Information Technology

The Art of Integration

Chris Huff,
Manager Enterprise Application Integration,
B2B Architectures @
The Home Depot
Overview

• Intro
• History - A Recurring Theme
• Current Integration Philosophy
• Challenges within the Enterprise
• Implications – How to Prepare
Common Theme?

What’s was missing:
- Ubiquitous Standards
- Open standards
- Open community

Goal → Flexibility and Reuse

What is still missing:
- Consistency
- Discipline
Current Philosophy – The Theme Continues

Quality Works
ReUse

Appliances
Donations to our nonprofit training business are tax deductible.

Open
Wednesday – Saturday
10 a.m. to 6 p.m.
Current Integration Philosophy

- EAI, ESB, SOA, ESA, EIEIO…
- BPM, Workflow
- ESB is very similar to that of "Enterprise Application Integration" (EAI) tools, apart from three significant terms:
  - Web services, ubiquitous and lightweight. ¹
- “The emergence of the ESB Concept is closely linked with the lasting trends that have been slowly transforming the EAI market for the last few years: standardization of infrastructures with
  - J2EE
  - Microsoft .NET
  - Web services
  - the redistribution of roles of integration software component vendors.” ¹

¹ - http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci913058,00.html
Service Oriented Architectures

How do I deliver business flexibility?

Can Information Become a Service?
What is Driving Demand?

• Why the Business demands agility:
  – Fast integration of Mergers/Acquisitions
  – Adapt to new and increasing customer expectations
  – Quicker response to business, market and industry changes
  – More effective use of invested capital

• Why IT needs flexibility:
  – Increasingly complex systems
  – More distributed processing
  – Larger data centers
  – More change, less time
Recurring Theme – Abstraction layers

- BI Apps
- Portals
- B2B Networks
- Composite Apps

EAI (application integration)
BPM (process management)
ESB (services integration)

DATA SERVICES
(data abstraction layer)

Files
Message Queue
XML
EDI
Unstructured Data
Application Databases
Mainframe
Data Warehouse
Why?

• To deliver applications faster!
  “Shifts focus to application assembly rather than implementation details”

Benefits to SOA
*(Rob High, IBM Chief Arch SOA)*
1. Customer Satisfaction
2. Business Operations Efficiency
3. Process Efficiency
4. Reuse

1 “ARTS Retail Enterprise Architecture & Services Oriented Architecture” - Copyright © 2006
Principles Should Chart the Course

Forget the acronyms and stick to the principles

- Low total cost of ownership
  - Repeatability
  - Ease of maintenance
  - Object Reuse
  - Reliable performance
  - Rapid development
- Adherence to architectural principles
  - Design that is flexible and versatile (abstraction between applications)
  - Application Decoupling (Loose coupling reduces assumptions two parties make about each other, while tightly coupled architectures tend to result in brittle, hard-to-maintain, and poorly scalable solutions)
  - Ease in importing application metadata
- Highly Scalable, Secure and Reliable
  - High volume processing capability
  - Scalability

Always ask yourself, “why is this better than point-to-point?”

- If it is not cheaper, then don’t do it
- If it’s not providing a desired end-state, then don’t do it (be careful with this one)
- If it can’t support your volumes and growth, then don’t do it
What’s the Return?

Significant cost savings will not happen without discipline

Big Payoffs at the End
- Dynamic application development (composites/mashups)
- Dynamic Business Processes
- Organizational Speed/responsiveness
- Customer Service

Incremental ROI along the way
- Granular Configurable Services
- Graphical development
- Extensibility (regression testing)
- Loose Coupling (upgrades, flexibility)
- Consistency, Standards
Challenges In the Enterprise
Political/Organizational

• Integration is often an emotional topic
  – Infrastructure often represents the “Commanding Heights” of a company. Who has greater control, the person making the decision or the person providing the options…(Truex)

• Budgets
  – Who pays for the additional work required to create reusable services or canonical XML formats

• The KISS clan vs. the early adopters vs. legacy nay-sayers
  – “Keep it simple at all cost”
  – “We must have the latest in greatest or we’ll get behind”
  – “Why change at all?”

• Consultants! Management will assume they have all the right answers, but they too have limitations:
  – They will implement what they’ve seen (we hire them for this reason, but you need to be aware of this when implementing an emerging technology or strategic architecture.)
  – Many do not act in the best interest of the company
Political/Organizational

Who owns what

• Development (COE vs. Dispersed)
  - Infrastructure Architecture
  - Integration Architecture
    “Effectively Managing Information Systems Architecture Standards: An Intra-organization perspective” Boh, Yellin, Dill and Berbsleb

• XML and Canonical formats
• Service Repositories
• Application Integration Standards
• Web Services
Vendors Muddy the Waters

• Application vendors have their own agenda despite putting “open standards” on the box
• Middleware vendors over-promise
  – Rhetoric - vendors create problems for the industry by over-promising results/functionality
  – Artificially create demand
  – Over-packaging products: do I need to buy everything or can I build?
• You simply cannot buy SOA or any other architecture
Implications & How to Prepare

• Get executive buy-in early
• Know your architecture – ADLs
• Focus on the principles and employ enterprise standards
• Organize around service orientation
• Establish standards that support & enforce cost savings
• Begin counting things and then publish cost savings & statistics
• Create an environment that fosters creativity and innovation
Document the Architecture
Build a Conceptual View
Rally Projects Around Your Vision

- Rollout Enterprise Service Repository
- Service Governance

- Capability for Service Oriented Processes

- Service enabled Processes reused
  - New apps built from service composition & Mashups

- Dynamic Services

- Dynamic Service Discovery

- Store Integration Standards
  - Managed Services

- Loosely Coupled (LC) Applications
  - Some LC Services
  - Initial services/reuse
  - Messaging Std
  - Some XML Stds
  - Interface Visibility

- Prepare Governance Standards
  - Select ESR
  - Establish SOA Governance Panel

THD Standards Adoption
- Industry Standards Adoption
  - Technology Standards Adoption
Implement Standards

- Know and plan your layers of reuse: code, modules/objects and services
- Certify that all service interfaces are extensible
- Spend time up front on API/Interface design
- Develop Middleware/SOA Cost Model
- Ask COTS package providers to supply “out-of-the-box” industry standard formats to exchange data
- Don’t get caught in the Web Service guise of loose coupling, broker synchronous calls unless they are true Web Services