Service Oriented Architecture and Design Strategies

Michael Rosen
Director, Enterprise Architecture
Cutter Consortium
mrosen@cutter.com
Mike Rosen

Consultant

- IT Architecture and Strategy
- Chief Enterprise Architect for service- and component-based systems
  - Finance, Insurance, Telecom
- SOA, EA and MDA implementation, strategy and training
- 25+ years experience in distributed systems, software and architecture

Cutter Consortium – Director of Enterprise Architecture

SOA Institute – Editorial Director

Author

- Cutter Consortium
  - ‘10 Things and Architect Does to Add Value’
  - ‘EA by Example’
  - “Designing Service Oriented Applications”
  - “EA – It’s not Just for IT Anymore”
  - “Agile Methods and Enterprise Architecture”
  - “Enterprise Architecture Roll-out and Training”
  - “Service Oriented Integration: Aligning SOA with Enterprise Integration”
  - “Implementing SOA on Common Technologies”

- Books
  - Integrating CORBA and COM Applications, 1998, Wiley
Agenda

- Problem
- SOA Solution
- Hype and Reality

- Challenges
  - Semantics
  - ownership

- Standards to the Rescue

- Conclusion
A common view of Healthcare Integration
But this was not sustainable…
The Result: Enterprise Application “Spaghetti”

Source: Gartner, 2000
The Result: Typical IT Budget Allocation

- Maintenance: 70-90%
- Integration: 15-30%
- New Applications: 5-15%
SOA: A Better Solution

Channels

Patient Management  Marketing  Diagnostics

Business Service Bus

Patient Record  Billing Service  Laboratory Service  Other Services

Integration Service Bus

Pharmacy System  Lab 1 System  Patient 1 System  Billing System  Lab 2 System  Patient 2 System

Data  Data  Data  Data  Data  Data
SOA Context in Healthcare

Source: Practical Guide to SOA in Healthcare
Healthcare Context Boundaries

- The **Inter-organizational Boundary** (outermost) represents inter-organizational considerations, such as policies, sharing agreements, and business partners.

- The **System Boundary** represents the physical platforms on which software and systems run, including servers, networks, and so on.

- The **Application Boundary** represents the software running on those platforms, inclusive of applications and data.

- The **Business Process / Orchestration Boundary** manages the intersection between software and workflow, and would manage coordination among multiple software components that all must interact to satisfy business needs.

- The **Service Implementation Boundary** depicts the implementations themselves, interacting across a service bus, and realizing the architecture.
SOA History

■ Service Oriented Architecture (SOA) is NOT new!

■ Many Successful SOA Applications have been built in the past:
  • CORBA (Wells Fargo, Credit Suisse)
  • Tuxedo

■ Many, many more attempts at SOA failed

■ But, we can learn from what failed, and what succeeded
SOA, Who Cares?

- Built on SOA, originally for Customer Service Representatives
- ...Expanded to 80 Lines of Business
- Agile / Flexible Industry leading functionality

Service Layer
SOA Solution for Unified Customer Service

- Customer service
- Internet banking

**Presentation**
- Call Center Services
- LOB’s (Wealth Mgmt)
- Internet Services

**Logic**
- Customer Centric Architecture Services
  - Check Account System
  - Save Account System
  - Account x System

**Operational Systems**
SOA is Hard!

- Previous technical infrastructures were very difficult to master
- We did not adequately understand the characteristics of services and service design
- Requires an understanding of the business and information and a strategic vision
- Requires an architectural based approach
- Requires an appropriate methodology
- Requires a supporting organizational structure
Enterprise SOA

Business Model

Enterprise Business Process

Service

Web Service

Web Service

SOAP Service Bus'

Processes, Guidelines, Tools

Common Semantics and Data

Web Service

Application Service Adapter

Defines tools, processes and technology for combining services into EBP

Specifies Definition and requirements of a service

Defines communications technology for application integration

Defines common semantics and data

Specifies service wrapping techniques

© Michael Rosen 2009
SOA Definition

- “In a nutshell, SOA provides an approach for business transformation based on dividing complex environments into well defined, formally specified functions based on the activities they perform (services).

- Each service has well defined responsibilities and authority.

- These services then work together in collaboration to support the workflow of the business, all within the context of governance and oversight that manages their coordination and performance.”

  • Practical Guide to SOA in Healthcare – OMG & HL7
A Definition of SOA

- SOA is concerned with the independent construction of services which can be combined to realize meaningful, higher level business processes within the context of the enterprise.

- A Service Oriented Architecture describes several aspects of services within an enterprise:
  - The granularity and types of services
  - How services are constructed
  - How the services communicate at a technical level
  - How the services are combined together (i.e. orchestrated)
  - How the services interoperate at a semantic level (i.e. how they share common meanings)
  - How services contribute to IT and Business Strategy
So what else is needed?

Common taxonomy or layering of types of services (e.g. process, core business, data access)

Common framework of supporting infrastructure services to manage the “…ilities”

Enumeration of meaningful, appropriate Services

Standards for Service interfaces, including agreed information and behavior semantics

Clarification of dependencies between services and relationship to key business processes
Types of Data

**Semantic Data:**
Described by common / shared information model. A view of the Common aspects of services Used for information exchange through interfaces.

**Domain Data:**
Described by internal data model. A view of the physical data. Used for implementation.

**Physical Data:**
Described by data base schema. Used for persistence.
So why standard Healthcare Service Specifications?

- Provide common architectural building blocks
  - Solve problems and create opportunities for developers/architects to improve healthcare with technology
  - For consumers (like KP) provides cheaper and faster integration
  - Enable inter-organization interaction over the internet using a common approach

- Tie good SOA practices and patterns to the rich models of HL7, CEN, OpenEHR

- Create true **Interoperability** specifications, not just **Integration** specifications

- Two important services
  - EIS – Entity Information Service
  - RLUS – Retrieve, Locate, and Update Service
EIS (Content Models)

An EIS instance contains:

- A Functional Profile – An Instance’s Supported Operations
- A Semantic Profile – the composition of semantic signifiers, e.g. HL7 RIM v2.14 Patient, OpenEHR Patient Archetype, HL7 V2.5 Patient, Provider, Device etc.
SOA Patient Information Solution

- Patient Registration
- Administration
- Diagnostics

Business Service Bus

- EIS
- RLUS
- Insurance
- Document History

Extended Integration

- Hospital 1 Patient
- Lab 1 System
- HMO Patient
- Specialist Patient
- Insurance Bureau
- Document System

© Michael Rosen 2009
Cross Domain EIS (XEIS - Hierarchic)

Source: OMG EIS Specification
As Simple As Possible…

…but not more so! (A. Einstein)

- Single system view
  - Enables consolidated view (read), but not data utility (CRUD)

- Single repository
  - Impractical. Data needs to be stored at the service, and then exposed and integrated into workflows

- Master Patient Index
  - Integrates data, but not workflows

- Big bang, analysis paralysis, uncoordinated efforts, not enough governance, too much governance, …
Summary

- SOA is a good solution for the challenges facing healthcare patient information
- Anyone can build a service…SOA is about making things work together to build higher level value
- This requires common understanding and semantics
- Use industry standards where they exist
- Accommodate organizational realities
- Adopt an incremental approach
- Have perseverance and patience
Thank You!

“Every complex problem has a solution that is clear, simple…and wrong”
— H.L. Mencken, 1949