Enterprise-MDA

Applying Model Driven Architecture (MDA) to Services Oriented Architecture (SOA) to enable the Executable Enterprise
Introductions

Cory Casanave

cory-c@enterprisecomponent.com

Primary author of “CCA” in EDOC
Case Study
U.S. General Services Administration (GSA)

Project: Financial Management Line of Business
Customer: GSA OCIO & CFO
Provider: LMI & Data Access Technologies
Tooling: Component-X, Magicdraw UML & Eclipse

Sections reproduced with the permission of the GSA – George Thomas
“Sea Change”

• Sea of change
  – Get-it-right (Initiative for better acquisition)
  – Merger of FTS/FSS (Major Internal Organizations)
  – Restructuring to provide a unified face to the customer
  – OMB and Congressional mandates and changes of mission
  – Integrating and modernizing financial management
  – Reduction of redundant processes and systems

• Implications
  – Massive organizational change
  – Massive system changes
  – Retraining staff
  – High cost of change
  – Risky and hard to achieve
  – Change combined with current costs and inefficiencies of redundant stovepipe systems is not practical
“Sea Change” Enablers & Cost Reduction

- Enterprise-MDA
  - Executable enterprise architecture based on business goals and processes with automated connection to systems layer
- Service Oriented Architecture (SOA)
  - Architecting the business and technology as services
- Value Chain Analysis (VCA)
  - Analyzing and restructuring business processes based on realized customer value
- Enterprise Service Bus
  - Common infrastructure for SOA
- Combined effect of more automated processes
System + Investment cost over 6 years

Business Advantage Savings Not Included

Note: Representative Numbers    Est. NPV Break Even – About 6 Years
Enterprise Modernization Strategy

- Identify components that will offer greatest ROI
  - Create target executable model
    - We have a baseline with the One-GSA model!
  - Identify system of systems to consider for target
  - Pick an alternative for each;
    - Evolve one or more current systems to support target processes, take on new capabilities and support One-GSA interfaces and/or
    - Harvest one or more systems to build a replacement and/or
    - Integrate functionality into shared services as common components and/or
    - Replace systems or parts of systems that are no longer suitable.
  - Model driven SOA provides the flexibility to mix and match approaches as required. Commonality where possible – diversity where necessary. Evolving over time from integration to common components.
  - End result – architected system of systems
Consolidation into Service Components

- The Good
  - Strategic reduction in operating cost – up to 50%
  - Agile business processes
  - Unification of the enterprise
  - Only way to achieve enterprise transition?

- The Bad
  - Investment in change – As high as 25%
  - Legacy and packaged systems are not componentized

- The Ugly
  - Change is expensive and can be disruptive
  - Current boundaries and ownership change – may require centralized authority and budgeting
  - Requires more “enterprise” agreement – very difficult to get consensus
Strategic Migration

Executable Enterprise Architecture Drives Agile Systems of Systems using Interoperable Components

Systems Composed of Interoperable Components

Standards based integration of Monolithic Systems

Ad Hoc Point to Point Integration of Monolithic Systems

Separate and Non-Interoperable Applications

• Are you here?
Consolidation into Service Components

• The Good
  – Strategic reduction in operating cost – up to 50%
  – Agile business processes
  – Unification of the enterprise
  – Only way to achieve enterprise transition?

• The Bad
  – Investment in change – As high as 25%
  – Legacy and packaged systems are not componentized

• The Ugly
  – Change is expensive and can be disruptive
  – Current boundaries and ownership change – may require centralized authority and budgeting
  – Requires more “enterprise” agreement – very difficult to get consensus
EA Governance Structure

- Initiative
- OMB - 300
- Business Drivers

One GSA Target EA

- Guides
- Refines
- Satisfies
- Specifies

EA Governance
MDA Enhanced Procurement

Current

Order & Requirements

Fund/Contract

Contractor Design Implement Test

Solution

Strategic

Analyze Requirements against or Create BA

MDA Enterprise Architecture

Elaborate Components

Service Component Reuse Library

Fund/Contract Reuse

Generate Adapt Construct

SC Integration Testing

Solution
Enterprise MDA

An approach to realizing executable enterprise architecture with MDA and SOA
Enterprise MDA

• Architecture at the Enterprise Level
  – Systems of systems
  – Collaboration of organizations, systems & people
  – Wide-scale collaborative processes
    • roles and responsibilities
  – Business Service Oriented Architecture
  – Enterprise Components
  – Componentizing functionality – not creating it
  – Executable processes – smooth transition from model to simulation to solution

• Executable Enterprise Architecture
ECA is a “profile of UML”, a way to use UML for a specific purpose - it is an OMG standard
  - That purpose is *modeling enterprise systems*.

You can also think of this as a “modeling framework” for enterprise computing

ECA is part of the “Model Driven Architecture” (MDA) initiative of the OMG
  - Using precise modeling techniques as part of the development lifecycle to speed
development and provide technology independence

ECA has been adopted by the OMG as part of the EDOC Profile for UML specification.
Value Focused Target Architecture

One GSA Target EA

- Business Drivers
- Current Processes
- FAR
- Current Environment
- Initiatives
- Trends

- Business Models
- Workflow
- I.T. Systems Specs
- Collaborative Environment
- Documentation & Training
- Projects
- Time Line Trends Critical Success Factors
Simulated Model Driven Architecture

- Domain Architecture
- Enterprise Architecture Model (CIM)
- ECA Standard “Meta-Model” & UML Profile
- Simulator
- Refine/Iterate
- Live Process Simulation
Automated Model Driven Architecture

Minimize and structure manual implementation

Mapping is tuned to the infrastructure

Tools Produce & Integrate

Enterprise Components

Framework & Infrastructure
(E.G. -J2EE-WS) PSM

Domain Architecture

Enterprise Architecture Model (CIM)

Tools

Infrastructure Mapping
(E.G. J2EE-WS)

Meta-Model UML Profile
(E.G. ECA)
Automated Model Driven Architecture

Meta-Model
UML Profile
(E.G. ECA)

Enterprise Architecture Model (CIM)

Domain

Mapping
Infrastructur
Mapping
(E.G. .NET-WS)

Tools
Produce & Integrate

Multiple and Changing Technology Support

J2EE-WS
Enterprise Components

Framework & Infrastructure
(E.G. -J2EE-WS)
PSM

Simulated
Enterprise Components

Simulation Infrastructure
The Connected Enterprise
Content and Communication

Digital Map
Census Data
Police Records
House Drawings
Aerial Photos

Police Dispatcher Role
Multiple roles in a collaboration
Travel Expense Example

1: travelPermissionRequest
2: travelPermission
3: expenseReport
4: authorizedExpenseReport
5: paymentRequest
UML Collaboration Diagram
Travel Expense Model

Objects --> ClassifierRoles

Ruth (President)

Adam (Chief Accountant)

Paymaster

BookKeeper

John (Cashier)

Authorizer

Elsie (Programmer)

Kim (Methodologist)

Traveler

Joyce (Sales clerk)

Bill (Dispatcher)

Ann (Customer consultant)

Douglas (Marketing manager)
Collaboration Diagram

Traveler

Authorizer

Book Keeper

Paymaster
The Marketplace Example

Mechanics Are Us
Buyer

Status

Physical Delivery

Process Complete

Order

Conformation

GetItThere Freight Shipper

Ship Req

Shipped

Delivered

Acme Industries Seller

January 2006

Copyright © 2006 Data Access Technologies, Inc.
Where are the services?

Mechanics Are Us
Buyer

Web Service

Order
Conformation
Shipped

Web Service

Acme Industries
Seller

Web Service

Ship Req
Shipped
Delivered

Web Service

GetItThere Freight
Shipper

Status

Physical
Delivery
Inside the Seller

Order

Conformation

Shipped

Ship Req

Shipped

Delivered

Order Processing

Shipping

Event

Receivables
Roles to Systems

**Component in Role**

**Interaction Path**
(With Information)

**Interaction**

**Collaboration**

Framework, Middleware & Container

**Role**

Operating System

Net

Hardware
Drilling down – inside a role

• The open domain should make no assumptions about the “inside” of a role.

• Inside one role you *frequently* find more collaborating “parts” of the enterprise - the same model *may* be used.

• Until you get to system inside a managed domain
  - Shared resources (DBMS)
  - Common Management
  - Frequently a legacy system
  - Code
FMEA

• The Financial Management Enterprise Architecture (FMEA) project was sponsored by the US General Services Administration (GSA) Offices of the Chief Information Officer and Chief Financial Officer.

• The project was carried out under the umbrella of the “One GSA” Enterprise Architecture program.

• Project deliverables included:
  – A target business architecture for consistent and comprehensive financial management supporting all GSA services and staff offices.
  – A logical system architecture for a cohesive financial management suite supporting the business architecture, particularly in areas in which a transition needed to be made off legacy systems.
  – A set of interface definitions to act as the basis for a standard GSA financial management service-oriented architecture.
Model Driven Architecture

MDA as defined for GSA Enterprise Architecture work

- Computation Independent Model (CIM)
  - The business model

- Platform Independent Model (PIM)
  - Technology independent system specification
  - Conforms to the business model (CIM)

- Platform Specific Model (PSM)
  - Technology specific (e.g., middleware, application platform, etc.)
    system design
  - Conforms to the system specification (PIM)
Computation Independent Model

- Process Model (EDOC)
  - Business Environment
  - Discipline Roles
  - Enterprise Roles
  - Work Roles
  - Activities and Subactivities

- Information Model (UML)
  - Business Transactions
  - Business Entities
Roles and Collaborations

- **Role**: A specification of the responsibility to perform specific functions in the context of a business process.

- **Collaboration**: A closed set of roles interacting to carry out a business process to achieve some joint purpose.

- **Protocol**: A defined conversation between two roles that may be extended over time (i.e., responses of one party to the other may not be immediate).
Role Decomposition

• **Business Role:** A role played by a business organization in a specific business environment in which business processes emerge collaboratively, rather than by being managed by any single composite entity.

• **Discipline Role:** A role with responsibility for a major business “discipline” within a business organization, independently of the current physical organizational structure of that business.

• **Enterprise Role:** A role with top-level enterprise responsibility for some set of related business services provided within a certain discipline.

• **Work Role:** A role responsible for carrying out one or more business services within an enterprise role, at roughly the level that could be assigned to an individual worker or supported by a specific function in an information system.
GSA Business Environment

Provided protocol

Required protocol

Interaction via protocols

Business role
Focus of FMEA
Financial Management Discipline Role

Protocol representing delegated responsibility for interaction with an entity external to GSA.

Protocol representing interaction with another discipline within GSA.
"Roll-Up" Protocol

The protocols between discipline roles are composites that "roll up" the set of services provided by one discipline to another.

The sub-protocols within a roll-up protocol model specific business services provided by a discipline.
This role may be played by any agency that is authorized to provide Financial Management services to other client agencies.
GSA Financial Management COE

Various One GSA discipline roles may be used to support GSA’s responsibilities as a Financial Management COE, as well as its core mission responsibilities.

GSA plans to use its One GSA processes to provide services as a Financial Management COE.
Financial Management Enterprise Roles (Simplified)

Financial Reporting collects financial data from all other enterprise roles.
Example Enterprise Role

Business service provided by this enterprise role.

Business service this enterprise role requires from another role.

Time triggers for scheduled events.
Example Business Service Protocol

The protocols between enterprise roles model business services provided by one role to others.

The protocol is initiated by a business transaction request.

Responses to the request may indicate success or failure.

Each accepted transaction effects a change in the information and behavior of the receiving role.

Note that, while one role initiates and the other responds to the protocol, information may flow both ways across the protocol.
Example Work Roles

Enterprise role (within the Financial Management discipline)

Enterprise business service

Inter-work role protocol

Work Role
Activities and Choreographies

- **Activity**: A specification of a business function in the context of a role.

- **Choreography**: A specification of the sequencing of external interactions required in order to carry out given business responsibilities.
  - A work role is choreographed in terms of the activities required to perform the business services provided by the work role.
  - A complicated activity may be choreographed in terms of subactivities.
  - A subactivity (or simple activity without subactivity decomposition) is choreographed directly in terms of the event-triggered sequencing of its acceptance of inputs and sending of outputs.
Example Activities
Example Subactivity Choreography

- Input reception
- Output send
- Decision
  Should a recurring receivable be established?
Example Subactivity Requirements

Description: Record a new unfilled customer order, as established via a specific sales instrument.

Generate general ledger transactions to increase Unfilled Customer Orders and decrease Anticipated Reimbursements.

Requirement

RMA-03  Reimbursable agreement information. Capture and accumulate reimbursable agreement information that includes the following:
* Billing limit
* Billing terms
* Customer order amount
* Amount obligated
* Amount expended
* Advances collected
* Advances applied to earned revenue
* Remaining balance on advances
* Amount earned
* Amount billed
* Accounts receivable
* Collections on receivables.

Enable access to reimbursable agreement information by customer ID number, reimbursable agreement number, project, or fund.

JFMIP Core Requirements 2005
A term in the vocabulary represents a **class** of things to be described.

Entities may be described as having a unique **identity**.

**Attributes** specify descriptive information having simple types.

A relation between terms is described by an **association** between classes.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique **identity**.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.
Information Model: What Is It For?

The **process model** describes how business activities are (or are to be) carried out.

The **information model** details the vocabulary of the business entities and transactions used in the process model.

State changes due to the activities

Implicit memory of business information
For example, the billing address for a party being billed may change over time, but the billing address used for a specific bill submission always stays the same.

Note the use of composition associations.

A business transaction represents a snapshot of the information required to carry out a business action.

A business entity represents the current state of information that may change over time.
Three-Tier Component Architecture

Presentation Manager components provide user access to application services.

Service Manager components provide transactional implementation of application services.

Data Manager components persist data between application transactions.
Service-Oriented Component Architecture

Each Work Component in the PIM implements a Work Role from the CIM.

Service Managers implement as system services the business services defined in the CIM.
Example Work Role (from CIM)

Related to Customer Orders

Related to Receivables
Explicit component for scheduling triggers

Explicit cross-transactional coupling via the data tier

Role for human participation in the process
Example Service Manager Implementation

Activity implementation

Interfaces with synchronous operations
Example Activity (from CIM)
An activity implementation must conform to the choreography of the activity it implements. Directly implementing the subactivities makes this manifest.
Example Subactivity Functional Specification

1. **Receive** CustomerOrderEstablishment

2. **Let** newOrder = CreateCustomerOrder(CustomerOrderEstablishment.newOrder).data

3. **Send** GeneralLedgerTransaction to increase Unfilled Customer Orders and decrease Anticipated Reimbursements

4. **Send** newOrder as RecurrentCustomerOrder
   (**Note**: EstablishRecurringReceivables will check if there are actually any creation triggers.)

5. **Send** CustomerOrderEstablished
Example Transaction Information Model (from CIM)
Example Transaction Message Model

“Panoptic” inheritance of “namesakes”

“Restricted” inheritance allows for explicit inclusion and exclusion of attributes and associations
Example Persistence Model

Association indicates a reference to an entity persisted elsewhere.

Aggregation indicates entities persisted in the same data manager.
Realizing a Business Architecture
With SOA Technologies

Roles to Enterprise Components & Services
The primary port type has operations corresponding to the request flows in the protocol.

The callback port type has operations corresponding to the response flows in the protocol.
Example Transaction Message XML Document

```xml
<CustomerOrderEstablishment>
  <customerOrderEstablishment>
    <newOrder>
      <customerOrder>
        <customerOrderID> ... </customerOrderID>
        <customerOrderAmount> ... </customerOrderAmount>
        <orderingCustomer>
          <customer>
            <customerID> ... </customerID>
          </customer>
          <party>
            <name> ... </name>
          </party>
          </orderingCustomer>
        <controllingSalesInstrument>
          <salesInstrumentID> ... </salesInstrumentID>
        </controllingSalesInstrument>
      ... </lineItems>
      </customerOrder>
    </newOrder>
  </customerOrderEstablishment>
</CustomerOrderEstablishment>
```
Business (CIM) view - Collaborating Roles
“Upper” PIM View - Enterprise Component

People, organizations And/or enterprise components play roles in Business Processes.

“Rotate” to look At other aspects of the component
The "Enterprise Digital Assistant"

People, Organizations And systems play roles

Components frequently help people play these roles

People, organizations and systems components work together to realize roles

Components are the peoples Automated assistant

Enterprise components help people and organizations play roles by automating and monitoring The business process

From the system perspective. People and organizations become part of the implementation Of the role
People, Components & Organizations Collaborating
“Lower” PIM View - Enterprise Component Internals

Enterprise Component

UI Client Tier

UI Server Tier

Business Logic

Middleware Wrapper

Enterprise Component

Adaptation

DBMS

Data Managers

Containers

Enterprise [Web] Service

Browser

UI Framework

Legacy Systems

Enterprise Component

January 2006

Page 77
PSM View - Mapping to [web] Services
Mapping of a WSDL Engine

definitions obtained from component /BuySell_Deployment/Seller

```xml
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/
ENC="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xs2000="http://www.w3.org/1999/XMLSchema/
xmlns:xs2001="http://www.w3.org/2001/XMLSchema/
targetNamespace="urn:SellerServer" xmlns:tns="urn:SellerServer" xmlns:CoreTypes="urn:CoreTypes" xmlns:Ordering="uri:Ordering"
- <!--
```
Mapping of an Enterprise Component

- `<service name="MySeller">`
  - <!-- implemented service role
    /BuySell/Deployment/SellerServer/MySeller  -->
  
    `<documentation>`<p> </p>`</documentation>`

- `<port name="BuySellProtocol" binding="tns:BuySellProtocol">`
  - <!-- original service port was
    /BuySell/Deployment/SellerServer/MySeller/BuySellProtocol
    (extending Component
    &lt;/BuySell/SellerImplementation/MySeller/BuySellProtocol&gt; ) -->

    `<soap:address
</port>`

</service>`
Mapping of a protocol binding

```xml
<binding name="BuySellProtocol"
type="tns:BuySellProtocol">
  <soap:binding
    transport="http://schemas.xmlsoap.org/soap/http"
    style="rpc" />
  <operation name="Order">
    <soap:operation
      soapAction="urn:/BuySell/Community/BuySellProtocol/Order"
      style="rpc" />
    <input name="Order">
      <soap:body
        use="encoded"
        namespace="urn:SellerServer" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
```
Mapping of a protocol

```
- <portType name="BuySellProtocol">
  - <!-- original cx operation = /BuySell/Community/BuySellProtocol/Order -->
  - <operation name="Order">
    - <!-- original cx flow port = /BuySell/Community/BuySellProtocol/Order/Order -->
    <input name="Order" message="tns:Order" />
    <output name="OrderConfirmation" message="tns:OrderConfirmation" />
    <fault name="OrderDenied" message="tns:OrderDenied" />
  </operation>
</portType>
```
Mapping of message types

```
= <message name="Order">
  <part name="Order" type="Ordering:Order">
    <message name="OrderConfirmation">
      <part name="OrderConfirmation" type="Ordering:OrderConfirmation" />
    </message>
  </part>
</message>

= <message name="OrderDenied">
  <part name="OrderDenied" type="Ordering:OrderDenied" />
</message>
```
Mapping of data types

```xml
<xsd:simpleType name="CompanyID">
  <xsd:restriction base="xsd:nonNegativeInteger"/>
</xsd:simpleType>

<xsd:simpleType name="OrderID">
  <xsd:restriction base="xsd:nonNegativeInteger"/>
</xsd:simpleType>

<xsd:simpleType name="Item">
  <xsd:restriction base="xsd:nonNegativeInteger"/>
</xsd:simpleType>

<xsd:complexType name="Order">
  <xsd:sequence>
    <xsd:element minOccurs="1" maxOccurs="1" name="CompanyID"
       type="CoreTypes:CompanyID"/>
    <xsd:element minOccurs="1" maxOccurs="1" name="OrderID"
       type="Ordering:OrderID"/>
    <xsd:element minOccurs="0" maxOccurs="unbounded" name="Item"
       type="Ordering:Item"/>
  </xsd:sequence>
</xsd:complexType>
```
Enterprise Service Bus

Logical SOA Tiers and Components

<table>
<thead>
<tr>
<th>Client</th>
<th>Presentation</th>
<th>Business</th>
<th>Intermediary</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Apps</td>
<td>Portlet or Web Container</td>
<td>EJB, POJO, or Servlet Container</td>
<td>JBI Container</td>
<td>Heritage Systems</td>
</tr>
<tr>
<td>Java Apps</td>
<td></td>
<td></td>
<td></td>
<td>RDBMS</td>
</tr>
<tr>
<td>Mobile Device</td>
<td></td>
<td></td>
<td></td>
<td>Other Apps</td>
</tr>
<tr>
<td>B2B Gateway</td>
<td>Domain Objects</td>
<td>Domain Objects</td>
<td>Domain Objects</td>
<td>BPM Repository</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rules Repository</td>
</tr>
</tbody>
</table>

* Complements of jBoss
Many BPEL Processes support the CIM
Federal Enterprise Architecture

Support for the FEA as a view of the enterprise architecture
FEA (from reference)
Community Process

Roles, processes, activities

Reference model associations via aspect/properties
<table>
<thead>
<tr>
<th>name</th>
<th>baseline</th>
<th>planned</th>
<th>actual</th>
<th>achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Avoidance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity QuantityPerTime</td>
<td>0</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Productivity ProductsPerFTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity PercentResourcesUsed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity PercentImprovement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity PercentElectronic</td>
<td>0</td>
<td>100</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>CycleTime ProductionTime</td>
<td>5</td>
<td>0</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>CycleTime CycleToWaitTimeRatio</td>
<td>1</td>
<td>0</td>
<td>0.2799999999</td>
<td></td>
</tr>
<tr>
<td>CycleTime PlannedVersusActual</td>
<td>0.8</td>
<td>1</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>Quality EffectivityRate</td>
<td>0.8</td>
<td>1</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>Quality ComplaintsPerCustomer</td>
<td>0.1</td>
<td>0</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management ApplicationsRequired</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Management PolicyCoverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management PolicyComplianceExtent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management UnidentifiedRiskEvents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Net effect

• Using these open standards and automated techniques we can;
  – 80% Reduction in complexity (Conservative)
  – Achieve the strategic advantage of an open and flexible enterprise
  – Produce and/or integrate these systems FASTER and CHEAPER than could be done with legacy techniques
  – Provide a lasting software asset that will outlive the technology of the day
Sample Applications

- One-GSA Executable Enterprise Architecture for the General Services Administration
- Enterprise Component Architecture for U.S. Army PEO-STRI
- Intelligence application for Raytheon & DARPA
- Collaboration Architecture for Kaiser Permanente
Contact

Cory Casanave
Data Access Technologies
www.enterprisecomponent.com
cory-c@EnterpriseComponent.com
Military Example

Linking Tactical C2 systems with “Live”, “Virtual” and “Constructive” simulations (SIMCI)
Call for Fire - MLRS

Call for fire is a “protocol” of action and information exchange between parties.
Thread 2b - FS: CFF To GS MLRS (AFATDS)

Systems view of call for fire protocol

Alternate Fire Mission Sources
- Abrams
- Bradley
- Kiowa
- Longbow
- Fox
- Stryker
- Chinook
- Landwarrior

Original Action
Forward/Info Action
(Number denoted sequence)
<table>
<thead>
<tr>
<th>Step</th>
<th>Activity/Task</th>
<th>Originating OPFAC</th>
<th>System Information</th>
<th>Type</th>
<th>Destination OPFAC</th>
<th>System</th>
<th>Info Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System, Sensor, FO or FAC detects target; analyzes situation and makes call for fire request</td>
<td>OBS FBCB2</td>
<td>Call For Fire (CFF)</td>
<td>K02.4</td>
<td>Co FIST FOS</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Co FIST receives CFF request from platform, does a quick analysis (checking for dual targeting, etc.), and passes CFF to the Bn FSE</td>
<td>Co FIST FOS CFF Bn TOC AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Bn FSE receives CFF, makes determination that Bn assets cannot handle the mission, and passes the request to the Bde FSE</td>
<td>Bn TOC AFATDS CFF Pkg 10/11 Fmt Bde TOC AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Bde FSE receives CFF, makes determination that Bde assets cannot handle the mission, and passes the request to the FA Bde (Div Arty) FDC</td>
<td>Bde TOC AFATDS CFF Pkg 10/11 Fmt FA Bde TOC AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>FA Bde FDC receives CFF, makes determination that GS FA assets can handle the mission, and passes the request to the FA (GS) Bn FDC</td>
<td>FA Bde TOC AFATDS CFF FA Bn TOC AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FA Bn FDC receives Fire request/order to fire, assigns mission to MLRS battery</td>
<td>FA Bn TOC AFATDS CFF Pkg 10/11 Fmt MLRS Btry CP AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Btry FDS receives OTF and transmits this OTF to the MLRS platoon FDS</td>
<td>MLRS Btry CP ATADS OTF BCS Fmt MLRS Platoon FDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4 different messages transmitted back to the observer and as info to other echelons, depending upon the specific fire mission requested, some may not be necessary; &quot;Ready&quot; states that the battery is preparing to fire the mission, &quot;Shot&quot; stastes that the i</td>
<td>MLRS Btry CP AFATDS MTO, Shot, Spalsh, Rounds Complete Pkg 10/11 Fmt FA Bn TOC AFATDS A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thread 2b - FS: CFF to GS MLRS (FDS)

CFF mission (immediate suppression) initiated by System, Sensor, FO or FAC and fired by GS MLRS
Call for Fire - MLRS

Bde - FSE

Bde - Artillery

FA Bn - FSE

MLRS Battery

Co-FIST

Forward Observer

MLRS
Model Of CFF Thread
Model Information Flows

Call For Fire
Fire Ready
Rounds Complete
Fire Shot
Fire Splash
Mission Fire Rpt
End Of Mission

* Not technology details!
Choreography – Understanding When
Drill Down - Inside of a TOC

From “threads”
Inside of a TOC
Simulated or Real?

Operational

Simulated

Operational

Simulated

Operational

Simulated

Operational

Simulated
Simulating the Process

Tactical and Simulated components interacting
<CallForFire>
    <Content>
        <WeatherType/>
        <TerrainRelief/>
        <TargetFormation/>
        <FireMissionType/>
        <FireMethodOfControl/>
        <FireMethodOfEngage/>
        <FireRequestForFire/>
        <IsQuickFire/>
        <TargetRadius>0</TargetRadius>
        <TargetProtection>0</TargetProtection>
        <FalseTargetDensity/>
        <FireMission>
            <ProjectileType/>
            <FiringUnits/>
            <NumVolleys/>
            <NumRounds/>
        </FireMission>
        <LocationUpdateTime>
            <IsValid/>
            <Data/>
        </LocationUpdateTime>
    </Content>
</CallForFire>
Simulation Summary

• We can simulate a process
• Integrating real and simulated components
• Understanding the interactions
• At any level of detail
Trying Alternatives