Enterprise IT Architectures

SOA (Service Oriented Architecture)
SOA Introduction
Agenda of this Session

▪ Last Session:
  – Introduction of BPM (Business Process Management)
  – Focus on Capturing processes and automated execution

▪ This Session
  – BPM needs to be augmented by integration to applications, components, services, data bases etc.
  – SOA provides the mechanisms to do integration ("Integration Platform") such that both sides of an integration are independent
  – Methodology to structure distributed applications including business processes as well as user interfaces
Positioning of SOA

Now focusing on the “Integration Platform”
What is SOA

- SOA is an *architectural style* or approach whose goal is to achieve loose coupling among interacting software agents.

- All functions (that need to be used by more than one system) are defined as "services".

- Service providers agree to a defined, implementation-independent interface with service clients.

- Services oriented architecture is the *policies, practices and frameworks* that enable application functionality and IT services to be provided and requested as a set of services using a standards based form of interface.
Service Oriented Architecture
**Moves IT Logic Out of Services**

Services defined as units of business logic separated from...

- Flow of control and routing
- Data transformation and protocol transformation
SOA addressing IT as well as Business – common shift

Shift to a Service-Oriented Architecture

From

- Function oriented
- Build to last
- Prolonged development cycles

To

- Process oriented
- Build to change
- Incrementally built and deployed

- Application silos
- Tightly coupled
- Object oriented
- Known implementation

- Orchestrated solutions
- Loosely coupled
- Message oriented
- Abstraction
SOA is different things to different people

A set of services that a business wants to expose to customers and clients

an architectural style which requires a service provider, requestor and a service description.

a set of architectural principles and patterns which address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse, composable and single implementation.

A programming model complete with standards, tools, methods and technologies such as web services.
SOA Key Concepts
Key Models and Methods for SOA – Enabling greater flexibility in Enterprise IT Architectures

The SOMA Method: Service-Oriented Modeling and Architecture

The SOA Solution Stack: Layered solution view
SOMA (Service Oriented Modeling and Architecture) provides SOA Methodology

SOMA is about identification, specification, realization, implementation, and deployment of services, components and flows

**SOMA Method**

- **Startup / Adoption**
  << Input from: Business Analysis & Existing Assets>>

- **Identification**
  of Candidate Services and Flows

- **Specification**
  of Services, Components, and Flows

- **Realization**
  Decisions, Solution Templates & Patterns, Architecture, Technology Feasibility

- **Implementation**
  Build/Assembly, Testing

- **Deployment**
  Packaging and Provisioning

**SOA Solution Stack**

- **consumers**
  JService, Portlet, WSRP, B2B, Other

- **business processes**
  process choreography

- **services**
  atomic and composite

- **service components**

- **operational systems**
  Packaged Application, Custom Application, OO Application

- **Composite Service Registry**

- **Governance**
  QoS Layer (Security, Management, and Monitoring Infrastructure Service)

- **Data Architecture and Business Intelligence**

- **Integration (Enterprise Service Bus Approach)**

- **SOMA Method SOA Solution Stack**

- **SOMA Methodology**

© 2012 IBM Corporation
SOA Layered View (Solution Stack)
IBM SOA Foundation Reference Model

**Strategy and Planning Services**
*Business-driven Enterprise Architecture and Standards*

**Business Services and Events**
*Supports the specification of enterprise business solutions through business architecture*

- **Interaction Services**
  - Enables collaboration between people, processes & information

- **Process Services**
  - Orchestrate and automate business processes

- **Information Services**
  - Manages diverse data and content in a unified manner

- **Partner Services**
  - Connect with trading partners

- **Business Application Services**
  - Build on a robust, scalable, and secure services environment

- **Access Services**
  - Facilitate interactions with existing information and application assets

**Asset and Registry Services**

**Infrastructure Services**
*Optimizes throughput, availability and utilization*

**Lifecycle Services**

**Development Services**

**Management Services**
*Manage and secure services, applications & resources*
The SOA Lifecycle (to be addressed in detail in Governance)

- Discover Services
- Construct & Test Services / UIs
- Compose Services
- Orchestrate Processes

- Gather & Model Requirements
- Model & Simulate
- Model SW Architecture

- Integrate People
- Integrate Processes
- Manage and integrate Information

- Manage Applications & Services
- Manage Identity & Compliance
- Monitor Business Metrics

- Financial Transparency
- Business/IT Alignment
- Process Control
Identification and Specification of Services (SOMA)
Step 1: Break down your business into components
- Decide what is strategically important, and what is just operations in the value chain domains
- Analyze the different KPIs attached to these components
- Prioritize and scope your transformation projects

Step 2: Define a Service Model
- Identify your services based on your business components
- Specify the services and components accordingly
- Make SOA realization decisions based on architectural decisions

Step 3: Implement a Service Model
- Develop a service-oriented architecture to support the Componentized Business
- Implement service based scoping policy for projects
- Implement appropriate governance mechanism

Business-Aligned IT Architecture
Example: Business Context Diagram for Business Process “Open Account” (Solution Viewpoint)

- Customer
- CSR (Store)
- Account Manager (HQ)
- Portal
- Account On-Boarding
- Forms
- Account History
- Credit Scoring Partner
- Account Owner (HQ)

To Be Process Model

As Is Process Model

Gap

Overlapping
Example: Use Case for JKE’s “Open Account”
SOA Modeling Constructs

**Business Processes**
(Flows)

**Services**
Atomic and Composite

**Service Components**

SOMA was created to specifically address modeling of all three constructs.
Introducing SOMA (Service Oriented Modeling and Architecture)

- SOMA is a business-driven modeling and design method

- SOMA provides in-depth guidance on how to move from the business models to the IT models required by SOA

- SOMA adds new service-oriented aspects and techniques in intelligent ways to enable an SOA with services directly traceable to business goals and requirements
At the heart of SOMA is identification, specification, realization and implementation of services, components and flows.

- Design is separated in Identification and Specification
- Realization are mainly decisions on how to implement, buy, or use existing assets
- Implementation and Deployment as “classical” Software Engineering
SOMA defines What we do and How we do it

**What we do?**

- Identification of candidate services and flows, leverageable existing assets
- Specification of services to be exposed, flows, and components (for realization of functionality)
- Realization captures realization decisions, selects solution templates, details SOA Solution Ref. Arch.
- Implementation incl. construction/generation, assembly, testing, deployment, monitoring and management

**How we do it?**

- Governance
- Input from: Business Analysis & Existing Assets
- Domain Decomposition
- Goal-Service Modeling
- Existing Asset Analysis
- Component Flow Specification
- Subsystem Analysis
- Service Specification
- Message & Event Specification
- Component Specification
- Information Specification
- Realization Decisions
- Solution Template & Pattern Selection
- Technical Feasibility Exploration
- Detail SOA Solution Architecture
- Service Flow Specification
- Solution Template
- Construction Generation
- Assembly Integration
- Unit Testing
- Integration Testing
- User Acceptance Testing
- Deployment (Packaging/Provisioning)
- Monitoring & Management

© 2012 IBM Corporation
Identifies Services

- **Domain Decomposition** *(Top-down Analysis)*
  - Process Decomposition
  - Functional Area Analysis
  - Information Analysis, Modeling, and Planning
  - Rule and Policy Analysis
  - Variation-Oriented Analysis

- **Existing Asset Analysis** *(Bottom-up Analysis)*

- **Goal-Service Modeling**

- **Additionally, Service Refactoring and Rationalization**
  - Service Litmus Tests
  - Exposure Decisions, including Exposure Scope
Service Identification Through 3 main Complimentary Techniques

Top-down Analysis
- Domain Decomposition
- Business Rules
- Business Use Cases

Goal Service Modelling

Service Candidates

Variation Oriented Analysis

Existing Asset Analysis

Data Analysis

Bottom-up Analysis
Service Design via SOMA – Service Identification

**Domain Decomposition**
- Techniques:
  - Process Modeling Tools
  - Design of KPIs/Metrics
- Services Identified
  - Open Account
  - Account Activation
  - Account Verification

**Goal Service Modeling**
- Techniques
  - Requirements Planning Tools
  - Design of KPIs/Metrics
- Services Identified
  - Determine Applicant Eligibility
  - Address Verification

**Existing Asset Analysis**
- Techniques
  - Asset Analysis Tools
  - Interviews/Documentation
- Services Identified
  - Account Inquiry (CICS 2.2)
  - AR Setup (CICS 2.2)
  - Account Setup (CICS 3.1)
  - Create Account (SAP)
Example: Domain Decomposition – Business Process Modeling for JKE’s “Open Account”
SOMA Specification uses comprehensive techniques to specify Services, Flows, and Service Components that Realize Services

- **Information Specification**
  - Data Model, Message Model, Business Glossary

- **Existing Asset Analysis – Fine Grained**
  - Determine the technical viability of existing applications and approaches to realize services

- **Service Specification**
  - Elaborates the Service Model, for example, service dependencies, service composition and flow, rules and policies, event specification, service operation, service message specification, QoS requirements, design decisions, and so on

- **Subsystem Analysis**
  - Partitions subsystems into service components that will be responsible for service realization

- **Component Specification**
  - Details component modeling, flow, information architecture, messages
Service Litmus Tests Are Gating Criteria Used to Determine If a Candidate Service Should Be Exposed

- 1. Business Alignment
- 2. Composability
- 3. Consolidation (Redundancy Elimination)
- 4. Technical Feasibility
- 5. [Externalized Service Description]
- 6. Project Defined/Customer Specific SLTs
Example: JK Enterprises Service Exposure Decisions

0. Open Account

1.1 Account Inquiry

1.2 Account Verification

1.3 Account Activation

1.2.1 Determine Eligibility

1.2.2 Address Verification

1.3.1 AR Setup

1.3.2 Account Setup

1.3.3 Create Account

Legend

= Service to be exposed
Enterprise IT Architectures

Service Model

Service Identification
- Service Portfolio
- Service Hierarchy

Service Specification
- Service Exposure
- Service Dependencies
- Service Composition & Flow
- Service Operations
- Service Messages
- Service Non-Functional Requirements
- State Management Decisions

Service Realisation
- Solution Templates
- Technical Feasibility
- Mapping to Reference Architecture

Service Implementation
- Assemble
- Deploy
- Manage
SOMA Realization (Includes SOA Solution Stack Instantiation)

- Select and instantiate Solution Templates and Patterns
- Technical Feasibility Exploration
  - Examine approaches to handle client requirements
  - Examine legacy application specific considerations
- Detail SOA Solution Stack
- Realization Decisions
  - Consider alternatives
  - Select the alternative
  - Provide justification
Iterative SOA Solution Design Process

As SOMA is applied during an engagement, we incrementally populate an architectural overview (“dashboard view”) of the SOA Solution.
SOA Layered View Details
Layer 1: Operational Systems (Leverage Existing Investment)

- Recognizes the value of existing IT investment
  - Use of existing “legacy” applications (e.g. COBOL application) and / or packages (e.g. SAP)

- Some SOA Related Activities:
  - Asset Inventory
  - Refactor existing applications to unlock business value

Examples for illustration: specifics are not in the scope of the reference model.
Layer 2: Service Components

- **The Service Component Layer:**
  - Enables IT flexibility by strengthening the decoupling in the system. Decoupling is achieved by hiding volatile implementation details from consumers.
  - Often employs container based technologies like EJBs

- **Each Service Component:**
  - Provides an enforcement point for service realization
  - Offers a facade behind which IT is free to do what they want/need to do
Layer 3: Services (Decouple Business and IT)

- The Services Layer forms the basis for the decoupling of Business and IT.
  - Captures the functional contract (incl. QoS – Quality of Service) for each standalone business function or each task in a business process

- The assumption is that (within an SOA) IT responsibility is to realize/manage service implementations that faithfully conform to the set of services in the service model.

- This layer contains all the exposed services in the SOA

- Each service is a contract between the consumer(s) and the provider(s)
Layer 4: Business Processes (Business process alignment of IT)

- This layer contains operational IT artifacts that implement business processes as a choreography of services.
- The set of services that are composed is restricted to those services that are defined in Layer 3.
- The choice of technology depends on a set of realization decisions that must be made when establishing a physical Reference Model for a given SOA.
Layer 5: The Consumer Layer
(Channel independent access to business processes)

- This layer exists to recognize that the technology chosen to expose Business Processes/Services must permit access from a wide set of interaction channels.
- It is important to populate this layer with the set of channels types that are required in a solution.
Cross-cutting concerns/capabilities

- Several concerns are not restricted to a single layer in the Reference Model, these concerns are captured in ‘Layers’ 6-9.
- These are not really layers but treating them as such gives us the ability focus discussions/decisions, for example “What is found where Governance intersects Services? i.e. what are the Governance concerns specific to Services?”
- Clearly there is interaction among these ‘layers’ also. For example, it is likely that most data architectures will be subject to governance.

for illustration: this is not saying that SOA requires an ESB.
Example JK Enterprise – a virtual company with an „Open Account Process“
Designing BPM / SOA Application: Process Modeling
Designing BPM / SOA Application: Layered View

Integration of “Legacy”

Architecture
Home Work: Exercise Layered View

- Usually a diagram (or set) which is used as a basis for discussion and explanation.

- Assume you will create many iterations of this document.

- Should contain processes, services, components, and operational systems
Exercise – SOA Solution Layer Perspective – Add Missing Components

Among the missing artifacts from this diagram, the Service Components (service realization)
Also missing are To-Be supporting operational systems
SOA Reference Model
IBM SOA Foundation Reference Model

**Strategy and Planning Services**
*Business-driven Enterprise Architecture and Standards*

**Business Services and Events**
*Supports the specification of enterprise business solutions through business architecture*

- **Interaction Services**
  - Enables collaboration between people, processes & information

- **Process Services**
  - Orchestrate and automate business processes

- **Information Services**
  - Manages diverse data and content in a unified manner

**Development Services**
- **Partner Services**
  - Connect with trading partners

**Management Services**
- **Business Application Services**
  - Build on a robust, scalable, and secure services environment

- **Access Services**
  - Facilitate interactions with existing information and application assets

**Asset and Registry Services**

**Infrastructure Services**
*Optimizes throughput, availability and utilization*

**Lifecycle Services**
*Manage and secure services, applications & resources*
Separation of Concerns: Example “Open Account” Process

The SOA Reference Architecture in Action

Business Services and Events
- Business-driven Enterprise Architecture and Standards
- Enables collaboration between people, processes & information
- Federated Query
- Manage and secure services, applications

Interaction Services
- Portal
- Build on a robust, scalable, secure services environment
- EJBs

Process Services
- Orchestrate and automate business processes

Information Services
- Manages diverse data in a unified manner
- Federated Query
- Connect with trading partners

Partner Services
- Build on a robust, scalable, secure services environment
- CICS Access
- Siebel Access
- DB Access

Business App Services
- Community Manager
- Approval
- Open Account

Development Services
- Integrated environment for design and creation of solution assets

IT Service Management
- Business Dashboard
- IT Management Console
- Manage and secure services, applications

IT Management
- Console

Infrastructure Services
- Optimizes throughput, availability and performance

Open Account
- Business Dashboard
- IT Management Console

© 2012 IBM Corporation
ESB (Enterprise Service Bus)
ESB (Enterprise Service Bus) – Definition and Purpose

- An Enterprise Service Bus (ESB) is an architectural pattern defining a flexible connectivity infrastructure for integrating applications and services.

- The architecture pattern is a guiding principle to enable the integration and federation of multiple service bus instantiations.

- An ESB performs:
  - Routing messages between services
  - Converting transport protocols between requestor and service – managing multiple protocols
  - Transforming message content between requestor and service
  - Handling business events from disparate sources
ESB (Enterprise Service Bus) – Service Virtualization

- ESB acts as an intermediary (proxy) between requestor and provider

- ESB provides service virtualization of
  - Location and identity
  - Interaction protocol
  - Interface

- Interactions are decoupled, supporting separation of concerns
ESB is today’s technology

Direct Connectivity (No middleware)

Connectivity, mediation & custom adaptation logic

Application

All connectivity, mediation and custom logic buried within the application.

Message Queuing / CORBA

Connectivity logic

Mediation & custom adaptation logic

Application

Removes the connectivity logic from the application

Enterprise Application Integration

Connectivity and mediation logic

Custom adaptation logic

Application

Removes the connectivity + mediation logic from the application

Enterprise Service Bus

Connectivity, mediation & custom adaptation logic

Application

Application as a service

Reduces application to its core business functions (i.e. a service)

Reduced development and maintenance; increased flexibility and reuse
ESB Pattern in Action – Retail Scenario
Encapsulate components for reuse
All components (e.g., services, rules, human interactions) are represented consistently and invoked identically
Standard SCA (Service Component Architecture) – Component Assembly
SCA (Service Component Architecture) – Example Part 1

- Modules: Encapsulate and Reuse Functionality
- Libraries: Share common definitions

Module: Process Order

- doOrder
- Process: Order
  - Get Customer Status
  - Approve Order
  - Store Order

Module: Approve Order Manually

- Customer Status
- Approve Order
- Human Task: Approve Order

Module: Customer Status

- Business Rule: Get Customer Status

Library: OrderLib

- BO:Order
- IF: StoreOrder

Module: Update Order Database

- Store Order
- Interface Map Convert to DB2
- Adapter for Relational DB

DB2
• Store Order in SAP instead of DB2
• No effect on common objects or consumers
Expanded View of the Enterprise Service Bus

Interaction, Process, Information, Partner, Business App, Access Services

Business Logic

Enterprise Service Bus

Interaction Patterns

Message Flows

Mediation Patterns

Message Models

Transport Protocols

Security

Management

Registry

IT Management Services
ESB – Multi-protocol Exchange – Intermediary decoupling heterogeneous consumers and suppliers

Enterprise IT Architectures

Domain of interest - Intranet Exchange

- WebSphere (WAS/Portal)
- .NET Client
- Some Client
- XML/MQ Client
- Text/MQ Client

ESB

- SOAP/JMS
- SOAP/HTTP
- XML/HTTP
- XML/MQ
- Text/MQ

Tooling

- SOAP/JMS
- SOAP/HTTP
- XML/HTTP
- COBOL Copybook/MQ
- CICS

- WebSphere provider
- .NET provider
- Some provider
Example of ESB use
Example A of ESB use: Multiple Channel Access to Backend Service

- **Integration Developer**
  - Review Export and Import
  - Build mediation flows
  - Deploy Service Module

- **Account System**
  - Java Client Appl

- **High Value Accounts**
  - J2EE Appl

- **ESB**
  - Service Import
    - HQ Eligibility
  - Service Export
    - ACT (SOAP/HTTP)
    - HVA (XML/JMS)

- **Determine Eligibility Service Module**
  - Eligibility Mediation
  - Transform Request/Response

- **Applicant Eligibility Service**
  - J2EE Appl
Example B of ESB Use: Create SAP Service

- Deploy Adapter
- Discover Enterprise Service
- Generate BAPI Business Objects
- Deploy/Test Service Import

Websphere Integration Developer (WID)

Test Client

Enterprise Service Discovery

ESB

Service Import
SAP Outbound

Business Objects

WebSphere SAP Adapter

Create Account Service Module

RFC/BAPI

SAP
Create New Customer Record
Interaction Services
Interaction Services: Using Portal As the “Front End” of SOA

Presentation Services

- Web Browser
- Rich Clients
- Mobile Client

MS Office & Windows

eForms
Xforms
What is an Interaction Service?

- Portlets can be
  - A Service Consumer (1)
  - A Service Provider (3)
- Portlets can
  - Initiate processes (1)
  - Act as a Participant in a process (3)
  - Communicate with each other

The Portal Framework Provides Service Aggregation

The Portal Framework Provides Service Aggregation

UI

Portlets

Enterprise Service Bus

Service A
WMQ

Request/Response
Coarse Grained

Service B
SOAP/HTTP

Request
Fine Grained

Service C
SOAP/JMS

Request/Response
Fine Grained

Service D
HTTPS

Request/Response
Coarse Grained

1

2

3

4
Information Services
Information Services in SOA Reference Architecture

- Delivering actionable information to people and processes

- Connect, enhance and deliver in-context information across diverse operating systems, applications and legacy systems through reusable services

- The Information Services enables consistent views and maintenance of data and content, providing a “single view of the truth” to people and processes
Information Services: Several Patterns

1. Information Service Enablement
   - Store/Retrieve Application Database
   - Account Open Process
   - Lookup Customer
   - Customer Master
   - Account Data
   - Account Application Data

2. Integrated information services
   - Store/Update Customer
   - Customer Master
   - Account Data
   - Account Application Data

3. Master Data Management
   - Store/Update Customer
   - Customer Master
   - Account Data
   - Account Application Data

4. Content Management
   - Request Documentation
   - Account Documents
Information Services: Pattern – Deliver Your Data Virtualized Through Services

- **As-Is Environment**
  - Data resides in disparate sources
  - Manual & redundant integration of data by multiple consumers results in high costs and inconsistent/inaccurate data
  - Slow response time due to inefficient real-time access

- **Solution Characteristics**
  - On demand integration instead of redundant data
  - Transparent & optimized access to distributed, heterogeneous sources

- **Results**
  - Real-time access to distributed information, fast response time
  - Scalable approach for adding more data sources
Closing Remark
Just remember – the future might bring more than you think

“I think there is a world market for maybe five computers.”
Thomas Watson, chairman of IBM, 1943

“Computers in the future may weigh no more than 1.5 tons.”
Popular Mechanics, 1949

“There is no reason anyone would want a computer in their home.”
Ken Olsen, founder of DEC, 1977

“Prediction is difficult, especially about the future”
Niels Bohr, 1957

“640K ought to be enough for anybody.”
Bill Gates, 1981
Questions