Input
Insecure Web Services
XML Signature and Encryption
Securing Web Services: WS-Security
Working with WSS
Message Handlers for Requests and Responses
Working with Handlers

Session: Best Practices

Best Practices Review
Web Services Design Patterns
Applying Design Patterns

Course Description: SOA Analysis is a three day in-depth course that explores what SOA is, the impact of SOA, what it means in terms of today's systems and architectures, and how to apply the concepts in identifying business and systems services. This course was designed for software analysts who wish to understand what services and service-oriented architectures are and what best practices and processes to use in supporting the design and implementation of SOA-based applications. This course presents a strong perspective on services as an essential and important part of enterprise systems as well as how to identify, design, and develop of complex services using sound analysis and design techniques.

Who Should Attend: This an intermediate and beyond level SOA training course, designed for analysts who need to identify and support the design of SOA applications and infrastructures. We will explore and apply the terminology, the specification, the processes and technologies specific to SOA.

Prerequisites: Attendees should have an extensive working knowledge with analyzing and working with enterprise applications. This is not a programming class.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Explain the business impact of SOA.
- Understand the history of services-oriented architecture and what design processes led up to SOA.
- Discuss the challenges to adopting SOA in the enterprise.
- Apply the concepts and principles of SOA to on-going and future projects.
- Explain how Enterprise Application Integration affects the reuse of existing applications.
- Relate what SOA means from an architectural perspective.
- Discuss business process analysis and its relation to BPEL.
- Understand the difference between OO analysis and design and SOA analysis and design.
- List the various roles involved in Service-oriented Analysis and Design (SOAD).
- Perform SOA Analysis to identify useful and manageable services.
- Understand the importance of business process modeling.

Course Outline:

SOA Overview

Introduction to SOA
Business impact/ROI
History
Myths/Reality
SOA and client/server
SOA and web services
Adoption issues

SOA: the Business Proposition

Web services standards
ESBs
Leveraging business processes
Challenge to adoption
The SOAD Process
Roles and Skills

SOA: an Architectural Perspective

SOA Design Principles
RPC vs. REST
Web services, SOAP, and WSDL
Architectural Issues
SOA Maturity Model

SOA: a Development Perspective

The SOAD Process
Architecture
OO/UML
Services, operations and data
Policies
W3C standards
Web services standards
Testing

Service-Oriented Architecture

Principles
Business Process-driven development
Roles
Messaging
Transactions
Security
Business modeling
Design issues

EAI
ESB
Web services
RPC vs. REST
Web services, SOAP, and WSDL
Integrating legacy applications
Extending the lifetime of legacy apps
External integration
Governance
What needs to be governed and what is already governed?
Governing IT vs. SOA
Continuous improvement
Strategies

Defining Business Processes Using BPEL

BPEL Overview
Top-down Process Design
Bottom-up Process Design
Using WSDL
Process Elements and Properties
Select Expression Language
Partner Links
Variables
Fault Handler
Compensation Handlers
BPEL Process as a Service

Service-oriented Analysis and Design

OOAD vs. SOAD
Analysis
Design
Implementation
Process

SOA Analysis

Use cases
Identifying services
Operations and data formats
Error conditions
Service reusability
Identifying processes
Best practices

SOA Design

Business process modeling
XML and XML Schema
Asynchronous services
Callbacks
Messaging
Non-blocking calls
Best practices

SOA Best Practices

Planning
Standardizing
Designing
Managing
Implementing

SOA Governance

Governance Overview
Importance
Responsibilities
Implementation

Course Description: The movement towards Web Services and Service Oriented architecture (SOA) paradigms requires new security paradigms to deal with new risks posed by these architectures. This session takes a pragmatic approach towards identifying Web Services security risks and selecting and applying countermeasures to the application, code, web servers, databases, application, and identity servers and related software.

Many enterprises are currently developing new Web Services and/or adding and acquiring Web Services functionality into existing applications -- now is the time to build security into the system!

Who Should Attend: This course is for those people who want to understand the real risks in SOA, WebServices, and XML.

Prerequisites: Students should have a basic understanding of SOA, web services, and XML. In addition, students will benefit more from the course if they have completed a basic course in web application security.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Architect security services in Web Services and SOA
- Understand how an attacker looks at Web Services
- Use best practices

Course Outline:

Web Services attack patterns

Common XML attack patterns

Data and XML security using WS-Security, SAML, XML Encryption and XML Digital Signature

Identifying services and federation with SAML and Liberty

Hardening Web Services servers

Input validation for Web Services

Integrating Web Services securely with backend resources and applications using WS-Trust

Secure Exception handling in Web Services

Understanding the impact of Web 2.0 technologies like Ajax, and REST on distributed systems security