An Overview of Total Architecture
BPM and SOA In Practice

Paul C. Brown, Principal Software Architect

© 2009 TIBCO Software Inc. All Rights Reserved.
In the Beginning, Architecture was Simple…

... and evolved slowly
Transport Technology Helped Communities to Emerge…

... and grow
Better Infrastructure Fostered Denser Communities…

... and more of them
So How Do You Organize and Manage All This?

- How do you ensure you get the business results you want?
  - The desired business benefit
  - Within cost constraints
  - While preserving the flexibility to address tomorrow’s needs
Enterprise Complexity
The Idealized Enterprise View Looks Simple

- A functional organization with well-defined responsibilities...

![Diagram of an idealized enterprise view with hexagons labeled as follows: Customers, Sales, Human Resources, Engineering, Manufacturing, Marketing, Purchasing, Logistics, Finance, Suppliers, Shipping Partners.](image-url)
But There Is a Lot of Dialog Between the Organizations

☐ How do you make sense of this?
You Think in Terms of Business Processes

This picture does not tell you how the order-to-cash process actually works.

Order-to-Cash Process Scope
Business Process Models Provide That Understanding

- Activities and their structure
- Participants
  - Swimlanes represent roles
  - Activities in the lanes represent responsibilities
- Interactions between participants
  - Artifacts
    - Messages
    - Physical objects
  - Relationships to activities
- Interactions with other business processes
  - Where does the product catalog come from?
You Must Think About Information as Well

- Understanding utilization scope tells you little about the information itself.

**Sales Order Information Usage**
Logical Data Models Provide That Understanding

But they don’t characterize:

- Who owns the data
  - Organization
  - System
- Where the data physically lives
- Where data is replicated
  - How consistency is maintained
Each “Organization” is Actually a Stack

There may be multiple components at each layer:

- Interfaces
- Logic components
- Data components
- Infrastructure

Traditional three-tier structure
Traditional Means of Supporting Organizational Interaction

- **Conversation**
  - Human level

- **EAI**
  - Logic level

- **ETL**
  - Data level
People May Interact with Multiple Systems

- EAI and ETL can be used within an organization as well
Overall, Understanding Interactions is Complicated

This is the problem that SOA and BPM are supposed to solve!
The Concept of Total Architecture
The Enterprise is an Operation

- Enterprise operations transform inputs into outputs using resources (employees, systems, facilities)

<table>
<thead>
<tr>
<th>Enterprise Operations</th>
<th>customers</th>
<th>customer orders</th>
<th>customer payments</th>
<th>suppliers</th>
<th>materials</th>
<th>supplier invoices</th>
<th>employees</th>
<th>systems</th>
<th>facilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>goods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shipping notices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer invoices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>purchase orders</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>supplier payments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enterprise Operations are Comprised of More Specialized Operations

- Operations interact with one another via artifacts

Your enterprise is a network of interacting operations
Operations Execute via One or More Business Processes

- Financial Operations
  - customer order
  - customer invoices
  - customer payments
  - shipping notices

: Accounts Receivable Process

- purchase orders
- supplier invoices
- supplier payments

: Accounts Payable Process

- Viewed from the operation’s perspective, the business process appears to be local
In Reality, Processes Interact with Other Processes
Changes Commonly Impact Multiple Operations

- **Artifact changes** alter both producing and consuming operations and their systems

- **Participant changes** (human or system) impact mechanisms for producing and consuming artifacts

- **Process structure changes** can impact the sequencing of interactions with other operations

- Speeding up a process may also require speeding up related processes in other organizations
Changes Impact Both Business Process Design and Systems Architecture

Changing who plays a role (person or system)

Introducing intermediaries (services) to isolate parties from changes

change role 1
role 1: Person
role 1: System

change role 2
role 2: Person
role 2: System

introduce an intermediary
role 1: Person
role 1: System

service: System

role 2: Person
role 2: System
Changing a ProductID format can impact many processes

- Remember the 5-digit zip code?
For Success, A Total Architecture Perspective is Required

- **Business Processes**
  - Sales order management
  - Inventory management
  - Accounting

- **People**
  - Participants in the business processes

- **Information**
  - What information is being used

- **Systems**
  - Computers, networks, applications, infrastructure

- **Business Purpose**
TIBCO BPM/SOA Execution Model

**Execution Strategy**

1. **Step 1**: Develop Vision & Program Roadmap
2. **Step 2**: Define & Implement Organization & Governance
3. **Step 3**: Define & Implement Technical Infrastructure & Standards
4. **Step 4**: Analyze Process & Develop Project Roadmap
5. **Step 5**: Repeat for each project
6. **Step 6**: Operate the Business

**Solutions & Operations**

- Continuous Improvement
  - Governance
  - Project Life Cycle Management and Control
  - Measure Business, IT and Organizational KPIs & SLAs / Analyze ROI

© 2009 TIBCO Software Inc. All Rights Reserved..
Focus on business processes first
- They are the source of business value
- They are the glue that binds people and systems together

Figure out what it is before you try to improve it!
SOA and BPM refine the traditional structure
- Introduce a separation between processes and services

The business process is a first-class architectural concept
- Their structure needs to be aligned with both organizations and systems
- Processes interact with one another just like systems do!

Processes are assembled from services
- Some performed by people, others by systems
- Ideally, each service gets used in multiple business processes
- Separate service access mediation from services
  - Service access control based on
    - Security
    - Quality-of-service agreements
  - Routing of service requests
  - Load distribution across multiple service providers
  - Logging of service utilization
  - Performance and SLA measurements
- Acknowledge different types of processes

- **Unmanaged Processes**
  - Components/services are “hard wired” together to form the actual process
  - One component’s results become the inputs to the next component
  - Proactive monitoring and breakdown recovery is required for high availability

- **Managed/Orchestrated Processes**
  - One component coordinates the work of other components and services
  - Manager can monitor process status
  - *The process of starting the manager is always an unmanaged process!*
Managed Processes Often Span Multiple Systems

- Process manager may need to access data (or non-service functionality) directly when it is not available as a service.
Packaged Application Workflow

- Generally assumes that the activities being managed are:
  - Executed via the application interfaces
  - Implemented within the application

- Facilities for managing external activities are generally limited
  - Lack full EAI suite required for convenient management

- User interface is generally not extensible
  - Difficult to add new/different interfaces
Augmented Packaged Application Workflow

- Uses EAI and ETL to coordinate with activities in independent applications
- No overall process manager
  - Part of process is “hard wired”
  - Defeats purpose of using workflow
- Designed to coordinate activities no matter where they are
- Able to access services where they are available
- Able to leverage EAI technologies to access non-service functionality
- Truly separate interface from process
Hard-Wired Processes May Involve Any Technology

- Often hybrids of SOA, BPM, and older technologies
  - EAI and ETL remain important
  - EAI provides the “events” for an event-driven process

- Note the absence of a manager!
SLAs require monitoring and critical processes require breakdown detection

Complex event processing (CEP) provides the basis for this type of monitoring

- Event recognition/capture remain challenges

Master data management has a similar distributed architecture

- Blend of ETL, EAI (events), and workflow to keep data consistent
Events are What You Can Observe

- Providing meaning to them requires associating them with:
  - Processes
  - Data
  - History
Complex Event Processing (CEP) Provides the Tools

- Concept models provide the data context
- State models represent process milestones
- Event history allows changes to be identified
- Generated events announce conclusions
Separate processes and presentation

- Sometimes you want to make the same process available via different channels
  - Avoid duplicating the business rules
- Some of the channels may not be conventional presentations
  - May provide web-service access
- In such cases, you want to make the process itself into a service
  - Accessible from a variety of presentation components
- Embrace total architecture
  - SOA and BPM provide many opportunities to organize and manage the enterprise

- The challenges are organizational as well as technical
Services
A service is a reusable unit of functionality with standardized interfaces.

- Native semantics for operation and data
- Native technology for operation and data

Some level of standardization:
- Technology of access
- Data semantics
- Operation semantics
Benefits of Services

- **Reuse**
  - Saves money - avoid the cost of replicating functionality
  - Faster time to market for projects - reduced need for development

- **Isolation of service consumer**
  - Service provider design can evolve without impact
  - Service provider can be replaced without impact

- **Flexibility**
  - Platform independence - provide functionality anywhere and access it from anywhere

- **All benefits require service interface stability!**
  - Win occurs when interface is more stable than the functionality on either side
Where Do Services Make Sense?

- When there is functionality that is either used in more than one place or is provided in more than one place, particularly when those “places” are different applications.
Service Granularity

- The amount of work encapsulated by each service operation
  - Coarse grained – encapsulates a relatively large amount of functionality
    - Generate Bank Statements for all current accounts
  - Fine grained – encapsulates a relatively small amount of functionality
    - e.g., Withdraw Cash

- For a service to make sense, the overhead involved in invoking an operation must be less than the amount of work performed by the operation
Composites can directly invoke lower level services

BUT nested service calls may not perform well (particularly retrieving the status information)
  - Underlying services may not be able to handle the load

Performance may be better if the underlying service publishes status information to the composite
There are three starting points for identifying services:
1. Top-down from Business processes
2. Middle-out: from functional perspective
3. Bottom-Up from Data
Withdraw Cash Business Process

- We are going to design the two service operations required to support the Withdraw Cash business process
  - Obtain disbursal authorization
  - Report funds disbursal

Business Processes and the Service Operations they use:

- **Withdraw Cash**
  - : obtain disbursal authorization
  - : report funds disbursal

- **Transfer Funds**
  - : obtain disbursal authorization
  - : report funds disbursal

- **Make Deposit**
  - : deposit funds

The underlying service and its operations:

**Account Balance Management Service**

- Service Operations
  - +obtain disbursal authorization()
  - +report funds disbursal()
  - +deposit funds()
  - +get balance()
Example 1: Withdraw Cash Using Services

<table>
<thead>
<tr>
<th>Customer</th>
<th>ATM System</th>
<th>Bank System</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM Card</td>
<td>PIN</td>
<td></td>
</tr>
<tr>
<td>ATM Machine</td>
<td>ATM Server</td>
<td></td>
</tr>
<tr>
<td>select withdraw cash</td>
<td>display transaction choices</td>
<td>cardID - bank association</td>
</tr>
<tr>
<td>enter amount</td>
<td>prompt for amount</td>
<td></td>
</tr>
<tr>
<td>call</td>
<td>obtain disbursal authorization service operation</td>
<td>determine bank and forward request</td>
</tr>
<tr>
<td>: Disbursal Authorization Request</td>
<td>forward response</td>
<td>identify customer, identify account, authorize disbursal</td>
</tr>
<tr>
<td>: Disbursal Authorization Reply</td>
<td>approved?</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>cash</td>
<td>forward acknowledgement</td>
</tr>
<tr>
<td>disburse cash</td>
<td>: Disbursal Report</td>
<td>update account balance</td>
</tr>
<tr>
<td>: Disbursal Report ACK</td>
<td>forward report</td>
<td></td>
</tr>
<tr>
<td>cash</td>
<td>: Disbursal Report ACK</td>
<td></td>
</tr>
<tr>
<td>receipt</td>
<td>: Disbursal Report</td>
<td></td>
</tr>
<tr>
<td>remove receipt</td>
<td>remove cash</td>
<td></td>
</tr>
</tbody>
</table>

© 2009 TIBCO Software Inc. All Rights Reserved..
Proposed Utilization Context

- **Who is providing the service?**
  - Externally provided services
    - Read-only, updating
  - From trusted partners
  - In-house services only
    - Consumed internally
    - Consumed externally

- **How does it fit into the Business Process?**
  - Functional requirements
  - Non-functional requirements
  - Is the service appropriate for use in the different business processes?
    - In particular the granularity of the service
Qualification

☐ Is there more than one utilization context?
  ☐ If not, what is the justification for making this a service?

☐ Does the service fit all the proposed utilization contexts?
  ☐ Granularity
  ☐ Volume
  ☐ Response time

☐ Has the proposed service been reviewed by the stakeholders that will use the service?
Service Mediation

- Managing the interactions between service provider and consumer

- Typical mediation activities
  - Data transport and transformation
  - Service distribution and routing
  - Service access control

- These activities are typically responsibilities of an enterprise service bus (ESB)
Abstracted Location and Access Control

- Abstracting out physical location of the service provider and the access control of the service provider

Service Provider
- +accessControl()

Service Consumer

Proxy
- +accessControl()

Service Provider
- ...

Enterprise Service Bus
- +accessControl()

Service Provider
- ...

SOAP over HTTP
- physicalDestination

SOAP over HTTP
- physicalDestination

SOAP over JMS
- logical destination

SOAP over JMS
- logical destination
Distributed Systems Require Distributed ESB

- Service consumers and providers may be at different geographic locations
- Routing of requests may require introspection of the request message data

Diagram:
- Bank teller application
  - Request for Account Information
  - Routing of request based on location of account
  - Submission of request to service provider
  - ESB Node in North America
    - North American accounts: Customer Account Service
  - ESB Node in Europe
    - European accounts: Customer Account Service
Service producers and consumers using location-independent service definitions
Communications in ActiveMatrix

Node 1

Service Provider in Container

Service Consumer in Container

Policy Agent

Policy Agent

Message Normalization and Routing

EMS

Node 2

Service Provider in Container

Service Consumer in Container

Policy Agent

Policy Agent

Message Normalization and Routing

EMS
Policy Type Summary

- Authentication
- Authorization
- Encryption
- Signature
- Logging
- Tracking
- Auditing
- Routing
- Schema Validation

* Availability of specific policies varies depending upon agent location (proxy, embedded), type of binding, and implementation type.
Policy Independence

- Policies and services are managed independently
Runtime Policy Architecture

Stand-alone Business Works and Policy Agent

Business Works and Policy Agent in an AMX Service Grid Node
The Security Credential Problem

- Different systems require different credentials
- Older systems are not capable of using modern credentials
- Many systems are not designed to pass credentials from input to output
  - Often use “trusted system credentials” to access back-end
Modern front-end systems can pass user credentials through to services (where appropriate)

Policy rules can check credentials at any modern interface

Policy rules can map new credentials to back-end system credentials
Where’s the customer?

- Retail Banking System
  - accountInfo
  - accountHolderInfo

- Mortgage System
  - accountInfo
  - accountHolderInfo

- Credit Card System
  - accountInfo
  - accountHolderInfo

- Investment Banking System
  - accountInfo
  - accountHolderInfo
The Data Warehouse Approach

- Data inconsistencies are never resolved
  - Merge logic becomes increasingly complex

- No home for additional data
  - E.g. relationships between customers

- Only as current as the last ETL extract
The MDM Approach

- MDM includes data maintenance
  - Updates to maintain consistency
  - Workflow to reconcile inconsistencies

- MDM can house additional data
  - E.g. relationships between customers

- ODS required for high-volume query
  - MDM maintains ODS contents
Transactional Data Requires a Separate Path

- Data typically does not require MDM resolution logic
- Frequency of update inappropriate for MDM tooling
- Stored data is often an aggregate/summary of transactional data
  - Implemented with traditional event-driven approach

© 2009 TIBCO Software Inc. All Rights Reserved.
Total Architecture Synthesis
Efficiently Developing Solution Architectures
Positioning Architecture in the Project Life Cycle

Charter Project

Business Benefit, Cost and Schedule Expectations

Define Requirements

Business Process Requirements

Define Business Process Architecture

Business Process Definitions

Define Systems Architecture

Component Structure and Responsibilities

Specify Components and Services

Component and Service Specifications

Implement Components and Services

Unit-tested Components and Services

Assemble and Test System

Working System

Deploy and Use System

Knowledge Gained from Using System

Total Architecture Synthesis Scope
Organizational Issues
Business Processes and Services Cross Organizational Boundaries

Lack of Overall Responsibility

Services and Integrations Span Silos

Shrinking Time Frames

Data Center
- Front-Office Applications
- Application Silo
- Application Silo
- Services, Integration, and Process Management Silo
- Application Silo

Communications and Services Infrastructure

External Applications

© 2009 TIBCO Software Inc. All Rights Reserved.
Many Development Processes Have Become Degenerate

They assume a single system is being worked on
Degenerate Processes Will Not Work for SOA and BPM

Multiple organizations and systems are involved

Requirements → Development → QA → Production
A Richer Development Processes Is Required

- Quantified business benefits
- Cost and schedule constraints
- Business risks

UI & Process Developers
User Groups and Subject Matter Experts
Business Process Policies & Practices Implementers
Who Owns Projects That Span Silos?

Who owns projects that span silos?

Business Executive Sponsor
IT Executive Sponsor

Business Manager
Business Manager
Business Manager
Business Manager
Business Manager
Business Manager

IT System Owner
IT System Owner
IT System Owner
IT System Owner
IT System Owner
IT System Owner

Data Center
Front-Office Applications
Application Silo
Application Silo
Services, Integration, and Process Management Silo
Application Silo
External Applications

Communications Infrastructure
Multi-Silo Projects Include Members from All Silos

3 key leadership roles needed on every project

[Diagram showing various roles and teams across different silos]
The Executive Sponsor Can’t Oversee All These Projects
Enterprise Projects Group Should Manage These Projects

- The group provides a reporting structure for projects that span organizational silos.
Who Provides the Cross-Project Service Perspective?

Who looks ahead for future usages?

Today’s Project

Future Project

Future Project

Service Interface

Call New Service

New Service
The Enterprise Architecture Group Coordinates Projects

- Establishes the vision
  - Ensures projects collectively converge on a single coherent architecture

- Maintains cross-silo perspective at all levels
  - Business
  - Application
  - Infrastructure

- Responsible for:
  - Architecture
  - Standards
  - Best practices
  - Governance
Key Questions

- **Is there an architect on every silo-spanning project?**
  - Responsible for end-to-end business process and systems design

- **How are cross-silo projects managed?**
  - Who negotiates with silos?
  - Who resolves conflicts?

- **Who validates the future applicability of services?**
  - Functionality
  - Granularity
  - SLAs
The Challenges of Silo-Spanning Projects Are Diverse

- Knowledge is scattered throughout the enterprise
- For success, business and IT must align
  - Total architecture focus on producing business value
- New skill sets are required
  - Total (business process and systems) architecture
  - Project management focused on business results
- Clear ownership and control is needed for cross-silo projects
- Executive sponsorship is needed to align priorities
  - Resolve political conflicts
- A Proactive Enterprise Architecture group is required
  - Guide multiple projects towards a cohesive whole
- Governance is essential
Service orientation is a paradigm that frames what you do. Service-oriented architecture (SOA) is a type of architecture that results from applying service orientation.

We have been applying service orientation to help organizations consistently deliver sustainable business value, with increased agility and cost effectiveness, in line with changing business needs.

Through our work we have come to prioritize:

- **Business value** over technical strategy
- **Strategic goals** over project-specific benefits
- **Intrinsic interoperability** over custom integration
- **Shared services** over specific-purpose implementations
- **Flexibility** over optimization
- **Evolutionary refinement** over pursuit of initial perfection

That is, while we value the items on the right, we value the items on the left more.
Guiding Principles: we follow these principles

- Respect the social and power structure of the organization.
- Recognize that SOA ultimately demands change on many levels.
- The scope of SOA adoption can vary. Keep efforts manageable and within meaningful boundaries.
- Products and standards alone will neither give you SOA nor apply the service orientation paradigm for you.
- SOA can be realized through a variety of technologies and standards.
- Establish a uniform set of enterprise standards and policies based on industry, de facto, and community standards.
- Pursue uniformity on the outside while allowing diversity on the inside.
- Identify services through collaboration with business and technology stakeholders.
- Maximize service usage by considering the current and future scope of utilization.
- Verify that services satisfy business requirements and goals.
- Evolve services and their organization in response to real use.
- Separate the different aspects of a system that change at different rates.
- Reduce implicit dependencies and publish all external dependencies to increase robustness and reduce the impact of change.
- At every level of abstraction, organize each service around a cohesive and manageable unit of functionality.
SOA Manifesto Authors

Ali Arsanjani  
Grady Booch  
Toufic Boubez  
Paul C. Brown  
David Chappell  
John deVadoss

Thomas Erl  
Nicolai Josuttis  
Dirk Krafzig  
Mark Little  
Brian Loesgen  
Anne Thomas Manes

Joe McKendrick  
Steve Ross-Talbot  
Stefan Tilkov  
Clemens Utschig-Utschig  
Herbjörn Wilhelmsen
For More Information on Total Architecture…

- Succeeding with SOA
  - The business and organizational perspective
    - For CxOs, managers, architects

- Implementing SOA
  - Creating the total architecture
    - For enterprise and project architects, CTOs

- TIBCO Accelerated Value Framework (AVF)