Practical Experiences with SOA

Paul Giangarra
Sr. Technical Staff Member

e-mail: ppgx@us.ibm.com
Agenda

- Paul’s Theory of Swivel Chair Integration
- SOA Core Concepts
- ESB Core Concepts
- Architecture and SOA
- Patterns and SOA
- Experiences with SOA
- Past, Present, and the Future
Swivel Chair Integration

1969 (Apollo 11) to 1983 (Shuttle Mission 9)
2003 (Today): Air Operations Center

http://images.pennnet.com/articles/mae/cap/cap_129312.jpg
More 2003 (So What’s Changed?)

Swivel Chair Integration: The End Result?
SOA 101 – SOA and Web Services

What is SOA and how is it related to Web Services?

- A SOA consists of a set of business aligned I/T services that support an organization’s business process goals and objectives, ... using interface-based service descriptions that decouple the provider and consumer through open standards and protocols. These services can be combined and choreographed to produce composite enterprise scale services that allow dynamic re-configuration of business value-nets and I/T systems.

- SOA can be realized through Web Services. Web Services are a standards-based technology that represents both a business process and IT capability that can enable a more flexible and responsive enterprise regardless of underlying technologies, platform or programming language.
SOA Is Key to Business Flexibility

The flexibility to treat elements of business processes and the underlying IT infrastructure as secure, standardized components (services) that can be reused and combined to address changing business priorities.

Building blocks are services

- Packaging business functions from new and existing applications in a simple and standardized way creates services that are available for use.
- Services are used to help get the right information to the right people at the right time.
- Services can be reused and combined to deploy composite applications to address new opportunities.
- Increasing use of “Web” services based on open standards complements existing services technology.
Service Oriented Architecture

- Application functionality contained in building blocks
- Create complete applications by connecting blocks together
- Model business relationships by providing blocks to others and using others' blocks
- Building blocks are services

Service Oriented Architecture (SOA) is a composite concept because it both describes both
- a mode of business process organization and implementation (as an orchestrated collection of services) and
- prescribes a way of delivering those services through information technology, i.e. a system development paradigm.
SOA: The Synergy Between Business and IT

SOA offers a holistic synergy between business and IT that allows the business flexibility

SOA can be expressed as

- a **set of architectural principles** and patterns which address characteristics such as *modularity, encapsulation, loose coupling, separation of concerns, composition, choreography, and single implementation*

- an **architectural style** in which a service provider offers a service description, a service requestor finds the description and binds to the implementation often through a service directory

- as a **programming model** complete with standards, tools, methods, and technologies such as web services

- A **linkage between business functions and aligned IT services** that collectively support changeable business functionality in an environment where changes in non-functional requirements often imply changes in service providers
SOA Is an Interoperability Enabler

- **A Service Oriented Architecture (SOA)**
  - Leverages open standards to represent software assets as services
  - Provides a standard way of representing and interacting with software assets
  - Allows individual software assets to become building blocks that can be reused in developing other applications
  - Shifts focus to application assembly rather than implementation
  - Can be used internally to create new applications out of existing components, or to integrate with applications outside of the enterprise

- **Business Values of a Service Oriented Architecture**
  - Makes development simpler
  - Decreases the complexity, risk, and cost of integration
  - Maximizes application and development flexibility
  - Optimizes existing IT investments and promotes reuse

---

“By 2007, SOA will be the mainstream software engineering practice, ending the 40-year domination of monolithic software architecture”
-- Gartner Group

“Enterprise customers who implement such SOAs will potentially see substantial business benefits in the form of increased business agility.”
-- Jason Bloomberg, Market Research Analyst, ZapThink LLC
SOA is based on a 3-layer model of “components”, “services” and “processes”.

- It can be enabled through the Web Services open standard.
- If done right, services based interoperability will facilitate an enterprise wide transformation.
SOA and Web Services Adoption

Entry points
Based on business priorities

1. Implementing individual Web Services
   - Creating services from tasks contained in new or existing applications

2. Service-oriented Integration of Business Functions
   - Integrating services across multiple applications inside and outside the enterprise for a business objective

3. Enterprisewide IT Transformation
   - An architected implementation enabling integration across business functions throughout an enterprise

4. On Demand Business Transformation
   - Broad transformation of existing business models or the deployment of new business models

Different paths to eventual adoption of SOA depending on your business goals and IT constraints
## Proven Solutions for Enterprises at All Stages

<table>
<thead>
<tr>
<th>SOA adoption paths</th>
<th>Web services role</th>
<th>Customer examples</th>
</tr>
</thead>
</table>
| 4 On demand business transformation | Broad transformation of existing business models or the deployment of new business models | Enables standards-based connection of entire value network  
Services, products, partners and process flows may vary with market conditions and services available | BEXINS |  
Since 1891 |
| 3 Enterprise Wide IT transformation | An architected implementation enabling all business functions throughout an enterprise | Enables a publishing and consumption of services through self-description and—dynamic combination—within a service-oriented architecture | Charles Schwab | MIAMI-DADE |
| 2 Service-oriented integration of business functions | Integrating services across multiple applications inside and outside the enterprise for a business objective | Integrates partners, divisions or channels with Web services for basic transactions  
Integrates a private network of partners or divisions | Boeing | AVNET | IBM | US Bank | AVIS | VISA | BlueCross | CIBC |
| 1 Implementing individual Web services | Creating services from tasks contained in existing or new applications | Exposes existing functionality as Web services, unlocking value from current infrastructure | PNC | Abbey | Aspen Inn at the Buttermilk |  
Colorado Software Summit |  
Paul Giangarra — Practical Experience with SOA |  
Page 15 |
Interoperability Leads to Transformation, and...

Transformation includes...

People

Technology

Business

build  run  manage
What We’ve Learned about SOA Implementation

- Define business rationale, not technical features
- Find a pragmatic balance between technical rigor & time to market
- Value ongoing flexibility over a one time efficiency gain
- Invest (where useful) in a diversified portfolio of applications
- Centralize business processes across business units, partners and valued customers
- Actively manage risk to balance competitive advantage and systemic performance

And, what some others are saying...
7 Fallacies of SOA

Fallacy #1: There's Nothing New Under the Sun, and SOA Is No Exception.

Fallacy #2: SOA is a Revolutionary Paradigm Shift

The ZapThink Take:

“...The best advice ZapThink can give people who are considering SOA is to tackle such an initiative with your eyes open. While SOA isn't all hype, there's no question there's plenty of hype out there – exaggerating SOA's strengths as well as its weaknesses. Always remember that SOA is challenging and often quite risky, so solid education, thorough preparation, and a careful approach are all important. But also remember the promise of SOA – building an IT infrastructure flexible enough to respond to the needs of the business. With a value proposition as broad and strategic as that, it's easy to accept that SOA is inevitable.”

Fallacy #3: SOAs are All Hype, No Substance

Fallacy #4: SOA is a Panacea

Fallacy #5: The Overhead from SOA Leads to Unacceptably Poor Performance

Fallacy #6: A Bottom-Up Approach to SOA is Good Enough

Fallacy #7: SOA is Optional

Source: Jason Bloomberg in ZapThink of August 5, 2004
# Gaps in SOA — and What Will Fill Them

The missing pieces of Web services-based SOAs will require mature Web services standards, many of which are years away from fruition. Meanwhile, more conventional technologies can get the job done.

<table>
<thead>
<tr>
<th>GAP</th>
<th>DESCRIPTION</th>
<th>KEY SOFTWARE</th>
<th>STANDARD TO WATCH</th>
<th>YEAR MAINSTREAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>Guaranteed delivery of messages, including support for complex message models</td>
<td>Messaging-oriented middleware and enterprise service bus</td>
<td>WS-Reliable-Messaging</td>
<td>2006</td>
</tr>
<tr>
<td>Security</td>
<td>Federated, policy-based authorization and authentication</td>
<td>Distributed identity management</td>
<td>WS-Policy</td>
<td>2006</td>
</tr>
<tr>
<td>Orchestration</td>
<td>Design and execution of composite Web services</td>
<td>Web services-savvy BPM tools</td>
<td>BPEL</td>
<td>2007</td>
</tr>
<tr>
<td>Legacy support</td>
<td>Incorporation of legacy systems and packaged applications into SOA</td>
<td>XML application adapters</td>
<td>N/A</td>
<td>2006</td>
</tr>
<tr>
<td>Semantics</td>
<td>Mapping specific business meaning to data and services</td>
<td>Cross-functional Web services</td>
<td>Industry-specific schemas</td>
<td>2007</td>
</tr>
</tbody>
</table>

Full article here: [http://www.infoworld.com/article/04/09/10/37FEwebservmiddle_1.html](http://www.infoworld.com/article/04/09/10/37FEwebservmiddle_1.html)
Gartner Predicts: (November 2003)

- By year-end 2005, enterprises will deploy Web Services management platforms in 65% or more of major Web Services implementations (0.7 probability).
- By 2006, more than 70% of new applications will use Web Services in some part of their architecture (0.8 probability).
- In 2006, more than 80% of business application products sold worldwide will be service-oriented business applications (0.7 probability).
- In 2006, Web Services standards and technologies will influence more than 60% of the $527 billion IT professional services market (0.7 probability).
- By 2007, IT professional services will account for more than 50% of the revenue of large enterprise application software vendors, creating a convergence of the software and IT professional services markets (0.7 probability).
- By 2008, services-oriented development of applications plus SOBAs* will enable Type A enterprises to increase programmer productivity be more than 100% (0.8 probability).

*SOBA = Services Oriented Business Application
7 Software Trends

1. Disappearance of Bloated Operating Systems
2. Evolution of Components and Objects
3. Rise of Mobile Code
4. Normalization of Distributed Computation
5. Proliferation of Embedded Systems
6. Mass Adoption of Wireless Networks
7. Change in Payment Models


http://www.parkway-g.co.uk/journals/infosec.shtml
SOA 102 – The Enterprise Service Bus

What is an Enterprise Service Bus?

- Enterprise Service Buses typically use messaging technology combined with a service-oriented architecture, XML, Web services protocols and intelligent routing to tie together disparate systems.
- Vendors and analysts say it's a less costly alternative to proprietary integration approaches, and they expect widespread adoption.
Gartner on ESB

Gartner predictions for Enterprise Service Bus:

- More than half of all large enterprises will have an enterprise service bus running by year-end 2006 (0.7 probability).
- One-third of all application development projects in 2007 will use an ESB (0.6 probability).
- All major integration suites will have an ESB core by YE06 (0.7 probability). (Most currently have a MOM core)

Alternatives for customers considering ESB according to Gartner:

- Buy an ESB from a pioneering vendor – leading edge projects seeking competitive advantage and time-to-benefit
- Build your own ESB using MOM, application servers, Web services tools and custom development – large, guru-rich development teams seeking to minimize use of small vendors
- Buy an integration suite – high end projects with major reliance on legacy and purchased applications
- Wait for the big vendors to offer a full ESB – conservative projects seeking to minimize near term risk and investment

Gartner recommends:

- “Build to integrate” or “buy to integrate” all major business applications using SOA and basic forms of EDA.
- Prepare for event-driven architecture: the next big thing.”
- Add an ESB to your IT strategic plan and application architecture.

Source: Roy Schulte, VP Distinguished Analyst at Gartner
The ESB represents a highway, mediating and choreographing the flow of messaging traffic from one point to another.
What Is inside the Enterprise Service Bus?

The ESB is an integrated set of middleware services supporting:

- **services oriented architectures** – where distributed applications are composed of granular re-usable services with well-defined, published and standards-compliant interfaces
- **message driven architectures** – where applications send messages through the ESB to receiving applications
- **event driven architectures** – where applications generate and consume messages independently of one another

These middleware services include:

- **Communication middleware** supporting a variety of
  - communication paradigms,
  - qualities of service,
  - APIs, platforms, and standard protocols
- A mechanism for injecting **intelligent processing of in-flight service requests and responses** within the network
- **Standard-based tools** for enabling rapid integration of services
- **Management system** for loosely-coupled applications and their interactions
Aspects of the Enterprise Service Bus

Higher-level function insulated from connection “details”

Customize interactions between services (examples)

Rich Comm patterns and QoS

Connect anything (examples)

Service Selection
Data Logging
Customized Routing
Format Translation

Enterprise Service Bus

Queues  Pub/Sub  Req/Rep  Assured  Secure  Available

MQ  SOAP/HTTP  JMS  WBI  CEI  .NET Adaptors
The ESB in the On Demand Operating Environment

Enabled By:
- XML
- Web services
- J2EE
- Open Grid Services Architecture
- Common Information Model
- Integrated system console
- Security and identity
- Choreography
- Transaction coordination
- Data persistence
- Workload management
- Collaboration
- Application connectivity
- Configuration tool
Basic SOA Enterprise Architecture

Within an SOA, all elements are expressed as services
SOA Is Key to Business Flexibility

- An architecture consisting of set of business aligned I/T services that support an organization’s business process goals and objectives ...

- ...using interface-based service descriptions that decouple the provider and consumer through open standards and protocols...

- These services can be combined and choreographed to produce composite enterprise scale services that allow dynamic re-configuration of business value-nets and I/T systems.
SOA in the Greater Scheme of Things

Flexible Business Models
- Transformation
- Business Process Outsourcing
- Mergers, Acquisitions & Divestitures

Flexible IT Architecture
- On demand Operating Environment
- Service Oriented Architecture (SOA)
  - Development
  - Infrastructure
  - Management
    - Software Development
    - Integration
    - Infrastructure Management

Composable Processes (CBM)
Composable Services (SOA)

Greater flexibility required in business models and the supporting IT architecture
End Goal

Silos Integration → Full Integration

Physical Virtualization → Grid

Manual Automation → Autonomic

Proprietary Open Standards → Interoperable

NIH Commercial IT → Acquisition

Enterprise-Wide Strategies For Success
SOA Basic Principles

- **Decoupling of applications**
  - through the use of synchronous and asynchronous web service requests between a service consumer and a service provider.

- **Process Choreography**
  - coordinating service calls across several service providers.

- **Common information model**
  - enabling process flows to be designed using a common semantic representation of data objects even though the services accessed in the process have different data models.

- **Common service model**
  - allowing services to be defined using a common Web Services Description Language (WSDL). The services can be accessed over a variety of protocol bindings including SOAP over HTTP, SOAP over Java Message Service (JMS), RMI over IIOP, Java Connector Architecture and pure JMS. Leveraging legacy applications by using adapter technology to expose existing functionality in legacy applications as services, thereby allowing them to participate as service providers.

- **Event processing**
  - allowing service providers to send anonymous events which are routed based on content, to the appropriate services for processing.

- **Common Programming Model**
  - to improve reuse and better leverage the infrastructure. Should be based on open computing standards.
Getting Different vs. Getting Better

“In the past, executives had the luxury of assuming that business models were more or less immortal. Companies always had to work to get better... but they seldom had to get different – not at their core.”

–Gary Hamel and Liisa Valikangas

“The Quest for Resilience”

Harvard Business Review
Building a New Kind of Enterprise

- Need for **flexibility** and innovation is forcing organizations to break their business processes into manageable parts

- Applications mirror this approach, becoming increasingly **modular**

- **Simplification** of underlying IT infrastructure is required to manage and support changes in the business
Horizontal Integration Is the New Challenge

Bridging the gap between modernization & transformation and IT

Customer Connections | Internal Systems | Supplier Networks

- Retek
- PeopleSoft
- MatrixOne
- SAP
- manugistics
- i2
- Ariba
Competing in Today’s Environment

To achieve the benefits of on demand, enterprises must transform themselves to leverage partners and technology.

Business key imperatives
- Focused
- Responsive
- Variable
- Resilient

Businesses transformation
- Enterprise Reconstruction
  - from product silos to customer-led competencies
- Industry Deconstruction
  - From Integrated value chains to industry value networks

Technology requirements
- Open Standards
- Integrated
- Virtualized
- Autonomic
# Technology Requirements

## Enabled by Technological Developments

<table>
<thead>
<tr>
<th><strong>Business Imperatives</strong></th>
<th><strong>Technology Attributes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on differentiating competencies; partner elsewhere</td>
<td>Adopt open industry standards to enable enterprise collaboration</td>
</tr>
<tr>
<td>Act quickly through insight-driven decision making</td>
<td>Integrate systems, data, and processes within and across organizations</td>
</tr>
<tr>
<td>Move from fixed to variable cost structures</td>
<td>Exploit demand/usage-based IT capabilities and costing</td>
</tr>
<tr>
<td>Reduce operational and marketplace risks</td>
<td>Deploy advanced self-monitoring and self-healing technologies</td>
</tr>
</tbody>
</table>

- **Focused**
- **Responsive**
- **Variable**
- **Resilient**
- **Open Standards**
- **Integrated**
- **Virtualized**
- **Autonomic**

Source: IBM Institute for Business Value analysis
Business Imperatives

The drivers correspond to the business proposition for on demand and its four key imperatives

**Focused:**
- Concentrates investment on core functions that add differentiable value
- Targets customers best suited to its operating model
- Accesses best-in-class capabilities, leverages scale efficiencies through partners
- Drives incremental revenue from insourcing strong components

**Responsive:**
- Rapidly develops new products and integrates new capabilities and acquisitions
- Customizes products to fit target customer needs enabling a value premium
- Aggregates data across the organization, turning it into useful information

**Variable:**
- Supplements internal capabilities with outsourcing to achieve peak capacity
- Outsources non-core components completely
- Shifts cost structure from predominantly fixed cost to predominantly variable

**Resilient:**
- Knows exposures to operational, market and credit risk in real time
- Effectively distributes risk with strategic partners
- Reduces capital requirements through robust risk management
- Builds robust, “self-healing” organizational capabilities (processes and technology)
- Recovers quickly from external disruptions to operations
Business Imperatives: The Disconnect in Business

Traditional business models do not match the environment

On demand Imperatives

Focused
Responsive
Variable
Resilient

Disconnect

Comprehensive
- Extended control of value chain
- Product level optimization
- Best-in-class in all processes
- Few partnerships

Predictive
- Deliberate decision making process
- Forecasting and analysis for planning
- Direction based on assumptions and firm history

Committed
- Inflexible organizational structure
- Large investment in fixed assets
- Focus on existing business model

Cautious
- Contingency planning
- Risk averse business design
- Financial hedging

Traditional Business

Traditional business models do not match the environment

- Expanded control of value chain
- Product level optimization
- Best-in-class in all processes
- Few partnerships

- Deliberate decision making process
- Forecasting and analysis for planning
- Direction based on assumptions and firm history

- Inflexible organizational structure
- Large investment in fixed assets
- Focus on existing business model

- Contingency planning
- Risk averse business design
- Financial hedging
The Essentials of an On Demand Business

Where you start depends on YOUR organization’s priorities.
Demands on the IT Organization Continue

<table>
<thead>
<tr>
<th>Business Challenges</th>
<th>IT Imperatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase the speed of business changes</td>
<td>Become a more responsive IT organization to quickly adapt to changing business priorities</td>
</tr>
<tr>
<td>Improve business efficiency and performance</td>
<td>Align IT more tightly with business strategies in a cost effective manner</td>
</tr>
<tr>
<td>Protect the privacy and security of critical information assets</td>
<td>Provide a secure and managed integration environment</td>
</tr>
</tbody>
</table>
Where to Begin

- Start by deconstructing your business model – breaking it down into discrete business processes and functions

- These processes and functions are what we call service components

- Each service component serves a unique purpose and interacts with other service components in the business model, using agreed upon cost structures and service levels
Deconstruct & Conquer: the Component Business Model

1. First, break down your business into its components

<table>
<thead>
<tr>
<th>Example: consumer packaged goods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy</strong></td>
</tr>
<tr>
<td>Category/Brand Strategy</td>
</tr>
<tr>
<td>Category/Brand Planning</td>
</tr>
<tr>
<td><strong>Tactics</strong></td>
</tr>
<tr>
<td>Brand P&amp;L Management</td>
</tr>
<tr>
<td>Matching Supply and Demand</td>
</tr>
<tr>
<td>Marketing Development &amp; Effectiveness</td>
</tr>
<tr>
<td>Product Ideation</td>
</tr>
<tr>
<td><strong>Execution</strong></td>
</tr>
<tr>
<td>Concept/Product Testing</td>
</tr>
<tr>
<td>Product Development</td>
</tr>
<tr>
<td>Product Management</td>
</tr>
<tr>
<td>Marketing Execution</td>
</tr>
<tr>
<td>Consumer Service</td>
</tr>
<tr>
<td>Product Directory</td>
</tr>
<tr>
<td><strong>Product Management</strong></td>
</tr>
<tr>
<td>Customer Relationship Strategy</td>
</tr>
<tr>
<td>Customer Relationship Planning</td>
</tr>
<tr>
<td><strong>Customer Relationship</strong></td>
</tr>
<tr>
<td>Assessing Customer Satisfaction</td>
</tr>
<tr>
<td>Customer Insights</td>
</tr>
<tr>
<td>Account Management</td>
</tr>
<tr>
<td><strong>Manufacturing</strong></td>
</tr>
<tr>
<td>Manufacturing Strategy</td>
</tr>
<tr>
<td>Supplier Relationship Management</td>
</tr>
<tr>
<td>Production and Materials Planning</td>
</tr>
<tr>
<td><strong>Supply Chain &amp; Distribution</strong></td>
</tr>
<tr>
<td>Distribution Oversight</td>
</tr>
<tr>
<td>In-bound Logistics</td>
</tr>
<tr>
<td>Out-bound Logistics</td>
</tr>
<tr>
<td><strong>Business Administration</strong></td>
</tr>
<tr>
<td>Corporate Strategy</td>
</tr>
<tr>
<td>Corporate Planning</td>
</tr>
<tr>
<td>Alliance Management</td>
</tr>
<tr>
<td>Line of Business Planning</td>
</tr>
<tr>
<td>Business Performance Management</td>
</tr>
<tr>
<td>External Market Analysis</td>
</tr>
<tr>
<td>Organization and Process Design</td>
</tr>
<tr>
<td>Legal and Regulatory Compliance</td>
</tr>
<tr>
<td><strong>Corporate Planning</strong></td>
</tr>
<tr>
<td>Accounting and GL</td>
</tr>
<tr>
<td>Indirect Procurement</td>
</tr>
<tr>
<td>Facilities and Equipment Management</td>
</tr>
<tr>
<td>HR Administration</td>
</tr>
<tr>
<td>IT Systems and Operations</td>
</tr>
</tbody>
</table>
Deconstruct & Conquer: the Component Business Model

1 2 3 4 Next, decide what’s differentiating & what is simply operating

<table>
<thead>
<tr>
<th>Strategic view</th>
<th>Product Management</th>
<th>Customer Relationship</th>
<th>Manufacturing</th>
<th>Supply Chain &amp; Distribution</th>
<th>Business Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Category/Brand Planning</td>
<td>Customer Relationship Planning</td>
<td>Supplier Relationship Management</td>
<td>Supply Chain Planning</td>
<td>Corporate Planning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Production and Materials Planning</td>
<td></td>
<td>Alliance Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Line of Business Planning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tactics</td>
<td>Brand P&amp;L Management</td>
<td>Assessing Customer Satisfaction</td>
<td>Manufacturing Oversight</td>
<td>Distribution Oversight</td>
<td>Corporate Strategy</td>
</tr>
<tr>
<td></td>
<td>Matching Supply and Demand</td>
<td>Customer Insights</td>
<td></td>
<td>In-bound Logistics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marketing Development &amp; Effectiveness</td>
<td>Account Management</td>
<td></td>
<td>Out-bound Logistics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product Ideation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>Concept/Product Testing</td>
<td>Value-Added Services</td>
<td>Make Products</td>
<td>Distribution Center Operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product Development</td>
<td>Customer Account Servicing</td>
<td>Assemble/Pkg. Products</td>
<td>Transportation Resources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product Management</td>
<td>Retail Marketing Execution</td>
<td>Plant Inventory Management</td>
<td>HR Administration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marketing Execution</td>
<td>In-store Inventory Mgmt</td>
<td>Manufacturing Procurement</td>
<td>IT Systems and Operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer Execution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product Directory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: consumer packaged goods

Strategy
differentiation

Competitive parity

Basic

Strategic view

Next, decide what’s differentiating & what is simply operating
Deconstruct & Conquer: the Component Business Model

1 2 3 4 Then, analyze costs

<table>
<thead>
<tr>
<th>Financial view</th>
<th>Strategy</th>
<th>Tactics</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product Management</td>
<td>Customer Relationship</td>
<td>Manufacturing</td>
</tr>
<tr>
<td></td>
<td>Category/Brand Strategy</td>
<td>Customer Relationship Strategy</td>
<td>Manufacturing Strategy</td>
</tr>
<tr>
<td></td>
<td>Category/Brand Planning</td>
<td>Customer Relationship Planning</td>
<td>Supplier Relationship Management</td>
</tr>
<tr>
<td></td>
<td>Brand P&amp;L Management</td>
<td>Assessing Customer Satisfaction</td>
<td>Production and Materials Planning</td>
</tr>
<tr>
<td></td>
<td>Matching Supply and Demand</td>
<td>Customer Insights</td>
<td>Manufacturing Oversight</td>
</tr>
<tr>
<td></td>
<td>Marketing Development &amp; Effectiveness</td>
<td>Account Management</td>
<td>Supplier Control</td>
</tr>
<tr>
<td></td>
<td>Product Ideation</td>
<td>Value-Added Services</td>
<td>Make Products</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Customer Account Servicing</td>
<td>Assemble/Pkg. Products</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retail Marketing Execution</td>
<td>Plant Inventory Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In-store Inventory Mgmt</td>
<td>Manufacturing Procurement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Customer Directory</td>
<td></td>
</tr>
</tbody>
</table>

Example: consumer packaged goods

<table>
<thead>
<tr>
<th>High capital area</th>
</tr>
</thead>
<tbody>
<tr>
<td>High cost area</td>
</tr>
<tr>
<td>High cost &amp; capital area</td>
</tr>
</tbody>
</table>
Finally, prioritize your transformation initiatives

1. Seek external provider/external utility
2. Consolidate and/or create internal utility
3. Integrate and redesign
4. No action

Example: consumer packaged goods

<table>
<thead>
<tr>
<th>Transformational view</th>
<th>Strategy</th>
<th>Customer Relationship</th>
<th>Manufacturing</th>
<th>Supply Chain &amp; Distribution</th>
<th>Business Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Management</strong></td>
<td>Category/Brand Strategy</td>
<td>Customer Relationship Strategy</td>
<td>Manufacturing Strategy</td>
<td>Supply Chain Strategy</td>
<td></td>
</tr>
<tr>
<td>Category/Brand Planning</td>
<td>Customer Relationship Planning</td>
<td>Supplier Relationship Management</td>
<td>Manufacturing Oversight</td>
<td>Supply Chain Planning</td>
<td></td>
</tr>
<tr>
<td><strong>Tactics</strong></td>
<td>Brand P&amp;L Management</td>
<td>Assessing Customer Satisfaction</td>
<td>Production and Materials Planning</td>
<td>Distribution Oversight</td>
<td></td>
</tr>
<tr>
<td>Matching Supply and Demand</td>
<td>Customer Insights</td>
<td>Supplier Control</td>
<td>In-bound Logistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing Development &amp; Effectiveness</td>
<td>Account Management</td>
<td>Manufacturing Oversight</td>
<td>Out-bound Logistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Ideation</td>
<td>Value-Added Services</td>
<td>Supplier Control</td>
<td>Distribution Center Operations</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Execution</strong></td>
<td>Concept/Product Testing</td>
<td>Customer Account Servicing</td>
<td>Make Products</td>
<td>Transportation Resources</td>
<td></td>
</tr>
<tr>
<td>Product Development</td>
<td>Retail Marketing Execution</td>
<td>Assemble/Pkg. Products</td>
<td>Assembly/Procurement</td>
<td>En route Inventory Management</td>
<td></td>
</tr>
<tr>
<td>Product Management</td>
<td>In-store Inventory Mgmt</td>
<td>Plant Inventory Management</td>
<td>Customer Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing Execution</td>
<td></td>
<td>Manufacturing Procurement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumer Service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Directory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Seek external provider/external utility
Consolidate and/or create internal utility
Integrate and redesign
No action

- Example: consumer packaged goods
Reality Check — What You Want vs. What You Have

- The highest impact transformation priorities are typically horizontal processes
- They’re cross-division — and probably cross-company
- Yet today, elements of the new process are enmeshed in business unit silos and their corresponding monolithic applications and infrastructure
- Traditional delivery and financing options have limited flexibility
Next Step: the Architectural Views

- Two key views will be discussed here:
  - Business
  - Technical
- Obviously there are more (e.g. Operational, Physical, *etc.*)
Business Integration Reference Architecture

Services to Solve Complex Business Requirements

Model, design, development, test tools

Common Runtime Infrastructure

Monitoring Services

User Interaction Services
Application Services
Information Services
Process Services
Community Integration Services

Enterprise Service Bus

Application Access Services
Data Access Services
Understanding Services

- What’s a Service?
- Where’s the Service?
- Who defines Services?
- What standards are they built on?
- Who “owns” the Services
- Who manages a Service?
- How are they used?
- Who uses them for what?
- Where does integration take place?

To help answer some of these questions the US Government is investigating the use of patterns in its SOA efforts.
What Is a Pattern?

A Pattern

- Solves an important recurrent problem
- Has a name
- Is an artifact of design reuse

Abbreviated Pattern Form

- Problem Context: Situation giving rise to a design problem
- Solution: A description of the technical solution
- Consequences: Benefits and limitations of the solution
What Is a Pattern?

- A Pattern
  - Solves an important recurrent problem
  - Has a name
  - Is an artifact of design reuse

- Abbreviated Pattern Form
  - Problem Context: Situation giving rise to a design problem
  - Solution:
  - Consequences: Benefits and limitations of the solution
Patterns and SOA

- Next 6 slides courtesy of MITRE Corporation
  - (special thanks to Ray Modeen from MITRE for providing the slides)
- Taken from a joint IBM / MITRE study
- Participants:
  - MITRE
  - IBM Research (John Vlissides and others)
  - IBM SWG, Office of the CTO (Paul Giangarra)
- Sponsor: US Air Force (Hanscom AF Base)
- Goal: understand how to apply patterns to SOA in architecture and design of future DoD systems
Node Information Services
Design Patterns for the C2 Enterprise

Is your program ready for NetCentric Warfare?
NetCentric Warfare is coming!

• Programs will need to deal with:
  – Service Design
  – Technology and vendor choices
  – Performance and reliability issues
  – Cross program dependencies
  – Migration from legacy platforms
  – Evolving DoD Infrastructure (NCES)

• Are you ready?
Background

How to Define Net-centric Service Relationships

Designing Node Information Services (NIS) with Patterns

Tensions, Tradeoffs, Resolution

Common Integrated Infrastructure

Node Platform

Node Platform

APP

APP

APP

APP

APP

APP
We’ve applied patterns to NetCentric Warfare

• We have an evolving library of patterns for designing services:
  – Basic Web Services
  – Publish/Subscribe
  – Security Guard
  – Enterprise Query
• Pattern workbench tool and process
• Currently working with key programs to develop NetCentric solutions
• How can we help your program?
Pattern-Based Service Design can help

- Design Patterns:
  - Relate war-fighter problems to technical solutions
  - Express benefits and limitations of a solution
  - Enable programs to reuse proven NetCentric designs
Example Pattern

Name: Brokeder Notification (Pub/Sub)
Type: Service-Oriented Architecture

Intent
Provide a structure for a community of interest to share information.

Context
A community of interest has three basic criteria: (1) a topic or set of topics, (2) a set of information publishers and (3) a set of information consumers.

The community of interest needs a mechanism for managing and updating a topic list as well as subscribing to the topics themselves. The community requires users to bring information to the community and users who benefit from the information provided.

Consequences
Benefits
Anonymous Access
Both the publisher and subscriber can access the system anonymously. This allows them to interact with the system while allowing the system to maintain security and the relevant validity and accuracy of the information. Therefore, both the publisher and subscriber access their area of responsibility without the burden of validating the information outside their sphere of control.

Broker Aggregator
Brokers can create new events. Brokers can combine events, aggregate from multiple sources or do complex event processing prior to generating a notification.

Limitations
Over subscription
Security is required to prevent a denial of service in an over subscription environment. Third-party subscriber can potentially over-subscribe a consumer. Third-party subscribers could also create security problems by subscribing unauthorized consumer.

Topic List Subscriptions
Offers a manageable topic list, there is nothing preventing a consumer/subscriber from subscribing in a topic list. This way, new topics are immediately sent to consumer/subscribers.

Implementation
WS-Notification (2004, schema, SOAP, WSDL) to be available from various OASIS implementations.

Related Patterns
One-way SOAP, One-way Broadcast, WS-Security

References
WS-Notification, Enterprise Integration Patterns, Holpe and Wolf
Example Application: Guarded Correlation Pattern

**Legacy Combat Support System**

**Legacy Combat Support System**

**Legacy Combat Support System**

**Low**

- Transformation
- Publish/Subscribe
- Monitoring
- Security
- QoS

**Low Side Enterprise Services Bus**

**Routing**

**Events**

**Text Guard**

**High**

- Transformation
- Publish/Subscribe
- Monitoring
- Security
- QoS

**High Side Enterprise Services Bus**

**Routing**

**Events**

**Text Guard**

**Service Integration and Orchestration**

- Process Choreography
- Common Information Model
- Message Transformation
- Common Store

**Legacy C2 System**

**Legacy C2 System**

**Legacy C2 System**
Challenges when Designing for SOA

- Complexity should decrease, not increase one more time
  - At least for the service modeler/developer
  - Exploit tools and runtime environments to hide complexity
- A pure use case-centric approach is not enough
  - Otherwise reuse (or co-use) remains a vision
- In the past, Business Process Modelling (BPM) was typically disconnected from architecture and Object-Oriented Analysis/Design (OOAD) efforts
  - Proprietary notations, no end-to-end tool support
- The additional SOA principles cause new challenges
  - Process choreography (exploiting standards like BPEL)
  - Roles for perform the modeling – analyst/architect/developer
Some Core Principles, Practices, Approaches

- Create/Use a reference Enterprise Architecture

- Base it on:
  - Service Oriented Architecture
  - Open Standards based COTS wherever possible
  - Scalable, reliable, secure infrastructure principles

- Exploit patterns from existing private sector work
  - (e.g. Patterns for e-Business)

- Develop specific public sector patterns

- Use modern development tools to simplify Business Integration projects and development
Paul’s Theories of SOA Approaches

1. 500 Pound Gorilla
2. Who’s the Boss?
3. Diplomacy and Policy

- Which applies best to public sector?
  - Answer: probably #3; but not always!
- Which applies best to private sector?
  - Answer: depends...

So now let’s see how SOA is being applied...
From the e-Government Initiative in the US

The President’s Management Agenda identifies expanded e-Government as key to ensuring that the Federal Government is well-run, citizen-centered and results-oriented

- In that agenda Expanded Electronic Government was defined as:
  - Simplify and unify around citizen needs
  - Support projects that offer performance gains across agency boundaries
  - Maximize interoperability and minimize redundancy

Expanding E-Government enables the Federal Government move from agency-centered to citizen-centered

- The Vision
  - An order of magnitude improvement in the Federal Government’s value to the citizen, with decisions in minutes or hours, rather than weeks or months

- E-Government Definition
  - The use of digital technologies to transform government operations in order to improve effectiveness, efficiency and service delivery

- The Principles
  - Citizen-centered, results-oriented, market-based
  - Integral component of the President’s Management Agenda
  - Simplify and unify
What’s Stopping Us? Public Sector Inhibitors (1)

- Established “fiefdoms” / stovepipes / SPOs difficult to “break”
- Government contracting practices and regulations
- Coalitions (multi-government cooperation/participation)
- Lack of Public Sector Interoperability Standards
  - Just XML is not enough
  - OASIS e-Government TC (a start)
- Then comes getting agreement on definition of standards and common core infrastructure, who owns it, who builds, and who runs it
- Unique security constraints
  - Multi Level Security
  - Compartmentalized vs. Aggregated Information
  - Privacy Laws vs. “Freedom of Information”
  - Conflicting Security Policies of various Departments and Agencies
  - My data, your data, never the twain shall meet (emotional?)
- Skilled [cleared] developers
What’s Stopping Us? Public Sector Inhibitors (2)

- Heavy use of Custom/Bespoke solutions vs. COTS product based solutions
- Cost constraints
- Availability of skills (especially at prices public sector can afford)
- Responsibilities between Local and Central Government often not well defined

Environmental Inhibitors:
  - Need to deal with emergencies, adverse conditions (e.g. floods, terrorism)
  - Non-deterministic communications often the norm
  - “Standard” Disaster Recovery scenario is not a temporary site outage

Large Enterprise Architecture efforts based on old technologies (e.g. IDEF, structured analysis) without taking into account the “to be” requirements
Is It Really This Complex?

yes but…
Back to Basics: Core Architectural Principles of a SOA

Infrastructure Services

- Systems and Policy Management
  - Security
  - Availability
  - Provisioning
  - Optimization
  - ... 

Resource Virtualization

- Servers
- Storage
- Network
- Other Resources

Enterprise Services Bus

- Mediation
- Transformation
- Publish/Subscribe
- Monitoring
  - QoS
  - Security

Derived from an EA Whitepaper Written for US Customs and Border Protection
Back to Basics:  Core Architectural Principles of a SOA

Derived from an EA Whitepaper Written for US Customs and Border Protection
[DoD] View of Service-Oriented Architecture Evolution

(Source: work done by United States Air Force, AF-ESC, Hanscom AF Base)

Web Services

Enterprise Infrastructure

Component Orchestration

Semantic Web

Complex Event Processing

Standards-based info management framework

Distributed collaborative processing with discovery

Orchestration of C4ISR components

Intelligent M2M collaboration

Warfighter events pattern recognition
Paul’s Vision for e-Government (a “Citizen’s” View)

- It starts with one and only one portal / entry into e-Government
- It probably starts where I live (local government)
- It incorporates state and central government services when needed because …
- It is personalized to my needs however …
- It is non-obtrusive but at the same time …
- Helpful and easy to use

Government then appears as a collection of joined up, tightly Integrated, personalized services, built just for me!
SOA Readiness Assessment

A structured approach for assessing the SOA environment in terms of stated goals (e.g., extensibility, flexibility, integration)

The **component structure** assessment category is used to assess the underlying components and/or component architecture that comprise the services portfolio under review.

The **services portfolio** assessment category is used to assess the current or planned or current business services that make up the services layer in the Web Services Architecture.

The **infrastructure** assessment category is used to assess the current or planned operational architecture.

The **management process** category is used to assess the describes the distinct processes necessary to meet the desired quality of service attributes for Web Services.
SOA Readiness Assessment

A structured approach for assessing the SOA environment in terms of stated goals (e.g., extensibility, flexibility, integration)

The **processes** category is used to assess the distinct processes necessary to meet the desired quality of service attributes for SOA.

The **infrastructure** assessment category is used to assess the current or planned operational architecture.

The **applications** assessment category is used to assess the underlying components and/or component architecture that comprise the services portfolio under review.

The **services architecture** assessment category is used to assess the current or planned or current business services that make up the services layer in the SOA.
Example SOA Assessment Results
The Path to Network Centric Warfare/Enterprise

You can't get to a Network Centric Enterprise without a common messaging bus! The dependency diagram looks pretty much like this:

Worse yet, you can't get to a Network Centric Enterprise without selecting an integration platform, and the longer you wait the more difficult it's going to be.
Today: Movement Toward Standards

However...
Standards Issues Have Been Around for a Long Time ...

- Sunday, February 7th, 1904:
  Baltimore, Maryland, 10:48 a.m.
  - Engine Company Number 15 receives a fire alarm
  - Quickly escalates to 1,231 firefighters, 57 engines, 9 trucks
  - Fire companies from Washington D.C., New York, Philadelphia, Wilmington, Delaware, Atlantic City and Pennsylvania rush to assist
  - Fire crews were unable to assist
  - Equipment stood idle while the fire raged
  - Why? The out of town hose couplings would not fit the Baltimore fire hydrants

- By Monday, February 8th, 17:00
  - 140 acres (70 blocks) devastated
  - 1,526 buildings were destroyed
  - 2,500 businesses impacted
  - 35,000 people left jobless
Standards Issues Have Been Around for a Long Time ...

- In 1904, there was no national standard for House couplings and fire hydrants
- NIST collected 600 DIFFERENT hose couplings in usage and developed the national standard.
- In early 1900s, there were no standards for:
  - Shoe sizes
  - National weight standards, even for railroad cargo
  - Electrical standards
... and Continue Through Today

- Southern California Wild Fire, October thru early November 2003
  - 739,597 acres burned
  - 3,631 homes destroyed
  - 24 people dead
  - 15,600 Firefighters deployed
  - Damage claims estimated at $2.03 billion

- California is implementing a Standardized Emergency Mgmt System (SB 1841) as the framework for response
  - Incident command
  - Multi-agency coordination
  - Mutual aid

“Sometimes the major obstacles, the political turf, are far greater than the technology, to overcome the human and institutional limitations”
-Glen Craig, Exec Director
CA Alliance for Public Safety Communications
### SOA Open Standards Timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998 / 1999</td>
<td>Java, XML and ebXML</td>
</tr>
<tr>
<td></td>
<td>Co-founder and lead architect for RosettaNet</td>
</tr>
<tr>
<td></td>
<td>Author of XML4J</td>
</tr>
<tr>
<td></td>
<td>Chair OMG XML Metadata Interch. Format</td>
</tr>
<tr>
<td></td>
<td>Co-author W3C Document Object Model</td>
</tr>
<tr>
<td></td>
<td>FounderXML.org</td>
</tr>
<tr>
<td></td>
<td>Elected to Board of Directors in OASIS</td>
</tr>
<tr>
<td>2000</td>
<td>Web Services and UDDI</td>
</tr>
<tr>
<td></td>
<td>Co-author of SOAP 1.1 and submission to W3C</td>
</tr>
<tr>
<td></td>
<td>Co-founder of UDDI.org and author of original UDDI specification</td>
</tr>
<tr>
<td></td>
<td>Co-author of WSDL</td>
</tr>
<tr>
<td></td>
<td>IBM contributes SOAP4J to Apache</td>
</tr>
<tr>
<td>2001</td>
<td>Web Services and Tools</td>
</tr>
<tr>
<td></td>
<td>Led submission of WSDL to the W3C</td>
</tr>
<tr>
<td></td>
<td>Co-chaired W3 Web Services Workshop</td>
</tr>
<tr>
<td></td>
<td>Founder of Eclipse.org</td>
</tr>
<tr>
<td></td>
<td>Co-author of W3C XML Schema standard</td>
</tr>
<tr>
<td></td>
<td>Chair of Web Services Interactive Applications TC</td>
</tr>
<tr>
<td>2002</td>
<td>Web Services and Security</td>
</tr>
<tr>
<td></td>
<td>Founder and chair, Web Services Interoperability Organization</td>
</tr>
<tr>
<td></td>
<td>Co-author of web services bus process specification (BPEL, WS-TX, WS-TC)</td>
</tr>
<tr>
<td>2003</td>
<td>Web Services and Security</td>
</tr>
<tr>
<td></td>
<td>Submission of BPEL to OASIS</td>
</tr>
<tr>
<td></td>
<td>Co-chair WSBPELTC in OASIS</td>
</tr>
<tr>
<td></td>
<td>Submission of Common Base Events to WSDM TC in OASIS</td>
</tr>
<tr>
<td></td>
<td>Submission of WS-Manageability to WSDM TC in OASIS</td>
</tr>
<tr>
<td></td>
<td>Co-authored and published EPAL specification to WC3</td>
</tr>
<tr>
<td></td>
<td>Co-chair WSDM TC in OASIS</td>
</tr>
<tr>
<td></td>
<td>More than 1,000 developers devoted to XML and more than 1,500 focused on Linux. Over 200 software products running on Linux</td>
</tr>
<tr>
<td></td>
<td>Led workgroup responsible for finalization of SOAP 1.2</td>
</tr>
</tbody>
</table>
Championing Horizontal Integration Standards

1998/1999 **Java®, XML and ebXML**
- Cofounder and lead architect for RosettaNet
- Author of XML4J
- Chair OMG XML Metadata Interch. Format
- Coauthor W3C Document Object Model
- Founder XML.org
- Elected to board of directors in OASIS

2000 **Web services and UDDI**
- Coauthor of SOAP 1.1 and submission to W3C
- Cofounder of UDDI.org and author of original UDDI specification
- Coauthor of WSDL
- IBM contributes SOAP4J to Apache

2001 **Web services and tools**
- Led submission of WSDL to the W3C
- Cochaired W3C Web Services Workshop
- Founder of Eclipse.org
- Coauthor of W3C XML Schema standard
- Chair of Web Services Interactive Applications TC

2002 **Web services and security**
- Founder and chair, Web Services Interoperability Organization
- Coauthor of Web services business process specification (BPEL, WS-TX, WS-TC)
- Coauthor for Web services security roadmap and specification

2003 **Web services and security**
- Submission of BPEL to OASIS
- Cochair WSBPELTC in OASIS
- Submission of Common Base Events and WS-Manageability to WSDM TC in OASIS
- Coauthored and published EPAL specification to W3C
- Cochair WSDM TC in OASIS
- More than 1,000 developers devoted to XML and more than 1,500 focused on Linux. Over 200 software products running on Linux
- Led workgroup responsible for finalization of SOAP 1.2

2004

<table>
<thead>
<tr>
<th>1998/1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Java®, XML and ebXML</strong></td>
<td><strong>Web services and UDDI</strong></td>
<td><strong>Web services and tools</strong></td>
<td><strong>Web services and security</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cofounder and lead architect for RosettaNet</td>
<td>Cofounder and lead architect for RosettaNet</td>
<td>Chair OMG XML Metadata Interch. Format</td>
<td>Chair OMG XML Metadata Interch. Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author of XML4J</td>
<td>Author of XML4J</td>
<td>Chair OMG XML Metadata Interch. Format</td>
<td>Chair OMG XML Metadata Interch. Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coauthor W3C Document Object Model</td>
<td>Coauthor W3C Document Object Model</td>
<td>Founder XML.org</td>
<td>Founder XML.org</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Founder XML.org</td>
<td>Founder XML.org</td>
<td>Elected to board of directors in OASIS</td>
<td>Elected to board of directors in OASIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elected to board of directors in OASIS</td>
<td>Elected to board of directors in OASIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1998/1999 2000 2001 2002 2003 2004
Standards Support-Today: /...Tomorrow /...Future

- Interoperability standards:
  - SOAP
  - WSDL
  - UDDI
  - XSLT

- J2EE standards
  - JAX-RPC
  - JSR 109 – Web Services for J2EE
  - JAXR

- Proposed interoperability standards

- Used within the ESB
  - WS-Security etc.
  - WS-Addressing
  - WS-Policy
  - WS-ReliableMessaging
  - WS-Notification
  - WS-Resource*

- Using the ESB
  - WS-Coordination
  - WS-AtomicTransaction
  - BPEL4WS
  - WS-BusinessActivity
Open Standards/Technologies I Consider & Watch

- **Presentation**
  - Portals, Portlets: JSR 168, WSRP
  - J2EE (Web Containers, Servlet, JSP, JSF)
  - Synchronous and Asynchronous Advanced Collaboration
  - Knowledge Management
  - Pervasive Computing, WML

- **Integration**
  - J2EE (EJBs, JMS, JTA, JCA, ...)
  - SDO (and data mediation)
  - SOA
  - Web Services Stack (SOAP++)
  - BPEL4WS

- **Enterprise Service Bus**
  - WS-Notification
  - WSRF
  - CEP, EDA
  - Mediation (of messages)

- **Infrastructure**
  - J2EE (Containers, Container Services, ...)
  - Virtualization
  - XML, Web Services
  - Autonomic Computing
  - Semantic Web
  - Grid Computing
  - Security: Privacy, Identity, Access Control, MLS

- **Data**
  - Data and Information Federation
  - Search
  - Metadata
  - Content/Records Management
  - Data Mining, Business Intelligence
  - Digital Rights Management

- **Other**
  - Open Source: Linux, Eclipse, Grid Computing, Apache, ...
Industry Adoption of Open Standards

- Operating System
  - Public Sector
    - Middleware
      - Database
      - Systems Mgt.
  - Operating System POSIX
- Hardware
  - Model Driven
- Applications
  - System of Systems
- Middleware
  - CORBA
  - COM
  - XML
  - Web
  - Database
  - Systems Mgt.
- Middleware
  - J2EE
  - BPEL
  - Simulation
  - Integration
  - Security
- Middleware
  - SOA
  - Enterprise
  - Service Bus
- Middleware
  - Virtualization
  - Grid
- Middleware
  - Modeling
- Applications
  - System of Systems
  - Model Driven
- Applications
- Applications
- Applications

- LOW
- INTEROPERABILITY & FLEXIBILITY
- HIGH

- 1970s
- 1980s
- 1990
- 1997
- 2003
- 2005
- 2006...

- client / server
- network
- on demand
Semantic Web (from w3c.org)

- The **Semantic Web** provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (**RDF**), which integrates a variety of applications using XML for syntax and URIs for naming.

- "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." – *Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001*

- So what does this all mean....
Semantic Web

- Built upon proven Web technologies
  - Extensible Markup Language (XML)
  - Universal character set (Unicode)
  - Universal addressing (Uniform Resource Identifiers)
  - Standard World Wide Web protocols (e.g., HTTP)
- These Web technologies are used widely (in Web services, *etc.*)
- Semantic Web languages build upon XML’s ability to define customized tagging schemes*
  - XML: Defines syntax; imposes no semantic constraints
  - XML Schema: a language for restricting the structure of XML documents

Semantic Web Includes

- A growing family of RDF-based specifications.
- RDF graph data model: freeform yet formal
- RDF schema: a basic vocabulary definition language
- OWL, the Web Ontology Language: formal vocabulary semantics
- RDF Data Access: a query language and Web service protocol for RDF [new work]
- ...moving into deployment phase, focus shifting from technology to supporting activities
Semantic Web (Continued)

- 10 Feb ‘04 W3C released recommendations on
  - Resource Description Framework (RDF)
    - Used to represent information and to exchange knowledge in the Web
  - OWL Web Ontology Language (OWL) as W3C
    - Used to publish and share sets of terms called ontologies, supporting advanced Web search, software agents and knowledge management
  - See http://www.w3.org/ for more information
- RDF and OWL are now international standards
- Both RDF and OWL observe the Open World Assumption: new knowledge can always be added to what already exists
Features of RDF and OWL

RDF Features
- Rules to allow for decentralized extensions
- Descriptive rather than prescriptive (contrast with XML)
- RDF vocabularies document claims about the world (not about documents)
- RDF is designed for data merging (easy when things have agreed IDs)

OWL Features
- OWL extends our vocabulary description, allowing us to express claims such as...
- Nothing can be both a Document and a Person
- grandParent and grandChild are inverses
- homepage, nasdaqCode and mbox are uniquely identifying properties
- A W3CTeamPerson is a Person whose workplaceHomepage is http://www.w3.org/
- ...as well as the formally specified interactions amongst all these features.
Semantic Web Stack

From: Semantic Web Ivan Herman, Head of Offices  Date:  2004/05/28 10:34:34
http://www.w3.org/Consortium/Offices/Presentations/SW_Advanced/Overview.html
What Is a “Complex Event”?

- An event that can only happen if some sequence and combination of other (specific) events has already happened.

Useful shorthand:

- an event, and particularly a complex event, is represented by an “object” that is instantiated when all of the predecessor events have occurred (and that occurrence has been recognized), *i.e.*, when we talk about the event, we are referring to the object that results from the recognition of a defined set of predecessor events.
Aspects of an Event

An event has:

- a name,
- a unique identifier,
- a time (or times, *e.g.*, start and stop),
- a causality (list of the predecessor events),
- form (attributes relative to the event),
- significance (description of the activity the event signifies), and
- relativity (the relationship between this event and other events).

It is essential that the event carry the identification of the events that caused it (if possible); this is what allows “drill down” inspection of the operation of a system.
Recognizing a Complex Event

A complex event occurs when some sequence and combination of other events occurs. To recognize such an event, a system must be able to:

- Identify a sequence of event occurrences, even if the events are received out of order (and other events are interspersed), i.e., events occur in a *cloud* rather than a stream.
- Filter relevant events based on form (attributes), time, causality, and relativity.

When the entire set of conditions is satisfied, then the event is said to have “occurred” and an event object is instantiated. This is an aggregate event object, because it encapsulates multiple causal events.

- Oops: The uncertainty principle – An activity in a target system may not have any observable signifying event.
What Does It Take to Do Complex Event Processing?

1. Capability to examine events as they occur (messages, database calls, or specific event notification). Must also be able to recognize that an event, or pattern of events, didn’t occur.
2. A (robust) language in which to define event patterns.
3. A rules language (with sufficient power of expression) to connect the event patterns, examine event attributes, and test event attributes against external data.
4. A pattern matcher (that is fast enough to keep up with the flow of events).
5. A rules engine (that allows rules to be changed in real time).
6. A constraint checker (rules that define boundary conditions).
7. A repository for causal event execution, \(i.e.,\) a place that stores the patterns, the rules, the constraints, and (when appropriate) the resulting event objects.
8. Some form of hierarchical execution analyzer (a “drill down” tool to trace the causes of events).
9. A “viewer” that displays summaries of groups of events, according to a set of rules, \(i.e.,\) a system monitor.
Why Do We Care about CEP?

- Global information spaghetti...
- The interaction of distributed information systems makes it difficult to identify or track the causes of events.
- There is essential management information in the pattern of events (or the lack of events).
- Applications are generally not instrumented to provide event notification. If they were, it would probably be the wrong information.
- Most of the real monitoring functionality that we have is at the network level, and most of that is reactive and “after the fact.”
- Events of significance are mostly inferred from aggregated sets of other events.
- Human monitoring, including recognition of constraint violations, must occur in near real time to be useful.
- Agile systems, which respond rapidly to changes in their environment, need a constant flow of information to understand the environment, including whether a portion of it has been destroyed.
An Old Wives Tale ...

Once adopted, Standards tend to stick and can have unintended consequences
Summary

- We’ve discussed:
  - Where we are today (Swivel Chair Integration)
  - Core Concepts of SOA and ESB
  - Ways to move forward
  - Architecture, Design & Development ideas
  - Experiences
  - Emerging Technologies (*e.g.* Semantic Web, CEP, and more)
- Horizontal “Interoperability” and Future Flexibility are the primary commercial IT metrics for Total Cost of Ownership (TCO)
- High-value business transformations are best executable with a strategic approach to information technology
- A powerful technical/business ecosystem has emerged based on open standards and independent investment
- SOA is real, important, and critical to Enterprise Transformation efforts
Thank You

Questions?
## Links

<table>
<thead>
<tr>
<th>Service</th>
<th>URL</th>
</tr>
</thead>
</table>
More Info on Semantic Web

- Semantic Web Home Page: http://www.w3.org/2001/sw/
- Semantic Web Advanced Development Home Page: http://www.w3.org/2000/01/sw/
- RDF Home Page: http://www.w3.org/rdf/
- Semantic Web / RDF Interest Group: http://www.w3.org/RDF/Interest
- Semantic Web / RDF IRC: irc://openprojects.net/#rdfig