Welcome

16 July 2008

Chris Raistrick, Kennedy Carter

Experiences in the Use of MDA and UML in Developing NATO Standards

chris.raistrick@kc.com
Use of MDA and UML in Developing NATO Standards

DODAF OV-1

- NATO Aircraft (Manned or Unmanned) plug-and-play equipped
- NATO Weapon (and Launcher) plug-and-play ready
- Integrated Air Picture
- Time Sensitive Mission Data
- Weapon release programme
- NATO Rapid Reaction Force
- Lethality Service User

Broker
- Publish
- Subscribe

- Store Description Data
- Store Load-out Configuration Data

Rapid Integration and Test

Discovered Lethality Service with NATO Network Enabled Capability

Lethality Service User

Mission Plan
NATO Aircraft, Launcher & Weapon Interoperability – Teams

Team 1: ALWI Technical Architecture
Consensus document for standards adoption

Team 2: Plug and Play Methodology and Architecture
Development of plug-and-play approach using MDA

Team 3: Universal Aircraft-Store Interface
Assessment of USAF Universal Armament Interface (UAI) approach

Team 4: Plug and Play Implementation
Realizing MDA based specifications in aircraft hardware and software

Use of MDA and UML in Developing NATO Standards
Plug and Play Weapons

In a perfect world...
It should be possible to load any of these weapons...

...onto any of these airframes...
...and make available a set of common core capabilities...
...even if some weapon-specific capabilities are not available
Plug and Play Weapons

In the real world...
Weapons have differing capabilities and different comms interactions...

...and each aircraft has a different execution environment...
Plug and Play Domain Architecture

- Achieves weapon type independence
- Achieves comms platform independence
- Achieves execution platform independence

Software System

Weapon Control

Existing Weapon | Future Weapon

Communications

Existing Comms Technology | Future Comms Technology

xUML Execution Platform

Any Operating System | Any Language

Target Hardware

Weapon specific plug-ins

Comms specific plug-ins

Language specific plug-ins
The ALWI Domain Architecture

Network Level Services

Mission Level Services

Weapon Control Services

Plug-in Weapon Control

Comms Services

Plug-in Comms Services

UML Execution Services

Plug-in Execution Technologies

Use of MDA and UML in Developing NATO Standards
The Data-Code Spectrum

A primary goal was to *avoid changing existing certified modules at all costs*

When adding a new weapon to an existing aircraft
1. Capture weapon-specific behaviour *in data* if possible, but if not possible...
2. Capture weapon-specific behaviour *in separate, small modules* (domains)

**Fully Data Driven,**
for weapons that have capabilities that can be fully described in data

**Partially Data Driven,**
for weapons that have capabilities that can be partially described in data, but require some additional code

**Fully Concrete,**
for weapons whose behaviour is so special they require dedicated code modules

Data Tables (Objects)

Plug-in domains

Use of MDA and UML in Developing NATO Standards
Weapon Control Domain Classes – Overview

The Weapon Control Domain is a metamodel…

…that captures in data the behaviour of each weapon.
Use of MDA and UML in Developing NATO Standards

Weapon Control Domain Regions

Receive Input Messages | Determine New Weapon State | Send Output Messages and Execute Weapon Actions

“Common Services” API

Weapon Plug-in
Generic State Machines and Messages

Use of MDA and UML in Developing NATO Standards
Generic State Machines and Messages – Adding A New Effector

Use of MDA and UML in Developing NATO Standards

(Part of) State Machine for Photon Torpedo

(Non) State Machine for Photon Torpedo

[Flux Generator Failure] / DeactivateFluxG

PhononGeneration

entry / RunGenerator

[Flux Generator fully loaded]

(Data Set Table for Input Message Causing Transition::PHOTON TRANSITIONS)

<table>
<thead>
<tr>
<th>weaponTypeName</th>
<th>inputMessageName</th>
<th>oldStateName</th>
<th>newStateName</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHOTON TORPEDO</td>
<td>INITIALIZATION_COMPLETE</td>
<td>INITIALIZATION</td>
<td>STANDBY</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>XRAAM_COMP_MODE</td>
<td>STANDBY</td>
<td>XRAAM_COMPATIBILITY_MODE</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>ACTIVATE_FLUX_GENERATOR</td>
<td>STANDBY</td>
<td>PHOTON_GENERATION</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>FLUX_GENERATOR_FULLY_LOADED</td>
<td>PHOTON_GENERATION</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>ASSIGN_TARGET</td>
<td>ACTIVE</td>
<td>TARGET_ASSIGNED</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>DEACTIVATE</td>
<td>ACTIVE</td>
<td>STANDBY</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>TARGET_REJECT</td>
<td>TARGET_ASSIGNED</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>NO_TARGET</td>
<td>TARGET_ASSIGNED</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>LAUNCH_TORPEDO</td>
<td>TARGET_ASSIGNED</td>
<td>FREE_FLIGHT</td>
</tr>
<tr>
<td>PHOTON TORPEDO</td>
<td>FLUX_GENERATOR_FAILURE</td>
<td>PHOTON_GENERATION</td>
<td>STANDBY</td>
</tr>
</tbody>
</table>
Plug and Play Implementation
Mapping PIM (xUML) to PSI (AADL)

<table>
<thead>
<tr>
<th>xUML</th>
<th>(Software) AADL (Hardware)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>System</td>
</tr>
<tr>
<td>State Machine</td>
<td>Process</td>
</tr>
<tr>
<td>Signal Event</td>
<td>Thread (Group)</td>
</tr>
<tr>
<td>Operation</td>
<td>Subprogram</td>
</tr>
<tr>
<td>Attribute</td>
<td>Data</td>
</tr>
<tr>
<td>Service</td>
<td>Port</td>
</tr>
<tr>
<td>Interaction</td>
<td>Connection</td>
</tr>
<tr>
<td>Actor</td>
<td>Device</td>
</tr>
</tbody>
</table>

Use of MDA and UML in Developing NATO Standards
The ALWI Process

The ALWI process can be summarised as...

- Build and test a system model, using a precise modelling formalism
- Populate the code generator with the system model and platform-specific configuration, and generate the target system

System Model

- Certified ALWI PIMs
- Plug-in Weapon Data
- Plug-in Weapon Code

System Generator

- Platform-Specific Implementation Generator
- Plug-in PIM-PSI Mappings

Generated System

- Generated Platform-Specific Implementation
- Runtime Layer
Summary

- From the ALWI Final Report:
  “The proposed methodology for standardizing platform independent services is based on the Object Management Group (OMG) Model Driven Architecture™ (MDA®) initiative. The combination of MDA common services and common ICDs promises to offer ‘plug and play’ in the long term.”

- The ALWI xUML architecture and MDA process are to be used as the starting point for a new NATO Study (SG125) for Unmanned Airborne Vehicles...

---

JCGUAV Unmanned Air Systems (UAS)
MALE/HALE/Weaponization ST

**Proposed NIAG Study on Weaponizing**
Unmanned Air Systems (UAS)

1 April 2008
Experiences in the Use of MDA and UML in Developing NATO Standards

16 July 2008

Chris Raistrick, Kennedy Carter

chris.raistrick@kc.com