Use of MDA in the SIAP Program

Presented at:
MDA Implementers’ Workshop:
Succeeding with Model Driven Systems
May 17-20, 2004 – Orlando, FL

By:
Basil Krikeles and Robert Merenyi, ALPHATECH, Inc.
John Brtis, The MITRE Corporation

©2004 – The MITRE Corporation
Agenda

- Single Integrated Air Picture (SIAP)
- The Integrated Architecture Behavioral Model (IABM) PIM
- From PIM to PSM – Procedure, Verification, Validation
- Issues
- Summary
Problem Definition

Operational Problem:
- Engagements constrained by
  - Procedural controls
  - Target ID
  - Sensor limitations
- Lack of interoperability among
  - Weapon Systems
  - Sensors
  - C4I

Operational Elements:
- Single Integrated Air Picture
- Combat Identification
- Integrated Fire Control
- Automated Battle Management Aids
- Attack Operations
- Passive Defense / Early Warning
Process Overview

- Integrated Architecture Behavior Model (IABM) requirements sourced from all Services
- JSSEO outputs test requirements for Joint Interoperability Test Command (JITC)
- JSSEO produces IABM for use by Services
  - PIM developed by industry, university, FFRDC and government team
  - Distributed through Program Offices to primes
The SIAP Goal

- A federation of peers over the network implementing the same processing logic
  - Each peer is a Platform-Specific Implementation (PSI) of the same IABM PIM
  - Peers are deployed on heterogeneous platforms
- Peers can share consistent information such as Combat ID
  - Common functionality implemented and maintained commonly
- Each peer can generate an improved, more complete, picture of the battlespace by utilizing information received from other peers
- Specifications for the shared behavior are expressed in a form that is not susceptible to human error or interpretation by the contractors
- Verification and validation of correct performance can be accomplished uniformly across all weapons systems
Using MDA to Implement SIAP

- Critical processing logic (business rules) captured in the IABM (PIM), guaranteed to be applied uniformly across all IABM Peers
- Requirements and specifications encapsulated in the IABM are precise and not subject to interpretation by prime contractors
- Potential n to 1 reduction in the maintenance cost of the core processing logic (n Weapon Systems, 1 IABM)
- Separates functionality and implementation
  - Provides design stability as implementation technologies evolve and improve
Cost Implications of MDA

OMG Presentation

Conventional Development Total Cost = Cost of System A + Cost of System B

- Analysis/Design
- Coding
- Integration
- Testing
- Program Mgt

System A
- a
- b
- c
- d
- e
- f

System B
- a
- b
- c
- d
- e
- f

Duplicate Effort

IABM Development (PIM)

- Analysis/Design
- Coding
- Integration
- Testing
- Program Mgt

System A (PSM)
- a
- b
- c
- d
- e
- f

System B (PSM)
- a
- b
- c
- d
- e
- f

MDA Cost Savings over n Weapon Systems = n * (a + b + c)

The cost of both PIM development and maintenance is amortized across multiple systems.
Overview of IABM Approach

**OMG Presentation**

Core domains capture common behaviour to ensure uniformly applied processing rules in a net-centric environment.

Isolates application from hardware devices and data buses.

Isolates application from underlying execution technologies.

Run-Time Interface Domain

Isolates application from user interface formats and devices.

Target Execution Environment

(Processors / Operating System / Language)

Target Hardware

- Weapons
- Sensors
- Comms

- Adaptation Layer
  - Sensor Interface Comms Domain

- User Interface Domain
  - HUD/HDDs
  - Displays
  - Other Tools

Isolates application from user interface formats and devices.
Notional PIM-to-PSI Transformation

Requirements Development

IABM in iUML

Platform libraries

TA-x (tool)

Xform rules in iUML

Compiler (tool)

Platform-Specific Model Compiler

Transformation Development

iCCCG (tool)
PSM Generation

**Creation of Customized Code Generator**
- Platform Translation Rules (Transformation model)
- Compiler Meta Model

**Creation of Target Executable**
- Platform-Specific Build Set (PSBS)
- Legacy headers & libraries
- Generated C++ (PSM)

**Execution Flow**
- PSMC (exe)
- g++
- Executable binary

**Tools and Components**
- Compiler Generator
- Platform-Specific Model Compiler (PSMC)
- Platform-Specific Implementation (PSI)
- xUML
- g++
PSI Verification and Validation

**Simulation Test Objectives**

- Does PSI behave as an expected instance of the IABM?
  - Can be verified using stub interface code
- Does PSI behave as expected by weapon system platform?
  - Initial stub testing can reveal early performance numbers
  - Final testing requires fully populated platform library code
SIAP System Development Process

OMG Presentation

[Diagram showing the SIAP System Development Process with various stages and steps, including development, testing, and integration phases.]
Challenges

• Tools supporting MDA are still evolving
  – UML standards for interchange and transformations being defined

• Configuration Management unique to the MDA process
  – See Wednesday’s “Practical MDA” sessions

• Horizontal acceptance
  – New government procurement paradigm
  – Affects prime’s business model
Summary

• SIAP is an important goal of the DoD that is now being addressed using a novel approach both in terms of development as well as government procurement

• The Joint SIAP System Engineering Organization has adopted Model Driven Architecture with xUML as their development and deployment methodology

• Shared processing logic is captured in JSSEO’s Integrated Architecture Behavior Model (IABM) – an xUML model can be deployed in multiple weapons systems