## Icons / Metaphors

<table>
<thead>
<tr>
<th>Icon</th>
<th>Metaphor</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍌</td>
<td>Information</td>
</tr>
<tr>
<td>💡</td>
<td>Common Realization</td>
</tr>
<tr>
<td>📚</td>
<td>Knowledge/Competency Pattern</td>
</tr>
<tr>
<td>🗯</td>
<td>Governance</td>
</tr>
<tr>
<td>🧑‍🤝‍🧑</td>
<td>Alignment</td>
</tr>
<tr>
<td>🧑‍🍳</td>
<td>Solution Approach</td>
</tr>
</tbody>
</table>
Agenda

1. Introducing SOA
2. SOA Patterns
3. SOA Methodologies
4. Mainstream SOA Implementations
5. SOA Implementation Case Study
6. Conclusion

Barriers to Business Agility

- Lack of business process and integration
- Limited architectural vision
- Multiple applications used to support common LOB needs
- Lack of overall roadmap to unite business needs with underlying technical infrastructure
Barriers to Business Needs

What's Preventing Them:
- Processes are inflexible or unclear
- Integration challenges prevent timely implementation
- Long lag-times and inadequate alerting mechanisms prevent insight or action
- Changes had been expensive with long payback periods

What Businesses Want:
- A way to change their operational processes quickly
- To deploy automated processes fast
- A real-time view of operations and be able to intervene
- To see results and value fast

Companies want change at the speed of business.

SOA Mainstream Definitions

"The policies, practices, frameworks that enable application functionality to be provided and consumed as sets of services published at a granularity relevant to the service consumer that can be invoked, published and discovered, which are abstracted away from the implementation using a single, standards-based form of interface.

"Service-oriented architecture is a client/server software design approach in which an application consists of software services and software service consumers (also known as clients or service requesters). SOA differs from the more general client/server model in its definitive emphasis on loose coupling between software components, and in its use of separately standing interfaces.

"A set of components which can be invoked, and whose interface descriptions can be published and discovered."
SOA in a Nutshell

**Definition**
Service Oriented Architecture is an Enterprise architecture paradigm that supports integrating and orchestrating a business process using a set of well defined services in a technology-neutral manner. **Four key aspects of SOA are:**
- Functionality available as reusable services
- Orchestration of services to implement a business process
- Real-time access to key business performance metrics
- Monitoring & Management through a governance model

**Benefits**
- Business Responsiveness: SOA is business process-centric and enables an IT organization adapt to changing business needs
- Re-use of IT Assets: SOA helps leverage existing IT investments and develop a long-term infrastructure
- Lower Integration Cost: SOA implementations are standards based, reducing costs and avoiding vendor lock-ins
- Business Activity Monitoring: SOA helps real-time tracking of key business indicators and alerts the user to business issues and opportunities

SOA View by Roles

<table>
<thead>
<tr>
<th>SOA Perspective</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CEO</strong></td>
<td></td>
</tr>
</tbody>
</table>
| A set of **services** that a business wants to expose to their customers and partners, or other portions of the organization | • Agile enterprise  
• Adaptive business processes  
• Responsive business model  
• Measurable key performance indicators  
• Speed to Market |
| **CTO**         |          |
| A set of **architectural principles**, patterns and criteria which address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse, composite services and single implementation | • Flexible systems  
• Able to support rapid changes in the business  
• Effective planning by Process Modeling  
• Lower Risk & faster Time to Value  
• Greater reliability & consistency  
• Reduced maintenance costs |
| **CIO**         |          |
| A **programming model** complete with standards, tools and technologies such as Web Services | • Standards based IT assets  
• Scalable Applications  
• Manageable Application portfolio  
• Maintainable IT and business systems  
• Extensible Infrastructure |
SOA Goals Summarized

**Business**
- Enable business agility, reduce time to market
- Enable to handle future growth in volume of business
- Reduce risk of business disruption
- Access real time business information accurately and rapidly
- Lower inventory costs
- Improve return on assets
- Accelerate mergers, acquisitions and alliances

**IT**
- Simplify technology infrastructure and maintenance cost
- Facilitate reuse and reduce development cost
- Simplify complex programming tasks
- Enable application convergence and reengineering
- Reduce redundancy
- Optimize cost of operations

SOA Value Proposition

Measure of SOA Overall Benefit = RO Investment + RO Assets + RO Effectiveness
SOA Architectural Patterns

SOA abstract the software functionality that business processes compose and orchestrate.

SOI enables loose coupling and coarse granularity.

SOM enforces the Quality of Service of SOI.

Web Services Security & Identity Management
Essential prerequisite for SOA.

SOM Capabilities:
- Encapsulation and Composition
- Web service instance management
- Dynamic routing
- Transport protocol translation
- Web services caching
- Synchronous / asynchronous conversion

SOA – Service Oriented Architecture
SOP – Service Oriented Process
SOI – Service Oriented Integration
SOM – Service Oriented Management
Common SOI-Related Patterns

Point to point exposure
- Expose existing embedded functionality rapidly
- Provided multi-point of access to given functionality

Service Adaptor/Connector
- Wraps a legacy function making it callable via Web services
- Enables consumers to access functionality that is not service-enabled

Service Proxy
- Supports access to service in the absence of direct access to the service provider’s service description and ability to directly invoke the service
- Provides consumers with an SOA interface

Remote Service Strategy
- Provide flexibility in the choice of the service provider
- Enables changing service provider based on QoS or functionality needs

Single Point of Access
- Provides one access point to a number of potential variants in functionality
- Eliminates redundant functionality and refracts/consolidates/replaces existing functionality as needed

Virtual Provider
- Focuses on one project or LOB at a time, yet relies on others for some functions not yet exposed as services
- Enables ramping up service critical mass in the presence of non-existent providers

Service Integrator (degenerated ESB)
- Provides single point of access
- Enables routing and transformation

Enterprise Service Bus
- Supports a general Enterprise integration approach
- Enables mediation, routing, transformation, policies, rules, events both inside the organization or between partners in the ecosystem / valuenet

Sample SOA Enterprise Reference Architecture
SOA Platform – Piecemeal Selection of Capabilities

- Other capabilities include security, auditing, service registry/repository, and namespace directory
- A Services Platform should also support distributed geographical deployment with a common model for management and administration
- Management capabilities include administration
  - Management capabilities provide a point of control over service addressing and naming

- Communications capabilities include routing, addressing, messaging styles, and transport protocols
- Communications capabilities provide location transparency and support for service substitution
  - Communications capabilities include routing, addressing, messaging styles, and transport protocols
  - Communications capabilities provide location transparency and support for service substitution

- Integration capabilities include integration styles/adapters, protocol transformation
- Integration capabilities provide support for integration in heterogeneous Services Platform environments and service substitution

- Interaction capabilities include service interface definition, service messaging model, substitution of service implementation
- Interaction capabilities provide support for SOA principles and allow the separation of application code from specific service protocols and implementations

Leveraging Integration Styles in an ESB Context

- ESB Communications Capabilities (routing, etc.)
- ESB Integration Capabilities (transformation, etc.)
- ESB Service Interaction Capabilities (mediation, etc.)

Note: Virtual Service Provider (VSP) with Remote Service Strategy provides a single point of access to services
Service Integrator (i.e., VSP with integration/mgmt rule objects) provides a single point of integration for services
ESB provides a common infrastructure for Service Communications, Integration, Interaction, and a point of control for Service Management, Security, and Auditing

Note: Access and Integration Ports are associated to a set of addresses that support particular protocols
ESB Gateway Integration Pattern

- ESB Gateway acts as a proxy to provide controlled access to Enterprise ESB.
- ESB Gateway exposes services to external parties as well as allows internal applications to access external services in a secure and controlled manner.

Enterprise ESB Integration with Other ESBs

- There are three possible patterns typically used to achieve this type of integration:
  - Direct ESB integration
  - Brokered ESB integration
  - Federated ESB integration

- A brokered ESB pattern with cascading registries is typically a best fit recommendation.
Typical SOA Methodologies

<table>
<thead>
<tr>
<th>Typical SOA Methodology</th>
<th>TOGAF ADM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business &amp; IT Requirements</td>
<td>Preliminary Phase</td>
</tr>
<tr>
<td>Enterprise Drivers - Business &amp; IT</td>
<td></td>
</tr>
<tr>
<td>Strategy</td>
<td>Architecture Vision</td>
</tr>
<tr>
<td>• Assess Enterprise Maturity</td>
<td>• Business Architecture</td>
</tr>
<tr>
<td>• Define SOA Strategy and Vision</td>
<td>• Information System Architecture</td>
</tr>
<tr>
<td>• Assess SOA Program Readiness</td>
<td>• Technology Architecture</td>
</tr>
<tr>
<td>• Set-up SOA Governance</td>
<td>• Opportunities and Solutions</td>
</tr>
<tr>
<td>Architecture</td>
<td>• Migration Planning</td>
</tr>
<tr>
<td>• Define Architecture</td>
<td>• Architecture Change</td>
</tr>
<tr>
<td>• Assess Architecture</td>
<td>• Management</td>
</tr>
<tr>
<td>• Define Architecture Plan</td>
<td></td>
</tr>
<tr>
<td>• Define Program Management Plan</td>
<td></td>
</tr>
<tr>
<td>Composition and Assembly</td>
<td>Implementation Governance</td>
</tr>
<tr>
<td>• Analyze Services</td>
<td>(covers design-time)</td>
</tr>
<tr>
<td>• Design Services</td>
<td></td>
</tr>
<tr>
<td>• Develop Services</td>
<td></td>
</tr>
<tr>
<td>• Test Services</td>
<td></td>
</tr>
<tr>
<td>Deploy and Manage</td>
<td>Implementation Governance</td>
</tr>
<tr>
<td>• Deploy Services</td>
<td>(covers run-time &amp; change-time)</td>
</tr>
<tr>
<td>• Manage Services</td>
<td></td>
</tr>
<tr>
<td>• Monitor Services</td>
<td></td>
</tr>
</tbody>
</table>

Typical methodology & mapping to TOGAF
Typical SOA Methodology at Work

**Analyze**
- Business Service requirement definition – Business Use Case Driven
- Business Use Case Realization – Business Process Model
- Service Building Blocks Identification and Reconciliation
- Service Specification
- Service Design
- Business Policies Specification
- Service Management Specification – Service Level Agreement (Performance, Availability)
- Security Requirements
- Business Activity Monitoring Specification

**Design**
- Service Design
  - Contract
  - Interface
  - Messages
  - Business Rules
  - Service Component
  - Association
  - Security
- Design of Components
  - Revision with/without
  - Modification, build new
  - Patterns based on Non-Functional Requirements
- Service Choreography
- Process Choreography
- Design of Process and Service
- Management (BAM, Performance and Availability)

**Develop**
- Infrastructure Component Implementation
- Business Component Implementation
- Service Implementation
- Service End Points
- Messages
- Security Policies
- Service Assembly
- Business Function
- Business Transaction
- Technical Function
- Process Assembly
- Business Processes
- Transformation
- Mediation
- Human Workflow
- Service and Process Management Implementation

**Monitor**
- Business Process Performance
- Business Events
- Business Exception
- Security Events
- SLA Conformance and Exception
- Alerts and Escalation
- Real-time Key Performance Indices
- Business Analytics

**Deploy**
- Infrastructure Provisioning
- Ecosystem Readiness
- Service Configuration
  - Registry Setup
  - Security Setup
  - Process Consumer configuration
  - Registry Setup
  - Rules Deployment
  - Service Deployment
  - Component Deployment
  - Service Bus Configuration
  - BAM Setup
  - Service Management Configuration

**Service Lifecycle**

Service Provisioning

Solution Implementation

Service Consumption

Discover

Request

Deposy

Certify

Publish

Execute

Version

Retire

Service Implementation

21

22
Sample Specialized Maturity Model for SOA

Agile
Responsive
Business Strategic
(Service Oriented Business Transformation)
IT Strategic
(Service Oriented IT Transformation)
Standardized
(WS/SOA implementations standardized across IT)
Tactical
(Project Driven SOA/Web Services Implementation)

Defined Service Maturity Model for an Enterprise to establish benchmarks and standards to measure SOA adoption and maturity based on business impact.

SOA Incubation – “Where to start?” “How to grow?”

Good Starting points are enumerated below – ideal incubation combines Business/IT transformational pilot.

Revenue Drivers & Market Forces dictate…

IT Responsiveness, Game Changing Transformation needs….

Inefficiency, Cost of IT, Integration drives…

IT Driven Business Driven

SOA Incubation “Under the Radar”

SOA Incubation “Getting IT ready for business Challenges”

SOA Incubation “Ideal Sweet Spot”

SOA Incubation “The Consultant’s Dream”

Shortest Path to “Quick Hit”

The Challenge – Initiating, Sustaining, Managing this Transformation.
SOA Reference Architecture Levels

Recommended Organizational Alignment
Typical SOA Service Offerings

- Architecture driven by common technology and design patterns
- Top-down, end-to-end SOA adoption in a heterogeneous environment
- Domain-specific, business process based SOA solutions to address a well defined problem space

Sample SOA Lifecycle Services

- SOA Project and Program Management
- Service Oriented Design and Realization
- SOA Solution Development, Delivery, and Implementation
- Service Oriented Testing
- SOA Business Value Assessment
- SOA Maturity Assessment and Roadmap Definition
- SOA Governance Modeling
- SOA Architecture Assessment
- SOA Business Architecture and Process Modeling
- SOA Architecture Definition
- SOA Stack Selection
- Proof of Concept and Technology Configuration
- SOA Governance and Management
- Business Activity Monitoring
Sample SOA Domain Solutions

Domain specific, business process based SOA solutions portfolio to address common & well defined industry solution patterns.

**Banking:** Credit Card processing application framework

**Travel & Hospitality:** Worldwide travel reservation framework

**Healthcare:** Patient Personal Health Record (PHR) application and framework

**Insurance:** WBSF based Insurance industry service integration framework

Sample SOA Entry Points

**SOA Entry Points** - Architecture driven by common technology and design patterns

**BPM**
- Model, simulate, optimize and implement business processes leveraging existing assets and deploy new services using SOA principles.

**Legacy Modernization**
- SOA patterns to modernize Legacy Mainframe applications leveraging existing assets

**SOA Infrastructure Management**
- Consolidate servers, leverage virtualization, grid and utility computing to provide scalable on demand SOA infrastructure.

**Portals**
- Standardized front-end SOA patterns for improving collaboration, productivity and user experience

**Service Oriented Integration**
- ESB patterns to integrate COTS, SAP, Legacy applications and Web Services

**Information Services**
- Data & information services for business entities for providing single version of truth by abstracting underlying data sources

**Key Benefits**
- Optimize Core and Out Source Ancillaries – Produces measurable business value
- Reuse existing assets. Support changing business needs
- Reduce TCO and Improve manageability and dynamic scalability
- Improves Collaboration and Standardizes Service Delivery
- Integrates IT systems across enterprise / businesses in a standardize way improves connectivity
- Business Aligns Data Managed Strategy
### Detailed List of Entry Points to SOA Ecosystem

<table>
<thead>
<tr>
<th>Entry Point</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPM</td>
<td>Model, Simulate, Optimize and Implement Business Processes leveraging existing assets and new services</td>
<td>Optimize Core and Out Source Ancillaries – Produces measurable business value</td>
</tr>
<tr>
<td>Infrastructure As Service</td>
<td>Consolidate Servers, Leverage, Virtualization, Grid and Utility, Strategize Provisioning Approach for IT H/W</td>
<td>Optimize Core and Out Source Ancillaries – Produces measurable business value</td>
</tr>
<tr>
<td>Information As Service</td>
<td>Build Services for Business Entities, Abstract underlying Data Sources</td>
<td>Incremental Approach towards Single Source of Truth</td>
</tr>
<tr>
<td>Integration / ESB</td>
<td>Use ESBs to integrate COTS, SAP, Legacy Applications and Web Services</td>
<td>Integrates IT systems across enterprise / businesses in a standardize way improves connectivity</td>
</tr>
<tr>
<td>Application</td>
<td>Build SOA infrastructure using SAP and Domain Frameworks as the basis</td>
<td>Reuse existing assets and COTS solutions</td>
</tr>
<tr>
<td>Portal</td>
<td>Consolidate delivery channels through portals and other technologies</td>
<td>Improves Collaboration and Standardizes Service Delivery</td>
</tr>
</tbody>
</table>

### Using Entry Points to Support Horizontal Slicing

Vertical slicing refers to identifying key Business Processes and re-engineering them across the Presentation to Infrastructure layers using the SOA paradigm. Horizontal Slicing refers to approaching SOA from Infrastructure, Information, Integration, Application and Portal perspectives.
Components of SOA Governance

- **Registry**
  Central Catalog of Business Services

- **Repository**
  Storage of policies and other metadata for service governance

- **Rules Engine**
  Manage the declaration of policies and rules and automating their enforcement

- **Policy Enforcement Points**
  Agents that enact the actual policy enforcement and control at design-time, run-time, and change-time

- **Environment**
  Configure and define policies and for managing governance workflows across the service lifecycle

SOA Governance Modeling

- SOA Governance Modeling focuses on creating executable principles and policies required to manage and execute SOA Life Cycle in line with enterprise transformation strategy and implementing them. With a view to achieve compliance, reuse and optimization of service portfolio

- The focus areas of SOA Governance Modeling Service Offering are
  - **People Governance**
    Definition of principles that drive the allocation of Roles and Responsibilities to people across Business and IT organizations of the Transforming Enterprise
  - **Process Governance**
    Definition of policies and principles that drive the creation of SOA Life Cycle Processes including Transformation Strategy, Service Testing, Service Roll Out, Configuration Management etc.
  - **Service Portfolio Design Governance**
  - **Infrastructure Governance**
    Definition of principles that drive the allocation of infrastructure to services at design time and run time

Typical Timeline: Ongoing, 6 week iterations
SOA Governance – What is Being Addressed?

- Which division has the Quality Data?
- Who owns the new service?
- Who funds it?
- Who is responsible for upgrades?
- Who decides who can use the service and how often?
- What will be the IT Infrastructure?
- What can be a common business services?
- Which client applications will use these services?
- What are the common policies?
- Which unique policies to some of the services?
- How can we isolate them?
- Are there existing services already that can be reused?
- Are there existing services already that can be reused?
- What are the SLA for response-time, throughput capacity, information security, etc.?
- How do we monitor service quality against the SLA at runtime?
- Where do we alert or report any exceptions?
- How do we raise the SLA required by the business?
- How do you organize services for best reuse?
- How do you identify service dependencies?
- How do you manage change to services?
- Who is allowed to change and who approves?
- Who is responsible for upgrades?
- Which division has the Quality Data?
- Who owns the new service?
- Who funds it?
- Who is responsible for upgrades?
- Who decides who can use the service and how often?
- What will be the IT Infrastructure?
- What can be a common business services?
- Which client applications will use these services?
- What are the common policies?
- Which unique policies to some of the services?
- How can we isolate them?
- Are there existing services already that can be reused?
- Are there existing services already that can be reused?
- What are the SLA for response-time, throughput capacity, information security, etc.?
- How do we monitor service quality against the SLA at runtime?
- Where do we alert or report any exceptions?
- How do we raise the SLA required by the business?
- How do you organize services for best reuse?
- How do you identify service dependencies?
- How do you manage change to services?
- Who is allowed to change and who approves?
- Who is responsible for upgrades?

SOA Governance Scope

**Architecture**

- Establishing corporate technology standards
- Defining the high-level SOA architecture and topology
- Determining the SOA platform strategy
- Decide particular vendor products and technologies
- Specifying the management & characteristics of the SOA, (operations, and QoS, security, reliability and availability)
- Establishing criteria for SOA project design reviews

**Service Lifecycle**

**Design-Time**

- Fitness of a service as an enterprise-class asset
- Mapping Service to business requirement
- Strategic design of business services
- Validating that their interfaces and implementation
- Implementation conforms established design patterns and standards
- Establish usage policy & decision rights

**Run-time**

- Checking a service against business rules
- Securing services to only authorized consumers
- Checking that the consumers possess appropriate permissions
- Validating that services against corporate standards

**Change-time**

- Manage inter-service relationships and dependencies
- Performing impact analysis while changing a particular service
- Managing the rollout of services in run-time environment
- Managing custody transfers through the design, coding, testing & deployment
- Managing changes to existing policies and service level agreements
Agenda

1. Introducing SOA
2. SOA Patterns
3. SOA Methodologies
4. Mainstream SOA Implementations
5. SOA Implementation Case Study
6. Conclusion

ESB Products Evaluation

<table>
<thead>
<tr>
<th>Key: 1 = lowest, 5 = highest</th>
<th>BEA AquaLogic Service Bus 2.1</th>
<th>IBM Websphere ESB 6.1</th>
<th>Oracle SOA Suite</th>
<th>TIBCO Businessworks 5.1</th>
<th>Cape Clear Software ESB 6.5</th>
<th>Software AG Enterprise Service Integrator 2.1</th>
<th>Progress Sonic Software SOA Suite 6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication (20%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Transformations/Mapping</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Transport Protocol Support</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Messaging Style</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Addressing</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Integration (25%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web services</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Adapter support</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Integration Styles</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Protocol transformation</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Interaction (15%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Interface Definition</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Service messaging model</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Substitution of service implementation</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

0 – 1 Unacceptable         2 – 3 Potentially acceptable (partially meets requirement)         4 – 5 Acceptable (fully meets requirement)
### ESB Products Evaluation

(continued)

<table>
<thead>
<tr>
<th></th>
<th>Oracle BDA+ Aquatix</th>
<th>IBM WebSphere ESB 6.1</th>
<th>Oracle SOA Suite 10g</th>
<th>TIBCO BusinessWorks 5.3</th>
<th>Cape Clear Software ESB 1.5</th>
<th>Software AG Composition/Service Integrator 2.1</th>
<th>Progress Sonic Software SOA Suite 8.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management (20%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Administration and Monitoring</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Configuration</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Quality of Service (QoS)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Other Capabilities (20%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Auditing</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Service registry/repository</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Design-time environment</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Deployment</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Namespace directory</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Orchestration</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Total Score</td>
<td>4.19</td>
<td>3.69</td>
<td>3.97</td>
<td>3.77</td>
<td>3.74</td>
<td>3.34</td>
<td>3.99</td>
</tr>
</tbody>
</table>

0 – 1 Unacceptable  2 – 3 Potentially acceptable (partially meets requirement)  4 – 5 Acceptable (fully meets requirement)

### Sample SOA Architecture and Microsoft Product Mapping

- **Business Intelligence**: SQL Server 2007 BI, PerformancePoint Server
- **Portal / Web**: Microsoft Office SharePoint Server (MOSS) 2007, ASP.NET AJAX, Silverlight
- **Legacy Connectivity**: Host Integration Server, Biztalk Adapters
- **Security**: Microsoft Forefront, Microsoft ISA (Internet Security and Acceleration Server), Identity Integration Server, Windows CardSpace
- **Governance/Registry**: Partner with Amberpoint, Systinet
- **Modeling**: Visual Studio Team Architect, Biztalk Orchestration Designer, Workflow Designer, Sharepoint Designer
- **IDE**: Visual Studio Team System 2005
- **Management**: Microsoft Systems Center (multiple products)
- **Monitoring**: Biztalk Server 2006 BAM
IBM SOA Life Cycle

Websphere Integration Developer
- Discover
- Design, construct, and test
- Compose

Websphere Process Server
- Integrate people
- Integrate processes
- Manage and integrate information
- Run and manage complex applications

Websphere Business Modeler
- Gather requirements
- Model, design & simulate
- Define business metrics

Websphere Business Monitor
- Monitor business metrics and runtime
- Dashboards, scorecards

IBM SOA Product Stack

SOA Reference Architecture: Supporting Products
IBM WebSphere Process Server Architecture

Service Components
- Business Processes
- Human Tasks
- Business State Machines
- Business Rules

Supporting Services
- Interface Maps
- Selectors
- Business Object Maps
- Relationships

SOA Core
- Service Component Architecture
- Business Objects
- Common Event Infrastructure

WebSphere Application Server (J2EE Runtime)

Service Component Architecture

Component
- Interface (I)
- Reference (R)
- Implementation

Implementation Types
- Java
- WSDL Port Type
- BPEL
- State Machine
- Business Rules
- Human Task
- Selector
- Interface Maps
## ESB Typical Focus Areas

- **Communication Services**
  - Protocol
    - HTTP
  - Message Format
    - SOAP
    - EDI (X12)
  - Messaging Style
    - Publish/Subscribe
  - Synchronicity
    - Asynchronous

- **Integration Services**
  - Enterprise Routing
  - Transactional Services
  - EDI message parsing and transformation

- **Infrastructure Services**
  - Transaction Tracking & Problem Resolution
  - Security
  - Centralized Service Configuration Management

### ESB Mediation

![ESB Mediation Diagram]

- **Mediation Module**
- **Mediation Flow Component**
  - Request Mediation Flow
  - Response Mediation Flow

- **Reference A**
- **Reference B**
- **Service A Import**
- **Service B Import**

**Interface**

**Mediation Export**
<table>
<thead>
<tr>
<th>Tools</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Business Modeler</td>
<td>- Support for BPEL, WSDL, XSD, FDL and ADF</td>
</tr>
<tr>
<td></td>
<td>- Basic, intermediate and advanced user modes</td>
</tr>
<tr>
<td></td>
<td>- Operational, BPEL, FDL technology modes</td>
</tr>
<tr>
<td></td>
<td>- RUP Plug-in for SOA</td>
</tr>
<tr>
<td></td>
<td>- Multi Dimension Model, Static and Dynamic Analysis</td>
</tr>
<tr>
<td></td>
<td>- Business Measures Editor (metrics, KPI, aggregation etc)</td>
</tr>
<tr>
<td></td>
<td>- Integrated with WBM, WPS, WID, RSA, MQ Workflow</td>
</tr>
<tr>
<td>Rational Software Architect</td>
<td>- UML 2.0 Profiles and Patterns for service and message design</td>
</tr>
<tr>
<td></td>
<td>- WS-I compliant web services</td>
</tr>
<tr>
<td></td>
<td>- RUP Plug-in for SOA</td>
</tr>
<tr>
<td></td>
<td>- Asset explorer for reusable services, Support RAS export/import, RAS repositories</td>
</tr>
<tr>
<td></td>
<td>- Includes RAD, RWD, RSM functionalities</td>
</tr>
<tr>
<td></td>
<td>- C and C++ development support</td>
</tr>
<tr>
<td>WebSphere Integration Developer</td>
<td>- Support for SCA, BPEL, Human Tasks, Business Rules</td>
</tr>
<tr>
<td></td>
<td>- Support for WBI Adapters, JCA Adapters</td>
</tr>
<tr>
<td></td>
<td>- Visual Snippet Editors</td>
</tr>
<tr>
<td></td>
<td>- Includes RAD, RWD functionalities</td>
</tr>
<tr>
<td></td>
<td>- Integrated Test environment (component/Module Level)</td>
</tr>
<tr>
<td></td>
<td>- Support for WSIF, UDDI, SAAJ, WSDL</td>
</tr>
<tr>
<td></td>
<td>- Support for XML based RPX(JAX-RPC), XML Signature, XML-Encryption</td>
</tr>
<tr>
<td></td>
<td>- Support for private UDDI-registry</td>
</tr>
<tr>
<td></td>
<td>- Support for Web Services gateway</td>
</tr>
<tr>
<td>WebSphere Process Server</td>
<td>- Merger of WebSphere ICS, WebSphere MQ Workflow, WebSphere Business Integration server</td>
</tr>
<tr>
<td></td>
<td>- Support for Service component Architecture, SDO and CEI</td>
</tr>
<tr>
<td></td>
<td>- Support for BPEL, Human Tasks, Business State Machine and Business Rules</td>
</tr>
<tr>
<td></td>
<td>- Integrates with WebSphere Integration developer</td>
</tr>
<tr>
<td></td>
<td>- Support for Common Event infrastructure for monitoring, auditing, tracking of business processes</td>
</tr>
<tr>
<td></td>
<td>- Support for mediations to intercept or modify the messages between two endpoints</td>
</tr>
<tr>
<td></td>
<td>- Uses configuration, administration, clustering and high availability capabilities of WebSphere Application Server ND</td>
</tr>
</tbody>
</table>
### Essential Tools Features

**Tools** | Features
---|---
WebSphere ESB | - Support three levels of interactions viz. Standards based, interaction model based, mediation based  
  - Pre-built mediation functions such as message transformation, message logging, message routing and database lookup  
  - Integrates with Tivoli access manager, Tivoli directory and Tivoli composite application manager  
  - Can be integrated with existing MQ messaging solution  
  - Message clients for C++ and .NET enable non-java applications to connect to ESB  
  - Supports Web Services standards such as WS-security and WS-Atomic  
  - Support for UDDI 3.0 service registry to publish and manage service end-points  
  - Support for JMS, SOAP over HTTP(S), SOAP over JMS

WebSphere Message Broker (Advanced ESB) | - Transformation of messages viz. Web Services, other XML and non-XML formats  
  - Route messages based on business rules to match information content and business processes  
  - Support for WSDL, SOAP, JMS and HTTP  
  - Integrated WebSphere MQ transports - for Enterprise, Mobile, Real-Time, Multicast and Telemetry end points  
  - Provide native JMS interoperability, acting as a bridge between any combination of different JMS providers  
  - Pre-built message definitions, WSDL import, MRM, XML parsers and optimized ESQL functions  
  - Extend an existing WebSphere MQ and WebSphere Event Broker infrastructure  
  - Eclipse based Message Broker Toolkit, Standard based XML schema, Visual debugging of flows, ESQL, Java, and graphical mappings

WebSphere Adapter | - JCA 1.5 specification, EMD specification for wizard-driven configuration and service data objects (SDO) specification for exchanging data  
  - Integration with multiple products such as WebSphere Process Server, WebSphere Message Broker, WebSphere Interchange Server, WebSphere Business Integration Server foundation and WebSphere information integrator  
  - Pre-built adapters for SAP, CRM, PeopleSoft etc and for XML, Web Services, HTTP, TCP-IP, FIX, SWIFT etc  
  - WebSphere Adapter Toolkit – to develop custom JCA adapters  
  - WebSphere Business Integration Adapter Development kit – to develop custom adapters based on WebSphere business integration framework (.J2SE Adapters)

Tivoli Composite Application Manager for SOA | - Includes Web Services Navigator, Plug-in to Rational  
  - Visualization of Web Service end-to-end flows  
  - Support for identifying and isolating performance problems  
  - Support for alerts when performance is degraded  
  - Report results against committed service levels  
  - Support for automatic service mediation
### Essential Tools Features (continued)

<table>
<thead>
<tr>
<th>Tools</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Business Monitor</td>
<td>- Support for Common Event infrastructure for receiving and emitting events implemented by WebSphere Process Server</td>
</tr>
<tr>
<td></td>
<td>- Integrated with WebSphere Business Modeler and WebSphere Process Server</td>
</tr>
<tr>
<td></td>
<td>- Support for business dashboards</td>
</tr>
<tr>
<td></td>
<td>- Tracking KPIs against goals chosen to monitor</td>
</tr>
</tbody>
</table>

### SOA Activities – Tool Support

#### SOA Life Cycle – Requirements and Business Process Analysis

- **Rational Portfolio Manager**
  - Assess risks and costs to prioritize SOA projects
  - Identify and manage projects and resources
  - Govern execution and costs

- **Rational Requisite Pro**
  - Document Business Strategy
  - Capture business goals, objectives, and requirements
  - Track business objectives, requirements, and create Business Use case outlines

#### WebSphere Business Modeler

- Model As Is business processes
  - Create Business Item
  - Create KPIs
  - Static Analysis
  - Export as BPEL/UML
- Model To Be business processes
  - Create Tasks
  - Assign Roles
  - Dynamic Analysis
  - Optimize BP Model
SOA Governance

- **SOA governance**
  - IBM SOA Governance Life cycle - Recommended strategy to develop, deploy, maintain and enhance effective SOA governance
  - IBM SOA Governance and Management method – includes detailed tasks and recommendations for each of SOA Governance life cycle.

- **Product supported in SOA Governance**
  - plug-in for the IBM Rational® Method Composer®, Rational Portfolio Manager
  - Rational Software Architect, Rational Data Architect, RUP for SOA
  - Tivoli federated identity manager, Tivoli access manager
  - Rational Clear case, Rational Clear quest, Tivoli Change & configuration
  - Tivoli composite application manager for SOA, WebSphere Business Monitor, Tivoli Service level Advisor
  - WebSphere Service Registry

---

SOA Skills Profile

<table>
<thead>
<tr>
<th>Role</th>
<th>Tools</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA Architect</td>
<td>WebSphere Business Modeler (Advanced), Rational Software Architect</td>
<td>UML Modeling, Service Oriented Analysis &amp; Design</td>
</tr>
<tr>
<td></td>
<td>Rational Software Architect / Rational Software Modeler</td>
<td>Service Modeling, Modeling, Service Oriented Analysis &amp; Design</td>
</tr>
<tr>
<td>Technology Specialist</td>
<td>Rational Application Developer</td>
<td>Development of Service Interface, Development of Service implementation, Adaptor development</td>
</tr>
<tr>
<td>SOA Test Engineer</td>
<td>Rational Functional Tester, Rational Performance Tester</td>
<td>Preparation of Functional Test plan, test cases, Performance test plan, test case and execution of test cases</td>
</tr>
<tr>
<td>SOA Integration Engineer</td>
<td>WebSphere Integration developer, WebSphere Application Server, WebSphere Process Server, WebSphere ESB</td>
<td>Service Assembly and Business Process integration, Service deployment</td>
</tr>
<tr>
<td>IT Manager</td>
<td>WebSphere Business Monitor, Tivoli Composite Application Manager</td>
<td>Service monitoring</td>
</tr>
<tr>
<td>SOA Program Manager</td>
<td>Rational Portfolio manager, Rational Requisite pro</td>
<td>Tracking Service Level financials, SOA team dependencies, Managing Resources</td>
</tr>
</tbody>
</table>
### Scenario-Based Recommended Tools Stack

<table>
<thead>
<tr>
<th>Key Requirements</th>
<th>Development Tools</th>
<th>Runtime Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Services, JMS services and MQ services with basic routing and transformation</td>
<td>- Rational Software Architect</td>
<td>- WebSphere Application Server 6.0 (Service Integration Bus)</td>
</tr>
<tr>
<td></td>
<td>- Rational Application developer</td>
<td></td>
</tr>
<tr>
<td>Web Services integration with advanced routing and transformation like filters,</td>
<td>- Rational software Architect</td>
<td>- WebSphere Application Server ND</td>
</tr>
<tr>
<td>mediators etc</td>
<td>- Rational Application developer</td>
<td>- WebSphere ESB</td>
</tr>
<tr>
<td>Business Process Modeling and orchestration that involves human, system, business</td>
<td>- WebSphere Business Modeler</td>
<td>- WebSphere Application Server ND</td>
</tr>
<tr>
<td>rules, state machines etc</td>
<td>- Rational software Architect</td>
<td>- WebSphere ESB</td>
</tr>
<tr>
<td></td>
<td>- WebSphere Integration Developer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Rational Application developer</td>
<td></td>
</tr>
<tr>
<td>Enterprise application integration that includes ERP, SAP, Legacy etc</td>
<td>- Rational software Architect</td>
<td>- WebSphere Application Server ND</td>
</tr>
<tr>
<td></td>
<td>- Rational Application developer</td>
<td>- WebSphere JCA Adapters</td>
</tr>
<tr>
<td></td>
<td>- WBI Adapters</td>
<td></td>
</tr>
<tr>
<td>Advanced ESB capabilities like any-to-any transformation, packaged application</td>
<td>- Rational software Architect</td>
<td>- WebSphere Application Server ND</td>
</tr>
<tr>
<td>integration, Legacy application integration</td>
<td>- Rational Application developer</td>
<td>- WebSphere JCA Adapters</td>
</tr>
<tr>
<td></td>
<td>- WebSphere Message Broker development tool kit</td>
<td>- WBI Adapters</td>
</tr>
<tr>
<td></td>
<td>- WebSphere Message Broker</td>
<td></td>
</tr>
<tr>
<td>Business Performance through Activity Monitoring</td>
<td>- Business Monitor Tool kit</td>
<td>- WebSphere Application Server ND</td>
</tr>
<tr>
<td></td>
<td>- Rational Software Architect Asset Explorer using RAS</td>
<td>- WebSphere Business Monitor</td>
</tr>
<tr>
<td>Dynamic Service Discovery</td>
<td>- WebSphere Registry and Repository</td>
<td></td>
</tr>
</tbody>
</table>

### Recommended Flow of Activities

- The sequence of development is recommended to be in the following order to ensure maximum output
  - Service Identification
    - Business Process Driven – Top Down
    - Use case Driven – Top Down
    - Existing Applications – Bottom Up
  - Message Design – (Service design needs messages to set as parameter and return types for operations. Hence complete Message Model is a prerequisite for Service Design model.)
  - Service Design using Rational software architect
    - Service Specification
    - Service interaction
    - Service granularity
    - Service invocation
  - Business Process Orchestration using WID
  - Generate Web Services stubs and skeletons using RAD / RSA
  - Implement web services using Java, EJB, Message etc using RAD
  - Orchestrate business processes using service implementation in WID.
  - Deploy Application Services into WebSphere Application Server
  - Deploy Mediation services into WebSphere ESB
  - Deploy Business Process services into WebSphere Process Server
  - Test the services
Case Study - Objectives

**Background**
- Large Credit card transactions processing enterprise
- Operations spread across multiple countries in Australia, Europe and America
- Initiated a strategic program to understand the value of SOA to have agile and cost-effective transaction processing system for the enterprise

**Objectives**
- Develop a business agile, reusable and cost-effective Exception Management that adheres to SOA paradigm and determine the feasibility of SOA as a future state Reference Architecture for Customer
- Business Agility - Agile Business environment to introduce new services or products quickly
- Integration - ESB adaptation as Messaging layer to integrate disparate applications and resources
- Centralized Exception management to remove redundancy, duplication and integration issues
- Common message standards to transport application data across the applications
- Measure SOA overhead in terms of Performance
Case Study – Problems Statement

- **Business**
  - Decentralized Exception management processing systems
  - Non-compliance to industry standards
  - Lack of flexibility for the addition of new functionality and adaptability of emerging regulatory changes
  - Low Straight Through Processing (STP) Rates
  - Lack of efficient business activity monitoring and business performance management
  - Lack of an efficient mechanism to define and manage SLAs
  - Lack of flexibility and increase in redundancy for adding new functionality in the Customer enterprise applications.

- **Technical**
  - Integration of multiple systems that are acquired to support various business models
  - Disparate and Redundant sub-systems leading to complex application environment
  - Closely coupled, point-to-point integrated applications
  - Lack of reusability across different applications
  - Higher Development time and effort due to various integration issues

Case Study – Business and Technical Drivers

- **Business Drivers**
  - Centralization of Exception management
  - Avoidance of duplication of common functionality and increased re-use
  - Rationalization of Systems and Interfaces
  - Introduction of new Services and Products at near zero-latency
  - Growth of business enabled through the higher Straight through Processing rates

- **IT Drivers**
  - Loosely coupled Infrastructure
  - Services shared across various channels
  - Conformance to Open Standards Architecture and Industry standards
  - Flexible architecture to enhance and modify Auto correction parameters
  - Easier adoption of mandatory changes
Case Study – Functional and Non-Functional Requirements

**Functional**
- Single Exception Management to handle all the exceptions across multiple systems
- Improve business performance by processing at higher STP rates
- Provide flexibility and capabilities to handle new business exceptions
- Ability to auto correct the exceptions with business rules
- Parameterized exception correction parameters

**Non-Functional**
- System Availability – Exception management system should be 24*7 available
- Capacity and Throughput – Should scale well above the current peak load of 50000 transactions / day
- Failover – Support seamless failover in the event of any failure
- Business Continuity – Disaster Recovery to reinstate services within 15 minutes
- Scalability – 10% increased usage every year
- Performance - Response time of 3 seconds for retrieval and 5 seconds for update of data
- 24 hours SLA for correcting the exceptions which includes both manual and auto corrected exceptions.

---

**Case Study – Activity Diagram**
Case Study – Business Process Model

- **Receive Exception**
  - Batch Process which converts records in to canonical format

- **Manage Exception**
  - **Process Exception**
    - Initial Processing of Record received from Source Systems
  - **Manual Correction**
    - Includes determining if an approval is needed based on business rules and validation
  - **Manual Approval**
    - Allows for approval and rejection of manual edits when required

- **Reprocess Exception**
  - Batch Process which “pulls” corrected records for reprocessing

- **Housekeeping**
  - **Purge Exception**
    - Deletes already processed records
  - **Auto Delete**
    - Delete records which match rules defined by client/source systems
### Case Study – Service Catalog

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Service Category</th>
<th>Service Component</th>
<th>Service Operations</th>
<th>Input Message</th>
<th>Output Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignApprovalGroup_AI</td>
<td>Business App Serv</td>
<td>BPCFramework</td>
<td>createGroupTask()</td>
<td>ExceptionDetails</td>
<td>ExceptionDetails</td>
<td>Created as the Manual Approval Sub-Process to assign an Approval Group for the Exception to be approved.</td>
</tr>
</tbody>
</table>

This Service is invoked in the Manual Approval Sub-Process to assign an Approval Group for the Exception to be approved.

### Case Study – Message Design

```
<Message>
    <ApprovalGroupDetails>
        <GroupName : String>
        <GroupId : String>
    </ApprovalGroupDetails>
</Message>

<Message>
    <CorrelationDetails>
        <STCOrrelationID : String>
    </CorrelationDetails>
</Message>

<Message>
    <AuditDetails>
        <STMAmodified : String>
        <STMAmodifiedDateTime : String>
    </AuditDetails>
</Message>

<Message>
    <HumanTaskDetails>
        <STCAcrookedBy : String>
        <STCAcrookedDateTime : String>
    </HumanTaskDetails>
</Message>
```
Case Study – Deliverables

- **Service Design**
  - Service Catalog
  - Service Specifications
  - Service interaction diagrams
  - WSDLs- Service interface
  - Message Design
  - Message Schema (XSDs)

- **Business Process Orchestration**
  - Optimized Business Process Model
  - Process Modules - BPEL
  - Mediation Modules – (Java / EJB)
  - Composite Services

- **Component Design**
  - Class diagram
  - Sequence Diagrams
  - Screen Specifications
  - Data Model

- **Service Development**
  - Stubs and Skeletons
  - Service Message Objects
  - Service Data Objects
  - Interfaces

- **Business Process Development**
  - Process Modules - BPEL
  - Mediation Modules – (Java / EJB)
  - Composite Services

- **Component Development**
  - POJO
  - EJB
  - Custom Adapters

---

**Agenda**

1. Introducing SOA
2. SOA Patterns
3. SOA Methodologies
4. Mainstream SOA Implementations
5. SOA Implementation Case Study
6. Conclusion
Summary

- SOA is an Enterprise reference style used to enable business agility
- Managing a service-driven Enterprise architecture requires several architectural patterns beyond SOA known as SOM, SOI, and SOP
- Managing the service-lifecycle, the adoption of SOA, SOA governance, and other SOA-related aspects requires SOA-specific methodologies
- Various mainstream implementation of SOA are available as ESB products by IBM, Microsoft, Oracle, RedHat, etc.

Class Project

- Project Description
  - The project focus is two-fold:
    - Based on a framework-based enterprise application of your choice, you will implement common facilities and application-level services on top of various types of application server technologies to support the various aspects of your chosen application
    - As you transition from one platform to another you will study and develop model-based migration and interoperability tools that leverage off of the cutting-edge concepts subsumed by modern Model Driven Architectures (MDAs)
Readings

- Readings
  - Handouts posted on the course web site
  - Explore mainstream SOA offerings
    - Microsoft, IBM, Oracle, RedHat, etc.
  - Read related white papers/documentation on SOA environments

Assignments

- Assignment:
  - #4a: Investigate mainstream SOA implementations development environments. Write a short report that documents your findings and recommendations with respect to selection criteria in support of development environments for SOA/ESB technologies covered in this session
  - #4b: See homework #4 specification
Project Related Frameworks

- Project Frameworks Setup (ongoing)
  - Apache Web Server (www.apache.org)
  - Perl (www.perl.com)
  - Microsoft IIS with COM+/Net and ASP
  - Apache Tomcat
  - Adobe JRun4
  - Apache Cocoon 2/XSP
  - Visibroker, Orbacus
  - RMI-IIOP
  - Oracle WebLogic Suite 11g, IBM WebSphere V7, JBoss 3.x
  - ESB add-ons to select application server products

Next Session: Business Process Management Servers