Modeling Enterprise Collaboration

A tutorial on applying Model Driven Architecture using the OMG Enterprise Collaboration Architecture with UML

Introductions

Cory Casanave
cory-c@enterprise-component.com

Primary author of “CCA” in EDOC
What is the Enterprise Collaboration 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 RFP.

The Agile Enterprise

*Has a competitive advantage in its capability to embrace collaboration and change*
Integrating Enterprises, People & Systems - Worldwide

Business Requirements
- Virtual Enterprises
- Enterprise Integration (EAI)
- Supply-chain automation (B2B)
- Customer Integration (B2B)
- Web deployment (B2C)
- Internet Marketplace (B2C)

Collaboration and Integration
The dynamic reality

- The information system must facilitate;
  - Rapid realization of business goals
  - Integration of independent processes and systems
  - Multiple and Changing
    - business requirements
    - business processes
    - technologies
    - standards
    - enterprise boundaries
    - partners

Technology Stew

- Web services
- .NET
- C'
- XML
- EAI
- Active Web pages
- EJB
- Java Beans
- Java
- Corba
- MQ-Series
- C++
- SQL
- Cobol
- IMS
- CICS
- ...

Technology is transient, but we must embrace and adapt to it to provide meet current requirements
**Problems to be solved**

- Integration Nightmare
- Infrastructure, Version & Vendor lock-in
- Complex, divergent and manual development and deployment processes

Typical solutions require buy-in (Lock-in) to expensive, pervasive and proprietary infrastructure

---

**Goals**

- A scalable and robust enterprise architecture
- Loosely coupled enterprise components
- Enable rapid provisioning of business solutions
  - Simple, reproducible processes supporting reuse
- Technology & vendor independence
- Enable the integration and collaboration of multiple;
  - Business units (internal and external)
  - Customers
  - Suppliers
  - Systems
  - Technologies
The new center

- The strategic core of your systems must be the business itself
- Only technology independent business focused models will survive the transience of technology and lock-in
- These models can become part of your source code, driving enterprise applications
- Enabler: Model Driven Architecture (MDA) with EDOC-ECA

**Extreme Modeling**
**Collaborative Vs. Managed business processes**

- The enterprise collaboration architecture is intended for collaborative business processes
  - People, departments, systems and enterprises working together within a defined process
- Contrast with the managed business process
  - A central authority directs the activities of others
- Both have their place
  - Collaborative - B2B, EAI
  - Managed - Fine-grain business processes - factories
Collaborative Business Processes

Independent actors working within a defined process

No central authority

Managed Business Processes

Someone or something keeps track of everything
Collaboration is Key

Collaboration is a key differentiation and key cost center (Healthcare Example)

- Customer Collaboration
- Claim processing
- Disputes
- Physician Collaboration
- Payer Collaboration
- Hospital Collaboration
- Broker Collaboration
- Government Collaboration
- Employee Collaboration
- Others...

The system integrates multiple collaborations

Automated Model Driven Architecture

Profile (E.G. EDOC) → Business Focused Model (UML) → Tools Produce & Integrate

Enterprise Components

Mapping is tuned to the infrastructure

Mapping is tuned to the infrastructure

Minimize and structure manual implementation
EDOC Structure

Enterprise Architecture

How to slice and dice
Loose Coupling

- Loose coupling is the ability for independent parts of systems to be built and evolve independently
- Tightly coupled systems
  - Prevent change (the next legacy system)
  - Cause lock-in
  - Become unmanageable
  - Prevent reuse
- Quality architecture is essential for loose coupling

Architecture Goals

- Create a system from loosely coupled “enterprise components” that can evolve independently
- Provide well defined interfaces and interaction points between these enterprise components
- Make each enterprise component a reusable asset that can serve many business processes
- Build the information system as a community of interacting enterprise components
- Utilize open standards such as XML, EJB and Corba to integrate the enterprise components
Enterprise Components

- Enterprise Components must be independent
- While being able to interoperate with each other
- Making the information system a lattice of cooperating components

Poor Monolithic Architecture

Making a monolithic web application doesn’t help!

All business rules, data rules, application logic, technology and user interface code are contained here
“Wrapping” Legacy Applications and Data

- Enterprise Components are defined in terms of their external contract; implementation may use existing applications
  - Can “call” existing application
  - Can read and write legacy DBMS
  - Can use “screen scraper” (Last resort)

- Legacy applications can appear as enterprise components but may not be implemented as components
Legacy “Wrapping”

Wrapping allows existing programs and data to work with and work as enterprise components

Technology Independence

Copyright © 2001-2002, Data Access Technologies, Inc.
Typical Requirement

Multi-tier implementation

Multi-tier implementation

Model Driven Architectures

Integrating and Automating Design To execution
**MDA Overview**

- Modeling has evolved
  - It's not just pictures anymore
- Use high level UML models made precise with profiles
- With technology specific mappings
- Define and produce substantial parts of the executable system

**Models and mapping**

- Platform-independent model
- Business Model
- Map to platform-specific artifacts (IDL/DTD)
- Versioned repository
- Management over time

Platform-specific artifacts:
- UML/CORBA
- UML/EJB
- XML
  - OAG
  - SOAP
  - ebXML
  - RosettaNet
- Legacy
Ways to map

- Generate code and other artifacts
  - Usually required to adapt to legacy
  - Emulates the manual process
- Assemble and configure existing generic and specific components
  - Both efficient and dynamic
- Interpret or otherwise “animate” the high-level model
  - High level & dynamic but may have performance issues

Example Generated Artifacts

Artifact generation involves multiple tools
- EJB Container provider; Deployment tools; Packagers;
- Java development tools (IDE); persistence provider; …
- Typical 10-20 per PIM Classifier
- 0-20% manual override
Reverse Engineering

Native artifacts

- Native meta-model is platform-specific
  - XML DTD/Schema; java introspection; SQL tables; legacy model; etc.
- Map navigates the native meta-model, populates PIM
- Limited semantic recovery
- Information and middleware models work best

MDA and Components

Platform independent

Business Model

Assemble & Execute

Components

Enabled but not required
What can we automate?

Summary of MDA benefits

- Isolates domain specifications from platform details
  - Reduces complexity
  - Preserves domain model semantics
  - Increases stability and lifetime
  - Generates to platform/legacy of choice
- Decreased development time
  - fast iterative development
  - separation between the engineering and business requirements
- Increased quality.
- Builds on industry directions
Collaborations and Roles

Conceptual Foundation

Portions copied with permission of Trygve Reenskaug - Taskon

OORAM

(http://www.ifi.uio.no/~trygver)

History

OORAM Object Oriented Role Analysis
UML Collaborations
Enterprise Collaboration Architecture

Influence
The Connected Enterprise
Content and Communication

Digital Map
Census Data
Police Records
House Drawings
Aerial Photos

Multiple roles in a collaboration

Copyright © 2001-2002, Data Access Technologies, Inc.
Roles to Systems

Component in Role

Interaction Path

Interaction (With Information)

Implementation

Framework, Middleware & Container

Operating System

Hardware

Roles to Systems

Collaboration

Role

Travel Expense Example

1: travelPermissionRequest

2: travelPermission

3: expenseReport

4: authorizedExpenseReport

5: paymentRequest

Copyright © 2001-2002, Data Access Technologies, Inc.
## Travel Permission Decision
### Travel Expense System

<table>
<thead>
<tr>
<th>Traveler</th>
<th>Peter</th>
<th>Period</th>
<th>week 11</th>
<th>Planned cost</th>
<th>USD 2000.-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Attend TOOLS Europe 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Current plan for Peter**

<table>
<thead>
<tr>
<th>Activity</th>
<th>10</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>week 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>week 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>week 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Budget + commitments**

<table>
<thead>
<tr>
<th>Item</th>
<th>Budget</th>
<th>Committed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel</td>
<td>10,000</td>
<td>4,000</td>
</tr>
</tbody>
</table>

- **Permit**
- **Reject**

---

## Synthesis - Components

Components play several roles:

- **ProjectManager**
- **ProjectParticipant**
- **Jane**

---

[Images and graphics related to the text are present, showing a flowchart and text information related to travel permissions and roles in synthesis.]
Discussion Point

- What are other ways of representing business semantics?
- What role do these play?
  - Business Processes
  - Information Models (DBMS)
  - Events
  - Workflow
  - Internet
- What is used internally?

Standards for Global Internet Computing

XML
EDOC
WSDL
.NET
BPML
SOAP
XLANG
XML-Schema

**XML Standards**

- **XML Schema & DTD**
  - Description and packaging of data
- **WSDL**
  - Specification a services, operations and flows available via that service
- **Soap**
  - Basic messaging and packaging
  - Extensions for Soap-RPC with WSDL
  - May be extended to support collaborative messaging

**Vision**

- **EDOC (a UML Profile)**
  - Provide an architecture for open collaborative computing
  - Simplify the development of component based distributed systems by means of a modeling framework, based on UML 1.4
  - Provide a platform independent, recursive collaboration based modeling approach supporting multiple technologies.
  - Embrace Model Driven Architectures (MDA) – Provide design and infrastructure models and mapping
- **ebXML**
  - Creating a single global electronic market
    - Includes process specification, transport and repositories
    - Collaboration Oriented
JSR 159

- Java Process Components
- Uses ECA (CCA) as the baseline component model
- Asynchronous document exchange
- Component composition
- A J2EE technology mapping for ECA
- Will make J2EE an excellent integration and web services platform

---

ebXML & EDOC

ECA as the normal form

The standard way to model and tool for multiple technologies

Summary of points thus far

- We must enable the emerging Internet Computing Model
  - Loosely coupled roles exchanging documents based on a contract of collaboration
- Web need interoperability at two levels
  - Messaging for the data
  - Metadata for the contract of collaboration, stored in repositories
- This model of collaborating roles is recursive, extending into the enterprise, into managed domains and into applications
  - Inside the enterprise we want to include resources entities, business events and business processes
- Supporting the open domain has some required parts and can be augmented with a “treasure chest” of tools and infrastructure
- Between EDOC & ebXML we are covering B2B and intra enterprise
How to realize global collaborations

EDOC Component Collaboration Architecture

The model of collaborative work
The Marketplace Example

Mechanics Are Us
Buyer

Process Complete

Acme Industries
Seller

Physical Delivery

GetItThere Freight
Shipper

Status

Order

Conformation

Ship Req

Shipped

Delivered

The Seller’s Detail

Order

Conformation

Shipped

Ship Req

Shipped

Delivered

Order Processing

Shipping

Event

Receivables

**Parts of a CCA Specification**

- Structure of process components and protocols
  - Process components, ports, protocols and documents
    - Class Diagram or CCA Notation
- Composition of process components
  - How components are used to specify components
    - Collaboration diagram or CCA Notation
- Choreography
  - Ordering of flows and protocols in and between process components
    - Activity Diagram

---

**The Community Process**

- Identify a “community process”, the roles and interactions in a collaboration

---

ECA Entity Profile

The model of things
Data Inside a “shared domain”

Adding Entities

- Entities are added to manage entity data
- Entity Roles are managers that provides a view of the same identity in another context
- The Entities have ports for managing and accessing the entities
- Non-entities which are owned by (aggregate into) an entity are managed by the entity
ECA Business Events

The model of when...
Loosely coupled integration within the enterprise and with “aligned” business partners

Event Based Business Processes

**Event Example**

![Event Diagram]

**Events or Services**

- **Services** - conversation between known parties
  - Good supply chain model
- **Events** - Actions based on notification of events
  - Parties do not know about each other!
  - More loosely coupled
  - Works best inside a managed domain (enterprise)
**EDOC CCA Example**

Buy-Sell Process

**Community Process**

UML Collaboration Diagram
Community Process (CCA)

Protocol

Protocol (CCA)

Define Documents
Protocol Choreography

EDOC Exercise - Community

- Part one
  - Pick a collaboration from the first exercise
  - Model it with EDOC notation as a community process
- Identify an interaction
  - Specify the documents of the interaction
    - What are some of the major elements?
  - Specify any choreography
Validation Component

UML Class Diagram

Validation Component (CCA)
Choreography

When

Order Validation Choreography

CheckCustomer

acceptOrder

reject

success

failure

Internal Logic

Composition (UML Collaboration)
Composition (CCA)

Sample Information Model
Adding Entities

- Entities are added to manage entity data
- Entity Roles are managers that provide a view of the same identity in another context
- The Entities have ports for managing and accessing the entities

Modeling Events

What to do when something happens
Event Example

Does ECA map well to WSDL?

An approach to the mapping
ECA/WSDL mapping

- ECA works well as a modeling framework for WSDL
- How major concepts could map
  - WSDL Port <-> ECA Port
  - WSDL Operation <-> ECA Flow port (one way) or Operation (Two way)
  - WSDL Service <-> ECA Component
  - WSDL Port type <-> ECA Protocol
  - WSDL Message <-> ECA Document type

WSDL/ECA Differences

- WSDL Adds
  - Technology binding and endpoints
- ECA adds
  - Choreography, nested conversations, two-way protocols, nested components.
The WSEC RFP

Web Services for Enterprise Collaboration

This RFP has not yet been issued

Solicits proposals for

- A Mapping from EDOC-Enterprise Collaboration Architecture to WSDL 1.1 with attachments and a SOAP binding.
- A Mapping from WSDL 1.1 with a Soap binding to the EDOC-Enterprise Collaboration Architecture.
- Any required extensions to the EDOC-Enterprise Collaboration Architecture to represent WSDL semantics.
ECA Methodology

A simple methodology for creating collaborative business processes

Basic Steps

- Identify roles and organize roles into collaborations
- Define collaboration documents
- Create basic business transactions
- Organize into protocols and events
- Use protocols to define ports on roles
- Drill-down into role detail
- Implement roles
- Configure implementations for deployment with technology specifics
- Deploy
Identifying roles and collaborations

Identify Documents
Distinguish protocols and events

Create Business Transactions
Organize into protocols

Add ports to complete community process
Drill-down

Add implementation

- As component compositions
- In a programming language
- By using an external service
Add technology specifics for deployment

Complete process

- Implemented roles are deployed
- Completing the process
Vision

Building and adapting systems for collaboration, reuse and change

Loosely coupled enterprise architecture

- Independent enterprise components
- Representing business concepts
- Link via open standards

Copyright © 2001-2002, Data Access Technologies, Inc.
Business Component Marketplace

- The business component marketplace is projected to be a 10b market in 5 years
- Consider the value of XML components that wrap popular legacy
- New application functionality built from components
- Components for integration and transformation
- XML and web services makes an excellent basis for such components
- Technology components, such as for repositories and DBMS
- Marketplace may be inside the enterprise or commercial

OMG Model Driven Architecture (MDA)

- High level - platform independent models
- Technology Models
- Mapping
  - Custom
  - Standard
- Standard Models produce technology specific standards artifacts
Automated MDA

Profile (E.G. EDOC)

Infrastructure Mapping (E.G. XML)

UML Design

Tools Produce & Integrate

Enterprise Components

Framework & Infrastructure (E.G. XML)

Mapping is tuned to the infrastructure

Technology Independence

Adapters
- ebXml
- BizTalk
- Rosetanet
- EJB

Business Logic Component

Adapters
- EJB
- MQ
- Corba
- CICS

High level tooling & infrastructure

MUST BE SIMPLE!
- We must be able to create better applications faster
- We must separate the technology and business concerns, enable the user

Tooling + Infrastructure
- Executable models are source code
- Tooling must be technology aware
- Infrastructure must support tooling, not manual techniques

Model based component architectures
Iterative Development

Net effect

- Using these open standards and automated techniques we can:
  - 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 asset that will outlive the technology of the day
Advanced EDOC

Platform Independent Models
Technology mappings

Platform Specific Models

Integration - viewpoints

CCA

Entities

Events

Process

Relationships

Patterns

Business Process Profile

how things are coordinated
**Business Processes**

- Specialize CCA
- Activity-centric view of a Process
- Express
  - Complex temporal and data dependencies between business activities
  - Iteration of activities
  - Alternative required Inputs and Outputs of activities
  - Roles related to performers, artifacts and responsible parties for activities

---

**Relationships Profile**

**Useful associations and dependencies**
**Relationships Profile**

- Enables non-binary aggregations
- Defines useful association and dependency stereotypes

```text
<<Aggregation>>
<<Packet>>
<<Assembly>>
<<Subordination>>
<<List>>
```

**Patterns Profile**

- reusing parameterised designs
Patterns Profile

- Profiles UML Parameterized Collaborations
- Based on Business Function Object Patterns (BFOP)
  - Multi-Layer
  - Based on Catalysis Approach
- Adds stereotypes for
  - Named Patterns
  - Inheritance
  - Composition
  - Pattern Binding with renaming

Platform Specific Modelling

- EJB, FCM, MOF
- Technology mappings from EDOC to Distributed Component and Message Flow Platform Specific Models
  - EDOC to J2EE/EJB mapping
  - EDOC to CORBA/CCM mapping
  - EDOC Business Process to FCM mapping
  - EDOC Business Process to CORBA mapping
ECA as the normal form

The standard way to model and tool for multiple technologies

Web Services (WSDL)
edXML (BPSS)
J2EE (Java RMI)
.NET
MOM (MQ-Series)

MDA Mappings

Discussion

- How much are we using these kinds of concepts today?
- What would be the advantages to going in this direction?
- What are the barriers;
  - Technical?
  - Political?
- What would be a good first step?
Contact

Cory Casanave
Data Access Technologies
www.enterprise-component.com
cory-c@enterprise-component.com
(305) 234-7077