Can MDA help Defense and Government Achieve Interoperable Components?
Introductions

Cory Casanave

cory-c@EnterpriseComponent.com

Primary author of “CCA” in OMG-EDOC
What we must achieve

- Network Centric Warfare
- The Objective Force
- Battlefield Awareness
- Adaptive Responsiveness
- Economy
- Unified Enterprise Architecture
- Agility & Adaptability
  - Design for change
  - Build for adaptability
- Collaboration
  - Horizontal
  - Vertical
  - Coalitions
- Joint Interoperability
- System of Systems
The Sad Reality

- Most Information Systems
  - Do not adapt quickly
  - Do not intemperate
  - Take years to produce
  - Fail or fail to meet needs
  - Are difficult to evolve
  - Lock the user into specific technologies
  - Are not economical

- These are monolithic dinosaurs
The Problem Space

- Systems of Systems – Wide scale interoperability, reuse and integration
- Massively redundant legacy
- Acquisition methods encourage closed systems
- Solutions are bound to the problem and technology of the day
- Massive information loss – trapped in paper documents and proprietary technologies
Strategic Migration to Reusable Components

Customer Focused Domain Models Drive Agile Systems of Interoperable Components

Systems Composed of Interoperable Components

Standards based integration of Monolithic Systems, eg. HLA

Ad Hoc Point to Point Integration of Monolithic Systems

Separate and Non-Interoperable Applications
Component Layers

Component Support Layer

Abstract Component Collaboration Layer
- Roles
- Interactions
- Information

Business Logic Component Layer
- Compositions
- Legacy Wrappers
- Code
- Models

Technology Component Layer
- Containers
- Languages
- Middleware
- Adapters

Application Components

Defines How
Implements

Standards
Meta Models
Infrastructure
Repository
Tools
Component Based Solutions
Fast Diagram

How

Iterate

Deliver Products

Validate System

Compose Solution

Test Framework

Runtime Framework

Automated Provisioning

Reuse Existing Components

Build & Adapt Components

Add to Library

Provision Tests

Component Library

Identify Existing Components

Design/Adopt Generic Component Architecture

Design Application Component Architecture

Fund & Contract

Add to Library

Provision Tests

Iterate

Why

New Requirements

Integrate Requirements

Requirement Repository

Design/Adopt Generic Component Architecture

Component Support

Abstract Layer

Business Logic

Implementation

Infrastructure
Simulated Model Driven Architecture

Enterprise Architecture Model (PIM)

Domain Architecture

Simulator

ECA Standard “Meta-Model” & UML Profile

Live Process Simulation

Refine/Iterate
Automated Model Driven Architecture

- Meta-Model UML Profile (E.G. ECA)
- Enterprise Architecture Model (CIM)
- Infrastructure Mapping (E.G. J2EE-WS)
- Tools Produce & Integrate
- Enterprise Components
- Framework & Infrastructure (E.G. -J2EE-WS)

Mapping is tuned to the infrastructure

Minimize and structure manual implementation
Automated Model Driven Architecture

- Meta-Model: UML Profile (E.G. ECA)
- Domain Architecture
- Enterprise Architecture Model (CIM)
- Infrastructure Mapping (E.G. J2EE-WS)
- Tools: Produce & Integrate
- Mapping is tuned to the infrastructure
- Multiple and Changing Technology Support

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

Simulated Enterprise Components
Simulation Infrastructure
The new center

- The strategic core of your systems must be the enterprise itself.
- Only technology-independent enterprise-focused models will survive the transience of technology and lock-in.
- These models can become part of your solution, driving enterprise applications and simulations.
Executable Enterprise Architecture

- MDA at the enterprise level – how systems of systems work together
- Focus on components and processes – components work together for goals
- MDA models work at the system composition level – “code” is useless at this level
- Executable architecture can be simulated and then be refined with “real” components to evolve to the real solution
Example - GSA

• General Services Administration
  – Achieve a “one GSA” enterprise architecture
  – Understand how processes can be facilitated with solutions
  – Uncover redundancy
  – Integrate to-be process with legacy systems
Specifying Interactions
Expanded Protocol Detail
Choreography of Process
Simulation
Drilling Down
Generated Web Service
Example – PEO-STRI

- Achieve common components across product lines
- Eliminate artificial boundaries between simulation domains
- Support joint interoperability
- Model how simulations and C2 systems work together
- Create common adapters between simulations and operational systems
Call for Fire - MLRS
Choreography – Understanding

When
Model Information Flows

* Not technology details!
Example - Intelligence

- Integrate diverse information from multiple sources
- Allow any “processor” to utilize any source
- Automate logical and technical transformations
- Highly distributed
- Create common component architecture
Common Themes

- Component Reuse
- Wide-scale interoperability
- Systems of systems
- Elimination of redundancy
- Technology Independence
- All fit well into the Enterprise Collaboration Architecture Approach
- Political and Acquisition Problems
Common Environment for Intellectual Capital

Integration of infrastructure

Value Chain Modeling
UML Modeling
Workflow Tools
Business Modeling
Collaboration Modeling

Meta Object Facility “Meta Models”

MDA Environment
Models define the system

MOF

Intellectual Capital

Apl 1
Cl CS
EJ B
.NET
Cust Sys
The Object Management Group (OMG)
- Major industry consortia for open interoperable systems
- Hundreds of members - every major vendor
- Successes include Corba, UML, MOF, CWM, XMI

The OMG is defining MDA as the way systems will be built and conceived
What MDA Achieves

- Rapid and reliable system specification and development
- Early visibility of results
- Adaptability to changing requirements and technologies
- Technology independence
- Automation reduces time, cost and errors
- Longer system life
- Interoperability and componentization
- Collaboration and integration
- System components become assets, not expenses
Iterative MDA Development

- Business Model Design
- Automation
- Infrastructure Development

Build > Build > Build > Build > Build

Release Build

Deploy

Repeat for future systems & evolution
Directly addressing our requirements

- Network centric warfare
- The Objective Force
- Battlefield Awareness
- Information centric
- Adaptable and responsive
- Economy

- Agility & Adaptability
  - Design for change
  - Build for adaptability

- Collaboration
  - Horizontal
  - Vertical
  - Coalitions

- Interoperability
- System of Systems
Important MDA Standards

• **Unified Modeling Language (UML)**
  – Providing the basic modeling framework

• **Meta Object Facility (MOF) and XMI**
  – Providing a shared and interoperable repository for models at many levels
  – XMI provides for interchange of models in XML

• **Enterprise Collaboration Architecture (ECA)**
  – Providing the way to model collaboration and integration using MDA components
  – Provisioning in multiple technologies
Conclusion

- There is a drastic need to solve a related set of problems
  - Component Reuse
  - Interoperability
  - Collaborative Processes
- Enterprise MDA is an essential enabler
- Political problem are the most difficult