FMEA: An executable services oriented enterprise architecture for financial management

OMG's Maximizing BPM Investments with SOA Workshop

Cory Casanave
cory-c (at) modeldriven.com
January 2008
FMEA in Context
* Based upon the One GSA Enterprise Architecture approach which provides a unified set of business services with supporting shared services. One GSA EA is intended to be a primary enabler of the Presidents Management Agenda (PMA), the Federal Transition Framework, and the ongoing demand from GSA customers to provide unified and customer focused services. The transition, as demanded by GSA's customers, is to present GSA services as a unified solutions framework supported by vertical horizontal lines of business.
Approach

• Business focus, facilitated with technology
• Services Oriented Architecture (SOA) at both the business and technical level
• Described with Collaborative Role Interactions, Processes and Information models based on OMG standards
• Model Driven Architecture (MDA) to connect the business and technical architectures
• JEE, JMS & Web services as the technical interface to the line of business
• Tools Used
  – Magic draw UML
  – ModelDriven.org (open source project) for MDA provisioning
  – Eclipse
  – jBoss suite
Business Focus Using Model Driven Architecture

MDA Terms

One GSA/FMEA Business Model
Enterprise Services (e-SOA)
Roles, Collaborations & Interactions
Process & Information

Logical System Model
Technology Services (t-SOA), Components
Interfaces, Messages & Data

Technology Specification
JMS, JEE, Web Services
WSDL, BPEL, XML Schema

Business Concerns

Refinement & Automation
Line-Of-Sight

Copyright © 2008 Model Driven Solutions
Incorporating Legacy Analysis

![Diagram showing the transition from As-Is to To-Be with categories: Computation Independent, Platform Independent, Platform Specific.]

- As-Is
  - FSS Business Model
  - OneGSA Business Model

- To-Be
  - System Specification
  - Technology Specification

- Knowledge Discovery Model
  - System Details
Value derived from the architecture

**Business Concerns**
- One GSA Business Model
- Business Services (b-SOA)
- Roles, Collaborations & Interactions
- Process & Information

**Logical System Model**
- Technology Services (t-SOA), Components
- Interfaces, Messages & Data

**Technology Specification**
- Web Services
- WSDL, BPEL, XML Schema

---

**Component Acquisition Specification**

**OMB 300**

**FEA/FTF**

**BRM**

**SRM**

**DRM***

**Technology Interfaces**

**Test & Simulation**

**Components**

**Deployment**

**Data**

**Adapters**

---

**Business Driven Technology**

**Facilitating Business Processes**

---

Copyright © 2008 Model Driven Solutions
Focus on the Business Model

- **Business Concerns**
  - One GSA/FMEA Business Model
  - Business Services (e-SOA)
  - Roles, Collaborations & Interactions
  - Process & Information

- **Logical System Model**
  - Technology Services (t-SOA), Components
  - Interfaces, Messages & Data

- **Technology Specification**
  - JEE, JMS, Web Services
  - WSDL, BPEL, XML Schema
The enterprise as services

- Think about the enterprise as a set of interacting roles providing and using services to enable agility, cost savings and an effective transition framework

- **Externally**
  - The enterprise is part of the global supply chain, providing services to customers and using the services of suppliers

- **Internally**
  - Consider parts of the enterprise as providing services to other parts of the enterprise, and in term using the service of others
  - Like everything was outsourced as a service, it just happens to be done inside the organization.

- Business is modeled in terms of interacting roles – providing and using services – the essential concepts of enterprise SOA
Financial Management Enterprise Context

- The service-oriented business architecture of an enterprise is modeled as a Collaboration of enterprise-level Participants.

This is the use of A service contract specification

Our Focus

External enterprise level participants

Copyright © 2008 Model Driven Solutions
A Composite Service Contract

Financial Management is responsible for providing a number of Acquisition Accounting services.
Inside Financial Management

Service representing delegated responsibility for interaction with an external participant.

Roles of participants inside of finance

Service representing interaction with another participant within Financial Management.
A service contract is modeled as a **UML Collaboration**.

The required conversation may be specified using an **Owned Behavior** (e.g., Interaction or Activity).

Note that, while one Participant requests the service and the other responds, information may flow both ways during the interaction.

First the submitter submits a bill to the receiver…

…then either the bill is successfully delivered or it is returned.
• Workflow is modeled using *UML Activities.*
Establish Unfilled Customer Order Subactivities

- Complicated activities may be decomposed into subactivities.

Diagram:

- Activity: Establish Unfilled Customer Order
  - Input parameters: customer order establishment, customer order established, customer order rejected
  - Output parameters: general ledger transaction

- Subactivity: Record Unfilled Customer Order
  - Internal information flow

- Subactivity: Establish Recurring Receivable
  - Recurrent customer order
Record Unfilled Customer Order Behavior

- Ultimately, behavior can be specified using basic UML Activity Diagrams.

Control flow

- Create unfilled customer order
- Send rejection
  - [no]
  - Successful?
    - [yes]
    - Post general ledger transaction
      - [no]
      - Should recurring receivable be established?
        - [yes]
        - Send recurrent customer order
      - [no]
  - Send acceptance
- Customer order established
- Customer order rejected
- General ledger transaction
- Recurrent customer order
Record Unfilled Customer Order Requirements

- Detailed requirements and business rules can be documented for activities separately from the process flow.

**Record Unfilled Customer Order**

**Description**

- Record a new unfilled customer order, as established via a specific sales agreement.

**Requirements**

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

2. If the Customer Order is against a Sales Agreement that requires recurring payments, establish a recurring receivable.

3. …
A term in the vocabulary represents a class of things to be described.

Attributes specify descriptive information having simple types.

Entities may be described as having a unique identity.

This indicates a compositional (as opposed to referential) association.

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

This means “zero or more”

This means “one or more”

A class may be specialized into sub-classifications.

An un-shaded class is not detailed on this diagram.

Bill

+ bills sent
  + payee
  + bills received
  + payer

Received Bill (payer is self)

Sent Bill (payee is self)

Bill Line Item

+ containing bill
  + line items

+ sent bills
  + billed receivables

Billed Receivable

+detail type: Code
+ unit price: Amount
+ quantity: Amount
+ extended price: Amount

Attributes: bill ID: Identifier
principal amount: Amount
interest amount: Amount
shipping charge: Amount
administrative costs amount: Amount
penalties amount: Amount
total amount: Amount
issue date: Date Time
due date: Date Time
bill number: Numeric
bill type: Code
terms: Code
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.
The CIM is a model of the business, not the information system.
Producing the logical model

**Business Concerns**

- One GSA/FMEA Business Model
  - Business Services (e-SOA)
  - Roles, Collaborations & Interactions
  - Process & Information

**Logical System Model**

- Technology Services (t-SOA), Components
- Interfaces, Messages & Data

**Technology Specification**

- JEE, JMS, Web Services
- WSDL, BPEL, XML Schema
FMEA Systems Architecture (High Level)

* Any server may be clustered or combined as required

GSA SecureNet/MultiNet Financial Network

FMEA Services Server

FMEA User Interface

Session Management

FMEA Presentation Server

FMEA Integration Server

SOA JMS Broker

SOA Web Service Adapters

SAML

Identity Management

Load Balancer

Finance Users

Http:

FMS

FAS

TIRES

NEAR AR

NEAR

IRIS

STAR

PBS

Pegasys/Momentum

FMEA DBMS Server

FMEA Data

VPN

https Web Service

SAML

JMS

SAML

SAML

Session Management

FMEA Services

Work Unit

Data Manager

Logs
Receivables Accounting Business Architecture (from Business Model)
The Participant Types act as the Service Interfaces.
Receivables Accounting Component Architecture

User of a consumed service by multiple internal parts.
Receivables Management Activities
(from Business Model)

Customer Order Establishment

: Establish Unfulfilled Customer Order

- General Ledger Transaction
- Customer Order Established
- Customer Order Rejected

: Generate Recurring Receivable

: Establish Receivable and Accrue Revenue

- Charge Establishment
- Receivable Accepted
- Receivable Rejected

Related to Customer Orders

Related to Receivables
Receivables Management Component Architecture

- Explicit component for scheduling triggers
- Explicit cross-transactional coupling via the data tier

- Implements the Establish Customer Order activity.

- Implements the Generate Recurring Receivable and Establish and Accrue Revenue activities.
Messages – Transfer Information in Services

- **Asset Accounting Message**
  - **Asset Disposal Notification**
    - **Property Disposal**
      - **Property**
      - (May be a transfer or a write off)

- **Success message**: Asset Disposal Recorded
  - +success message 0..1
  - +referenced message 1
  - +referenced message

- **Failure message**: Asset Disposal Rejected
  - +failure message 0..1
  - +referenced message

(usage = modify)
+disposed property

(usage = create)
+disposal
Messages can be modeled by "marking up" the information model. Realization is used to "mark up" information model elements with message model elements. Restricted realization allows for explicit inclusion and exclusion of attributes and associations.
Summary

The PIM is a model of the system and how it realizes the business requirements, but the technology decisions and details are abstracted out.
Technology Architecture

**Business Concerns**

One GSA/FMEA Business Model
Business Services (b-SOA)
Roles, Collaborations & Interactions
Process & Information

Logical System Model
Technology Services (t-SOA), Components
Interfaces, Messages & Data

Technology Specification
JEE, JMS, Web Services
WSDL, BPEL, XML Schema
Platform Specific Model

<table>
<thead>
<tr>
<th>Platform Independent Model (PIM)</th>
<th>Platform Specific Model (PSM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Financial System Specification</td>
<td>Core Financial System Implementation</td>
</tr>
<tr>
<td>Service Interfaces</td>
<td>Web Services</td>
</tr>
<tr>
<td>Enterprise Components</td>
<td>Enterprise Information Systems</td>
</tr>
<tr>
<td>Work Components</td>
<td>System Components</td>
</tr>
<tr>
<td>Service Manager Components</td>
<td>System Functions</td>
</tr>
<tr>
<td>Behavioral Specifications</td>
<td></td>
</tr>
<tr>
<td>Data Model</td>
<td>Data Definition</td>
</tr>
<tr>
<td>Message Specifications</td>
<td>XML Schemas</td>
</tr>
<tr>
<td>Data Manager Components</td>
<td>Data Bases</td>
</tr>
<tr>
<td>Persistent Data Specifications</td>
<td>Data Base Schemas</td>
</tr>
</tbody>
</table>
Example Web Services Generation

```xml
<wsdl:portType name="BillSubmission.BillSubmissionReceiverInterface">
  <wsdl:operation name="submitBill">
    <wsdl:input message="tns:BillSubmissionCluster"
               name="billSubmission">
    </wsdl:input>
  </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="BillSubmission.BillSubmissionSubmitterInterface">
  <wsdl:operation name="notifyBillDelivered">
    <wsdl:input message="tns:BillDeliveredCluster"
               name="billDelivered">
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="notifyBillReturned">
    <wsdl:input message="tns:BillReturnedCluster"
               name="billReturned">
    </wsdl:input>
  </wsdl:operation>
</wsdl:portType>
```
Example Transaction Message XML Document

```xml
<BillSubmissionCluster>
    <BusinessTransaction>
        <transactionID> ... </transactionID>
    </BusinessTransaction>
    <BillSubmission>
        <bill>
            <Bill>
                <billID> ... </billID>
                <principleAmount> ... </principleAmount>
                ... <payer>
                    <Party>
                        <partyID> ... </customerID>
                    </Party>
                </payer>
                ... <lineItems>
                    ... 
                </lineItems>
            </Bill>
            <billingAddress>
                <BillingAddressCluster>
                    <Address> ... </Address>
                    <BillingAddress> ... </BillingAddress>
                </BillingAddressCluster>
                <billingAddress>
            </BillingAddress>
        </bill>
    </BillSubmission>
</BillSubmissionCluster>
```
Putting it all together

<<Provision>>
Asset Accounting Services Tier
(Language = "Java+XML,
Technical Architecture = "JEE-Messaging")

<<MessageDrivenBean>>
<<StatelessSessionBean>>
: Asset Lifecycle Accounting Services

<<MessageDrivenBean>>
<<StatelessSessionBean>>
: Receivables Accounting

<<XSLT Implementation>>
: Property Data Manager

<<XSLT Implementation>>
: Project Data Manager

<<XSLT Implementation>>
: Party Data Manager


<provisioningContext name="service"...>
  <projectRef folder="EjbClient"/>
  <projectRef folder="AppClient"/>
  <projectRef folder="Ejb"/>
  <projectRef folder="ear"/>
  <projectRef folder="JbossConfig"/>
</provisioningContext>
Putting it all together
```java
public class Asset_Record_Establishment_ProviderInternal {
    ......
    static public Document establish_asset_record(Document request) throws CheckedException {
        // for an inbound operation, determine if we delegate or execute
        return ServiceFactory.getPipeline("Asset_Record_Establishment_Transaction_Manager.
            Asset_Record_Establishment_Provider.establish_asset_record.Pipeline").execute(request);
    }
}
```
public class Asset_Completion_Establishment_ConsumerAsset_Completion_Establishment_Provider_InterfaceInternal {
  ...
  static public Document establish_asset_completion(Document request) throws CheckedException {
    ...
    return gov.gsa.fmea.Asset_Record_Manager.Asset_Completion_Establishment_ProviderAsset_Completion_Establishment_Provider_InterfaceInternal.
    establish_asset_completion(request);
  }
}
Putting it all together

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes"/>

  <xsl:template match="/">
    <Asset_Project_Status_Notification_Transaction_Manager.Asset_Completion_Establishment_Consumer.establish_asset_completion..xslt

    <xsl:function name="mdf:Asset_Project_Status_Notification_Transaction_Manager.Asset_Completion_Establishment_Consumer.establish_asset_completion..xslt

    <xsl:apply-templates select="$documentFragment" mode="mdf.schema.copy">
      <xsl:with-param name="namespace" select="'http://www.modeldriven.org/xsd/FMEA_Asset_Accounting_Implementation_Model.uml/Asset_Completion_Establishment'"/>
    </xsl:apply-templates>

    <xsl:sequence select="Asset_Completion_Establishment_Consumer.establish_asset.completion"/>
  </xsl:function>

  </xsl:template>
</xsl:stylesheet>
```
End result – this executes

Service representing delegated responsibility for interaction with an external participant.

Service representing interaction with another participant within Financial Management.

Roles of participants inside of finance
On this infrastructure

GSA SecureNet/MultiNet Financial Network

* Any server may be clustered or combined as required
Late breaking news

• 2/3 of the way through this process, JSA decides to make a commitment to JBI on Sun Glassfish
• For 10% for funds, the same application is provisioned to a new technical architecture
• Change takes about 6 weeks, now components can be deployed to either infrastructure with no change.
• Most of the time is spent just debugging glassfish
• How would this change have been possible without MDA?
Conclusion

• FMEA is a general architecture of the federal financial services domain, done for GSA by Model Driven Solutions.
• It supports both internal GSA needs as well as the “line of business”.
• It uses MDA, SOA and BPM to provide a business centric architecture, drilling down to technology models.
• Artifacts can be generated for model based acquisition, the FEA, testing, service interfaces, data management, workflow and components.
• FMEA is entering the next phase of acquisition and implementation.
• All assets to produce this application are being donated to ModelDriven.org as open source.