EA Meets SOA

Best Practices and Winning Strategies

Paul Allen
Principal Consultant
Agenda

- SOA Vision and Concepts
- EA Meets SOA
- Pragmatic Progression
SOA Concepts and Vision

- Services Defined
- SOA as a Business Phenomenon
- Enabling Strategies
Service Defined

- A Service is a capability by which the need of the Service Consumer or Requestor is satisfied according to a contract
- Separate the what from the how, who and where
- Manage and govern through policies and contracts
- Communicate using messages that share schema not technology
- Are discoverable, at design and run-time
What Does This Enable?
SOA Vision I - Structural Improvement and Portfolio Rationalization

- ERP X
- APP N
- ERP A
- PROCESS Y

Reducing dependency horizon and scope of release modules

Service

Consolidation/Selection process

Consistent policies/rules
Rationalize process

Single Service encapsulating implementation complexity
Aggregating Information from multiple sources

- ERP
- CRM System 2
- CRM System 1
- Product-Based System

e.g. Multiple Sources of Customer Details

e.g. Single Customer Details Service

Standardizing capabilities across disparate components and processes

Rationalization behind Service façade
Consolidation/Commoditization/Addition

Consistent policies/rules
Rationalize process

Reducing dependency horizon and scope of release modules

e.g. Single Customer Details Service

Standardizing capabilities across disparate components and processes

Rationalization behind Service façade
Consolidation/Commoditization/Addition
What Does This Enable?
SOA Vision II – Flexible, Federated Real-time Business Processes

Enabling a virtual federation of participants to collaborate in an end-to-end business process

Enabling reuse of Services in different solutions

Enabling aggregation from multiple providers or flexible choice of provider

Enabling flexible use of resources including existing applications

Ordering

Payment

Ticket Sales

Ticket Collection

Inventory

Availability

Logistics

Manufacturing

Resource

Resource

Resource
SOA is more than infrastructure!

- Collection of knowledge & best practice
- Coherent conceptual approach
- Blueprint for enterprise
- Reference Model and Architecture
- Processes
- Rigorous – where necessary
- Standards based

**SAE = Service Architecture and Engineering**
SOA Principles

**Consuming Solutions**

- **Loose Coupled**
  - Contract based service specification encapsulates underlying resource – enables articulation

- **Standardized**
  - Single instance of enterprise capability delivers consistency of behaviour and reuse

- **Abstracted**
  - Generalized service provides inherent business flexibility

- **Modular**
  - Service forms basis for reduction in dependency and componentization

- **Composable**
  - Fractal architecture allows solution flexibility using alternative, specialized and or orchestrated services

- **Enterprise policy governs design & use**

- **Virtualized**
  - Encapsulated resource enables flexibility of supply and resource

**Functional Capabilities/Resources**
Full Life Cycle SOA – Three Levels of Abstraction

- Service concepts are equally applicable to the way both business and IT each thinks about the provision and consumption of capability and resources.
- SOA is as much of a business modeling approach as it is a software engineering paradigm.
- Services can represent meaningful business capability.
  - Recognizable by the business.
  - Enable Business/IT convergence.
- Run-time Services provide a further layer of flexibility over the software architecture.
  - E.g. Many Run-time Service instances of the same Software Service – resolved by dynamic routing.
EA Meets SOA

- Coverage
- Practical Techniques
- Example
EA meets SOA: Coverage

Enterprise Architecture:
“The who, what, why, when, where, and how of the business at every level from high-level corporate goals to the code of low-level programs that implement business processes used to achieve those goals.” *

- Blueprint “As Is” versus “To Be”
- Wide application coverage
- Wide data scope
- Technologies and topologies

SOA:
The discipline that enables software to be provided and consumed as services.

- Core versus Context
- Separation of Consumers and Providers
- Focus on Quality of Service
- Criticality of Specification and Policy

* Source: “TOGAF: Establishing Itself As the Definitive Method for Building Enterprise Architectures in the Commercial World”,
  -David Harrison and Lou Varveris
  http://www.developer.com/design/article.php/3374171
EA meets SOA: Techniques

**EA Concepts**

**Example**

Service-Oriented Techniques

Service-Oriented Viewpoints

Process Services

Rationalization Requirements,
Core Context Analysis

Domain Partitioning,
Service Identification

**Business Architecture**

**Application Architecture**

**Data Architecture**
SOA for Application Rationalization

**EA Concepts**

- Business Architecture
- Application Architecture
- Data Architecture

**SOA Concepts**

- Business Domain
- Services Grouped by Domain
- Services Selected for Sharing, Aggregation or Differentiation
- Services Organized into layers by Purpose and Type
How do you connect J2EE to .NET?

How do you connect SAP to Siebel?

How do you connect sales to customers?

Application and Data Layers
- Applications, Databases
- How do you connect SAP to Siebel?

Business Process Layer
- Cross Functional End-to-end Sales Order Process

Service Layer
- How do you connect sales to customers?

Technology Layer
- Hardware, Network
- How do you connect J2EE to .NET?
Example Logical Data Model

- Subcontractor
  - Warehouse Position
  - Vehicle
- Warehouse
- Journey
- Customer
  - Invoice
  - Item
  - Payment
- Account

*contains*
*visited by*
*current holds*
*is base for*
*current responsible for*
*current holds*
*current stores*
*requests*
*results in*
*contains*
*owns*
*records*
### Example Business Service Architecture

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>PROCESS</th>
<th>CAPABILITY</th>
<th>CORE BUSINESS</th>
<th>UTILITY</th>
<th>UNDERLYING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parcels System</td>
<td>Delivery Process</td>
<td>Shipping</td>
<td>Invoices</td>
<td>Address Formatter</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Finance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Customers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accounts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example Business Service Architecture Diagram:**

- **APPLICATION:** Parcels System
- **PROCESS:** Delivery Process
- **CAPABILITY:** Shipping, Finance
- **CORE BUSINESS:** Invoices, Subcontractors, Shipments, Warehouses, Customers, Accounts
- **UTILITY:** Address Formatter
- **UNDERLYING:**
Pragmatic Progression

- Adoption Roadmap
- Practical Experiences
- The Way Forward
Current State

- Tactical SOA
  - Project and or infrastructure driven
  - Delivering and using services with little or no structure
  - Little or no consensus or consistency across the organization
  - No explicit policies and repeatable processes that permit governance
  - Recipe for **Service Anarchy** and limited ROI
Introduction to SAE

- Tactical SOA
  - Project and or infrastructure driven
  - Delivering and using services with little or no structure
  - Little or no consensus or consistency across the organization
  - No explicit policies and repeatable processes that permit governance
  - Recipe for Service Anarchy and limited ROI

- Strategic SOA
  - Analogous to a manufacturing and assembly environment
  - Classes of service have appropriate, repeatable processes
  - Mature architecture and engineering processes and practices
  - Formality over policy implementation to ensure implementation of both business requirements and architectural policies
  - Manage outcomes that are fit for purpose, deliver future flexibility, utility and cost.

Service Architecture & Engineering:
A comprehensive, defined approach for service architecture together with repeatable service engineering processes that guide the delivery of the agile enterprise.
Everware-CBDI SOA Capability Maturity Model

- **Cultural Integration**: Service concept embedded in mature enterprise practices
  - **Reengineering**: Shared service platform allows wider collaboration and reengineering of IT and business
  - **Integration**: Repeatable processes deployed to create basic shared services capabilities
  - **Early Learning**: Initial exploratory SOA activity

A Maturity Model is a logical analysis of capabilities and their dependencies

- Management
- Architecture
- Infrastructure
- Process
- Organization
- Projects
Structural Improvement and Portfolio Rationalization
Case Study - Telco

- Aggressive SOA strategy to respond rapidly with new products and services in rapidly evolving telecoms market
- Aim to rationalize internal systems environment to achieve competitive advantage through cost leadership
- Had unique stacks in each country supporting approx 28,000 Global Services products
- SOA objectives to create one coherent environment and reduce no of products to 18; consistent set of processes and systems - significantly reduce cost of maintenance and time to market
- Developed canonical data model – defining core business data - Matrix Architecture
- Enterprise wide rollout of a service oriented architecture with the goal of rationalising some 5000 systems down to approximately 100, and to provide a common set of re-usable components
- Another goal is to reduce the time to bring products to market by two thirds - grouping systems into logical domains which separate and expose generalized services
- Recently centralized all its Line Of Business integration competency centres into a single pan-organizational unit targeted towards the delivery of services.
- Widespread acceptance of the core capability architecture is emerging with discussions on how to organise their IT operations around delivering, utilizing and supporting capabilities
- Centralized integration group initially set up with 400 staff (supporting 14 different integration platforms). Today reduced to < 100 key roles
- SOA pattern:
  - Common Component Service: Standardize & rationalize
  - Centralized integration
Structural Improvement and Portfolio Rationalization
High Level Roadmap

Early Learning

Wrapped Existing Application

LOB 1

Integration

ULOB 1

Retail & Wholesale POS

Reengineering

ULOB 2

UPOS 2
New Book by Paul Allen

ISBN: 0-521-843367
Independent Guidance for Service Architecture and Engineering

www.cbdiforum.com

www.everware-cbdi.com