Service Orientation & Service Oriented Architectures

Setting the stage for the semester
Duane Truex, Arun Aryal & Tianjie Deng
Cognitive Map of 8090

IS Architectures as Strategy

Work Systems Models
S. Alter, “Work System Method”

Level of Analysis
EA – Organizational
IS Architectural – Mid-Range
Work Systems / SOA – Low range

IS architectural components and development
Mid-Range Modeling

ERP as instantiation of Architectures

Architectural Modeling options
Agenda:

- Review from last week
- Strategy review
- SOA
  - Definition of SOA
  - Why SOA?
  - SOA Architecture
  - Traditional vs. SOA architecture
  - 8 steps towards SOA
- Integrating EA throughout organizations
  - SAP NetWeaver
- EA evolution & governance
Strategy Review: How Org’l Strategy Determines IS Structure

- Organizational strategy begins with an assessment of an industry
- That assessment leads to a competitive strategy which determines value chains
- Value chains lead to business processes
- Business processes determine information system structure, features, and functions
Strategy Review:
Porter’s Five Forces Model

Intenity of each force determines characteristics of the industry, how profitable it is, and how sustainable that profitability will be.
Firms engage in one of four competitive strategies:

- **Cost (Industry-wide)**: Lowest cost across the industry.
- **Cost (Focus)**: Lowest cost within an industry segment.
- **Differentiation (Industry-wide)**: Better product/service across the industry.
- **Differentiation (Focus)**: Better product/service within an industry segment.

To be effective, organization goals, objectives, culture, and activities must be consistent with organization strategy.
Strategy Review: How does IS Provide Competitive Advantage?

**Product Implementation**
1. Create a new product or service
2. Enhance existing products or services
3. Differentiate products or services

**Process Implementations**
4. Lock in customers and buyers
5. Lock in suppliers
6. Raise barriers to market entry
7. Establish alliances
8. Reduce costs

Information systems can help create a competitive advantage by being part of the product or by providing support to the product.
Architecture:

- Architecture implies a consistent and coherent design approach. Essential principles include:

- **Consistency:** The same challenges should be addressed in a uniform way.

- **Reliability:** The structures created must be fit to purpose and meet the demands for which they are designed.

- **Extensibility:** A design must provide a framework that can be expanded in ways both foreseen and unforeseen.

- **Scalability:** The implementation must be capable of being scaled to accommodate increasing load by adding hardware to the solution.
What are Services?

- Service is
  - component of distinctive functional meaning that typically encapsulate a high-level business concept
  - Lego block

- Service contains
  - Contract – message type def, constraint, description (comment)
  - Interface – set of operations
  - Implementation – Logic and data
Type of Services

Business Process
Composed of
Activities/Tasks/Service
Fulfilled by
Applications/Packages
Supported by
Infrastructure
Examples of a Service

- Creating a Purchase Order inside a mainframe application
- Requesting and reserving a room in a hotel
- Applying for a loan by filling out a loan request form
- Search books/music based on keywords
What Is A SOA?

Definition of SOA

**SOA is an architectural approach that allows to:**

- Expose enterprise data and business logic as loosely coupled, discoverable, structured, standards-based, coarse-grained, stateless units of functionality called services
- Choose a services provider and access to existing resources exposed as services
- Assemble new processes from existing services that are exposed at a desired granularity through well-defined, published and standards-compliant interfaces
- Share capabilities and reuse shared services across a network irrespective of underlying protocols or implementation technology
SOA Characteristics

- Based on open standards
- Foster inherent reusability
- Foster intrinsic interoperability
- Emphasizes extensibility
- Fundamentally autonomous
- Promotes dynamic discovery
- Promotes architectural composability
- Promotes loose coupling throughout the enterprise
- Supports incremental implementation
SOA Characteristics

- Services are platform independent, **self describing interfaces** (XML)
- Messages are formally defined
- Services can be discovered
- Services have **quality** of service characteristics defined in policies
- Services can be provided on any platform
- Can be governed
Potential Benefits of SOA

- Efficient and effective usage of ‘Business Services’
- Improved Integration, intrinsic interoperability
- Organizational agility
- Loosely-coupled with reusable assets and services
- Drives business processes closer to end users
- Leverage and integrate existing applications
- Provide standard connections between systems
- Abstract complexity for developers
Potential Benefits of SOA

- Feedback at different levels
- More efficient development process
- Adequate business infrastructure
- Cost savings
- Risk mitigation
- Evolutionary approach
- Independence from technology
- Reuse
Shift to SOA

Accidental
Rigid
Silo-Oriented

Layered
Extensible
Service-Oriented
Why SOA?

Distributed Data
Distributed Computation
Distributed users

Research & Development
Marketing
Accounting
Customer Service
Sales
Manufacturing

Distributed Business
Requires
Distributed Computing
Why SOA?

- Interoperation issues
  - Heterogeneous network protocols
  - Heterogeneous hardware platforms
  - Heterogeneous operating systems
  - Heterogeneous application formats
  - ......

- Increased Competition
- Enhancement of Business Capabilities
- There must be consensus On **Interoperability**
SOA architecture

Service Directory

Service Consumer

Finds and Retrieves

Service Provider

Registers

Invokes
# Traditional Architecture vs. Service Oriented Architecture

## Traditional Architecture

- Components are tightly coupled
- Interface between subsystems is explicitly defined in terms a stack of protocols
- Known implementation
- Components are not independent of implementation attributes
- Tends to be closed architecture –
  - Difficult to replace, or reuse components from one system to another
- Commonly, functions are accessible with the help of point-point connections over the network
- Tends to be confined to a single organization
- Based on standard set of layer – presentation, business, data access, Database

## Service Oriented Architecture

- Loose coupling by means of services with standardized interfaces
- Application components communicate only through services and can be plugged in to any infrastructure that implements the standardized service
- Uses abstraction and is based on XML over SOAP
- Largely independent of implementation attributes
- Loosely coupling between interaction software components – leads to re-use of software components
- Designed to follow publically accessible models for consumption
- Meant for enabling participation of multiple organizations
- Requires additional layers
  - Business layer => Service and business model / components
  - Service Bus / Service Facade
  - BPM
## Traditional Architecture vs Service Oriented Architecture

### Traditional Architecture

**STANDARDS**
- Involves only traditional J2EE and Web related standards
- Uses only HTTP
- Uses HTTPS for security
- More or less stable set of standards

**USAGE**
- Process centric
- Known context of usage

### Service Oriented Architecture

**STANDARDS**
- Includes standards related to Web Service
- Builds a messaging layer above HTTP using SOAP
- Prefer WS-Security for end-to-end security
- Implementations must deal with evolving set of standards

**USAGE**
- Workflow centric
- To a large extent, future context of usage unknown at the time of design i.e unknown users and usage platforms
Key components of SOA
Key components of SOA

- Services (common denominator)
- Service Description
- Advertising and Discovery
- Specification of associated data model
- Service contracts
Associated Terminology

- EAI
- OASIS
- XML
- UDDI
- SODA
- XSLT
- SOAP
- XSD
- Web Service
- Messaging
- Digital Signature
- WSDL
- SAX
- BEPL4WS
- DOM
- XSLT
- BPM
- Schema
- DIME
- DTD
### Associated Terminology

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPO</td>
<td>Business Process Outsourcing</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>ESP</td>
<td>Enterprise Service Provider</td>
</tr>
<tr>
<td>GDM</td>
<td>Global Delivery Model</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SODA</td>
<td>Service Oriented Development of Applications</td>
</tr>
<tr>
<td>SOBA</td>
<td>Service Oriented Business Applications</td>
</tr>
<tr>
<td>SOE</td>
<td>Service Oriented Enterprise</td>
</tr>
<tr>
<td>WS</td>
<td>Web Services</td>
</tr>
</tbody>
</table>
### Key Standards of and Technology of SOA

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XML</strong></td>
<td>Markup Language designed to carry/transport data. Structure of the document i.e. the tags can be user defined based on the data being transported.</td>
</tr>
<tr>
<td><strong>Web Services</strong></td>
<td>Loosely coupled software components delivered over Internet standard technologies.</td>
</tr>
<tr>
<td><strong>SOAP</strong></td>
<td>Message format communication between parties involved in a web service.</td>
</tr>
<tr>
<td><strong>WSDL</strong></td>
<td>Mechanism for describing a web service in a platform independent way.</td>
</tr>
<tr>
<td><strong>UDDI</strong></td>
<td>Facilitates registration and organization of web service descriptions into a searchable directory.</td>
</tr>
</tbody>
</table>
SOA Platform
Challenges of SOA

- Technical Challenges
  - Security challenges - loosely coupled environment
  - Performance - XML brings robustness not speed
  - Optimization
  - Organizing the services – registry & repository
  - Finding the right services and right interfaces
  - Transaction management is complex in interactions between logically separate system
Where SOA made a difference

- **eBay**
  - Abstracting enterprise information
  - Helped to manage more than 2 petabytes of data
- **IBM**
  - 77 shareable and reusable services in production
  - Reduced application inventories
- **Hewlett Packard**
  - Reuse across services
  - Cutting operational costs
- **Amazon.com**
  - Handle 60 million customers and one million partners
  - Handle growing Transactional load
- **Citi Group**
  - Governance
  - Enable “separation of powers” among corporate, divisions, departments
- **DreamWorks**
  - Simplify and consolidate key business operations
  - Use SOA to make movies a easier process
- **Volvo**
  - Better customer service by linking all dealership in Belgium
Why is SOA Different?

(1) **Terminology**: Both IT people and business people know what a service is.
(2) **Interoperability**: The interfaces and the wire protocols are based on standards.
(3) **Extension and Evolution** not rip and replace.
(4) **Reuse** of both functionality and machine resources.
SOA Defined

“SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business process realization using interface-based service descriptions.” From a paper by Borges, Holley and Arsanjani.
Web Services

- Web Services are XML-based technologies for messaging, service descriptions, discovery, and external features providing:
  - Pervasive open standards for distributed computing interface descriptions and document exchange via messages
Web Services

- Independence from the underlying execution environment and application platforms.
- Extensibility for enterprise qualities of service such as security, reliability, and transactions.
- Support for composite applications such as business process flows, multi-channel access, and rapid integration.
XML Preferred (1)

- The Extensible Markup Language is a common, independent data format across the enterprise and beyond that provides:
  - Standard data types and structures, independent of any programming language development environment or software system.
XML

- Pervasive technology for defining business documents and exchanging business information, including standard vocabularies for many industries.

- Ubiquitous software for handling operations on XML, including parsers, queries, and transformations.
Build an SOA in 8 Steps

(1) Business needs come first (not services) What problem are we trying to solve?

(2) What aspects can be implemented as services? Old services? New services? Legacy wrappers?

(3) Track services with registries and repositories.

(4) Govern the services. We need to encourage desired behavior at many levels, across enterprises, and at different stages. We need to monitor behavior, enforce policies & assess user satisfaction.

(5) Secure the services. established standards, we need privacy, identification authentication, and authorization. This may need to be federated security (over more than one organization.)
Build an SOA in 8 Steps

(6) Manage the services. Are messages arriving on time? Is everything operating properly?

(7) Virtualization through mediation. Are we free to move and change the services? Do we need an ESB that acts as a central hub for message routing and transformations?

(8) Design for interoperability through the adoption of standards.