Service-oriented architecture in e-commerce applications
What is a Service Oriented Architecture?

• Depends on who you ask
  – Web Services
  – A technical architecture
  – An evolution of distributed computing and modular programming
  – An IT strategy based on a philosophy of sharing IT services with the aim of achieving business agility
  – All about business-IT alignment

• Service-oriented enterprise
  – Service-oriented view to all the operations (c.f., process-oriented view. Enabler or the next step?)

Comes down to two things:

1) Enabling communications based on (mostly open and agreed upon XML-) standards

2) Thinking in terms of services
In terms of services...

• From a more practical point of view, the idea is to add an additional layer of abstraction upon to (existing) programmatic applications that renders their functionality as services
  – Services are typically fairly coarse-grained and business-oriented
  – Reusability is essential: “We took designs and held them up to the light. We asked, if we built it, how much reuse would we get?” G. Glass, BT Chief Architect
In terms of services...

- SOA organizes enterprise IT around services rather than applications
  - Discrete functions contained in enterprise applications become interoperable, standards based services that can be combined and reused quickly to meet business
  - Application functionality, components, data, processes, …, even people
    WS-BPEL Extension for People
- Because of their independent, modular nature, services can be used like building blocks to quickly and easily develop new business processes and composite applications, which can be simultaneously reused and deployed across multiple delivery channels
- ZapThink The Lego® Model of SOA
In terms of standards… (All of them use XML language)
The core WS standards

UDDI gives a services listing
WSDL describes the service
Technical service interface (+SLA, policies, etc.)

WSDL

UDDI

reads

Is accessed using SOAP

describes + registers

Application

SOAP client

communicates

SOAP server

Application

SOAP messages (envelopes) carrying the message payload
Carrier usually HTTP but SOAP messages can be transferred practically via any channel (service bus, email, FTP, …)
‘Black box’

- Something goes in
- Something happens
- Something comes out

NOK
Web services as black boxes

- Black box implementation is based on the API; i.e., WSDL description that tells the requester:
  - What functionality the ‘box’ has to offer
  - How it should be requested: input message and data types
  - How it responds: output message and data types
  - Where to send the request (physical location of a service), etc.

- [http://www.w3.org/2001/03/14-annotated-WSDL-examples](http://www.w3.org/2001/03/14-annotated-WSDL-examples)
SOA solution stack

An example…

SELECT custName, custAddress
FROM Customers
WHERE custID = 21323;

Requires direct access to the database through a supported API, intimate knowledge about the data models and schema, meticulous access control, …
Software packages built on top of existing systems.

Encapsulate a set of related functions or data.

Communicate via interfaces.

Reusable and substitutable.

May be implemented within the boundaries of a company’s IT infrastructure or remotely.

See Jups’ slides on components.

Related components working together
Data-as-a-Service

SELECT custName, custAddress
FROM Customers
WHERE custID = 21323;

Predefined reusable services
No direct connection to the database
Instead, data access via atomic services

SOA is about raising the level of abstraction so that requirements and business process can be discussed in a language understood by business people as well as IT folk (Graham, 2008)
Composite Applications automate new types of applications

- New functionality composed of parts from existing systems
  - Automation of cross-functional processes that span the boundaries of the organization and of existing systems
  - Support for strategic processes that require flexible workflows and integration of collaboration and unstructured information – such as documents and spreadsheets – with transactional systems for finance and operations
  - Expansion of the existing user base for enterprise applications within the organization by extending access and improving support for specific roles through focused used interfaces
  - Tighter integration with systems of supplier and key partners
  - Direct access for customer, suppliers, and key partners
  - Increased support for targeting specific niche markets
  - Improvement of change management processes such as mergers and acquisitions or program management

Woods (2008), Enterprise Services Architecture, O’Reilly
Business Process Execution Language (BPEL)

- OASIS standard for specifying business processes based on web service operations
  - Combines IBM’s Web Services Flow Language and Microsoft’s XLANG standard
- A language for process orchestration
  - Orchestration specifies an executable process that involves message exchanges with other systems;
  - Choreography specifies a protocol for peer-to-peer interactions, defining, for example, the permissible sequences of messages exchanged, with the aim of guaranteeing interoperability (Graham, 2008)
- Features a comprehensive syntax for describing business workflow logic
  - Abstract processes to describe the business protocols
  - Executable processes that can be compiled into runtime scripts
Driven by the business needs

- Combined loosely coupled services to support business requirements directly
- Elevating the level of abstraction so that requirements and business process can be discussed in a language understood by business people as well as IT folk
Business Process Execution Language (BPEL)

- Executable process encapsulates the process description within a Web service
- The BPEL document essentially describes the sequence and logic behind other services managed by the process
- A BPEL process description incorporates numerous language constructs that can accommodate most traditional workflow requirements
- The WSDL document representing the BPEL process contains interfaces (portTypes) for the process service itself, as well as any additional services involved with the execution of the process
Graphical representation

- No standards available
- Mapping to and from BPMN (with a reasonable success)

Example at [http://www.activebpel.org/samples/samples-3/BPEL_Samples/Resources/Docs/loan_approval.html](http://www.activebpel.org/samples/samples-3/BPEL_Samples/Resources/Docs/loan_approval.html)
XML representation of the loan approval process
Supporting WS-standards

• WS-Coordination to provide a fundamental context-management framework
• The business activity coordination type, as defined by WS-Transaction, is utilized to establish a standard mechanism for managing long-running services, including compensation
• WS-Choreography, WS-Atomic Transaction, WS-BusinessActivity, Xpath and others
Services domain

Web services for e-commerce

• Integrating internal systems and the ones of business partners.
  – Applications of the traditional EAI domain
  – Creating a business-driven services eco-system
• New possibilities with 3rd party services
  – e.g., AWS [http://aws.amazon.com/](http://aws.amazon.com/)
  – Amazon has been in the Web services business since its launch of (AWS) in 2002.
    • Prior to Web Services, Amazon replicated its entire infrastructure for partners, using customized modules tightly coupled with its underlying infrastructure.*
  – The services provide software developers, Web site owners and merchants with access to back-end features found on Amazon's Web site, such as its payment system.
    • Reuse of core functionalities located in legacy applications.
  – As the services have evolved, the face of the companies using them has changed.

* Kulkarni, Kumar, Mani & Padmanabhuni “Web Services: E-Commerce Partner Integration”, March/April 2005 IT Pro
Figure 1. Generic partner integration framework.

* Kulkarni, Kumar, Mani & Padmanabhuni “Web Services: E-Commerce Partner Integration”, March/April 2005 IT Pro
Summary

- Relatively cheap investment
- Builds on top of existing solutions
- Allows easy ‘outsourcing’ of non-core, laborious, expensive etc. functions
- Universally approved standards but
  - While Web services offer many advantages, they face a few deployment challenges. "Web services technology is readily available and works well. The obstacles stem from management issues," said ZapThink's Bloomberg. Many programmers do not want to fully understand how to build applications that take advantage of Web services features. Inertia is also a problem: Some programmers resist designing applications in a new way. Companies often need to invest in training courses in order to overcome those barriers.
Summary

• Most existing enterprise software infrastructures (especially old and legacy) conform to other architectures, such as client/server, thin client, or mainframe computing.
• Therefore, it can take a substantial amount of time, money and manpower to revamp their systems and realize Web services' potential benefits. Or does it?
• c.f., Ari’s case