An Introduction to Service Oriented Architecture
Definitions

- Sommerville
  - “Service-oriented architectures (SOAs) are a way of developing distributed systems where system components are stand-alone services, executing on geographically distributed computers”

- OASIS (Organization for the Advancement of Structured Information Standards)
  - “A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”

OASIS
Introduction

Components

• services
  – “A loosely-coupled reusable software component that encapsulates discrete functionality, which maybe distributed and programmatically accessed.” SOMMERVILLE

Connectors

• messages
  – meta-data
    • service descriptions, service interface etc.
    • semantic meta-data
Introduction

Functional Apps

Enterprise Apps

SOA Apps

Paul A. Strassmann
Professor of Information Sciences
Volgenau School of Information Technology and Engineering at George Mason University
http://www.strassmann.com/
Characteristics

Integration

- “mesh-up” of different services
  - does a newspaper qualify as an SOA?
- an application
  - various services linked together
  - really an ad-hoc application

Scope

- narrowly focused
- services are simple and generally perform a single task
Characteristics

Dependency

- deployment
- execution
- usage
  - need to know what a service expects and what it returns

Stateful vs. Statelessness

- ask the class??
Characteristics

Loose coupling

• service bindings can change whenever
  - different but equivalent services can execute at different times
• interface definition does not change

Reusability

• desirable to have reusable services
Characteristics

Independent of

- Platform
- Implementation

Discoverable

- published
- discovered via discovery mechanisms
  - UDDI – web services
  - simple search
Ian Gorton, Essential Software Architecture
Considerations

Performance

- **Computational penalties**
  - introduction of extra layers
  - slower than native/binary RPC
  - *Do we need to use XML based RPC?*

- **Communication latency**

Evolution

- **how should we handle legacy systems**
  - wrap the legacy system in service wrappers
Considerations

Service granularity

- reuse vs. performance

Fault-Tolerance

- partial failure vs. complete failure
- idempotent request

Service agreement

- availability, cost,
- performance
Considerations

Governance

- increases in difficulty
  - with an increase in diversely deployed services
  - many service providers
- meta-data management
- trust
Three Architectural Perspectives

Application Architecture

● business facing solution
● consumes services from one or more providers
● integrates them into the business processes

The Service Architecture

● a bridge between the implementations and the consuming applications
● logical view of sets of services which are available for use

Three Architectural Perspectives

The Component Architecture

- various environments supporting
  - the implemented applications
  - the business objects and their implementations

Three Architectural Perspectives

Considerations

Marginal Benefit

- benefits offered to the first application
- benefits offered to the nth application

Testing

- lack of tools
- lack of test services environment

Security

- authentication (WS-Security, SAML, WS-Trust)
- still very green
Infrastructure Services

Fundamental service layer

- data
- security
- computing
- communication
- applications
Another Look

SOA

• business-centric IT architectural approach
• consuming a service is usually cheaper then doing the work

Example

• DNS
  – reusable, scalable, fault-tolerant, well defined scope
Web Services

- would you classify web-services as SOA?
- web services
  - a technology
  - middle-ware
  - an implementation of SOA
From Applications to Services
From Applications to Services

Nan Yang Chief Architect (Client Solutions Greater China Sun Microsystems) ICSE Shanghai 2006
From Applications to Services
From Applications to Services

Check Customer Status
Check Credit
Check Inventory
Check Order Status
Create Invoice
Elemental Business Services

Custom Marketing System
AS400 Sales System
Oracle CRM System
SAP Finance System
Red Prairie Warehouse Mgmt. System
Another Business Unit
External Trading Partner

Data Repository

Nan Yang Chief Architect (Client Solutions Greater China Sun Microsystems) ICSE Shanghai 2006
From Applications to Services

Nan Yang Chief Architect (Client Solutions Greater China Sun Microsystems) ICSE Shanghai 2006
From Applications to Services

Nan Yang Chief Architect (Client Solutions Greater China Sun Microsystems) ICSE Shanghai 2006
From Applications to Services
From Applications to Services

Business Processes Are Composed Hierarchically to Create Composite Applications

Check Customer Status  Check Credit  Check Inventory  Check Order Status  Create Invoice

Custom Marketing System  AS400 Sales System  Oracle CRM System  SAP Finance System  Red Prairie Warehouse Mgmt. System  Another Business Unit  External Trading Partner

Data Repository

Nan Yang Chief Architect (Client Solutions Greater China Sun Microsystems) ICSE Shanghai 2006