OBJECT MANAGEMENT GROUP

Workshop on Distributed Object Computing for Real-time and Embedded Systems

July 14 – 16, 2008, Washington, DC, USA

Welcome

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

16 July 2008

Chris Raistrick, Kennedy Carter chris.raistrick@kc.com

KENNEDY CARTER

COMPLEXITY SIMPLIFIED

KC.COM

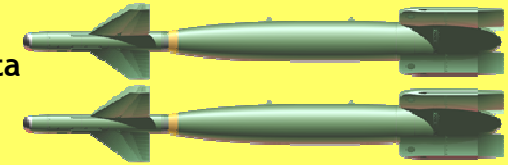
DODAF OV-1

Discovered Lethality Service with
NATO Network Enabled Capability

NATO Aircraft
(Manned or Unmanned)
plug-and-play equipped

Rapid Integration
and Test

Store Description Data
Store Load-out Configuration Data



NATO Weapon (and Launcher)
plug-and-play ready

Broker

Publish

Subscribe

Weapon
release
programme

Mission Plan

Integrated Air Picture
Time Sensitive Mission Data

NATO Rapid
Reaction Force

Lethality Service User

NATO Aircraft, Launcher & Weapon Interoperability – Teams

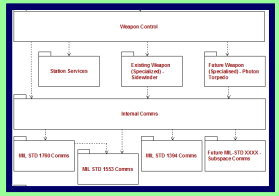
Team 1: ALWI Technical Architecture

Consensus document for standards adoption

Team 2: Plug and Play Methodology and Architecture

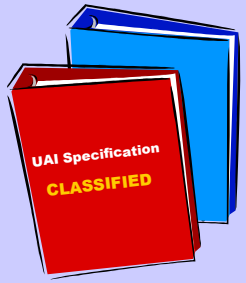
Development of plug-and-play approach using MDA

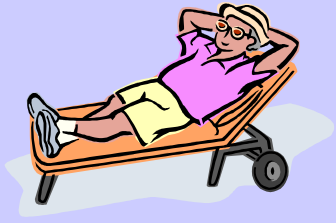
PnP PIM

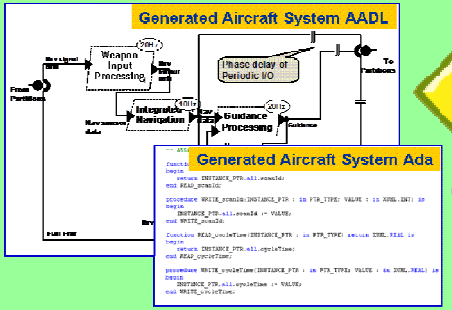


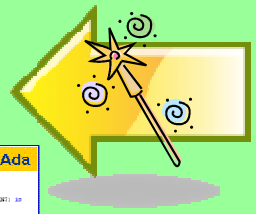
Team 3: Universal Aircraft-Store Interface

Assessment of USAF Universal Armament Interface (UAI) approach









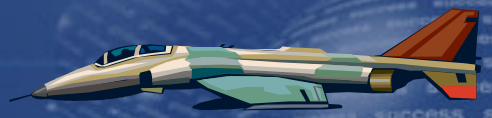
PnP PIM

Team 4: Plug and Play Implementation

Realizing MDA based specifications in aircraft hardware and software

Workshop on Distributed Object Computing for Real-time and Embedded Systems

July 14 – 16, 2008, Washington, DC, USA



Plug and Play Methodology and Architecture

KENNEDY CARTER

COMPLEXITY SIMPLIFIED

KC.COM

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

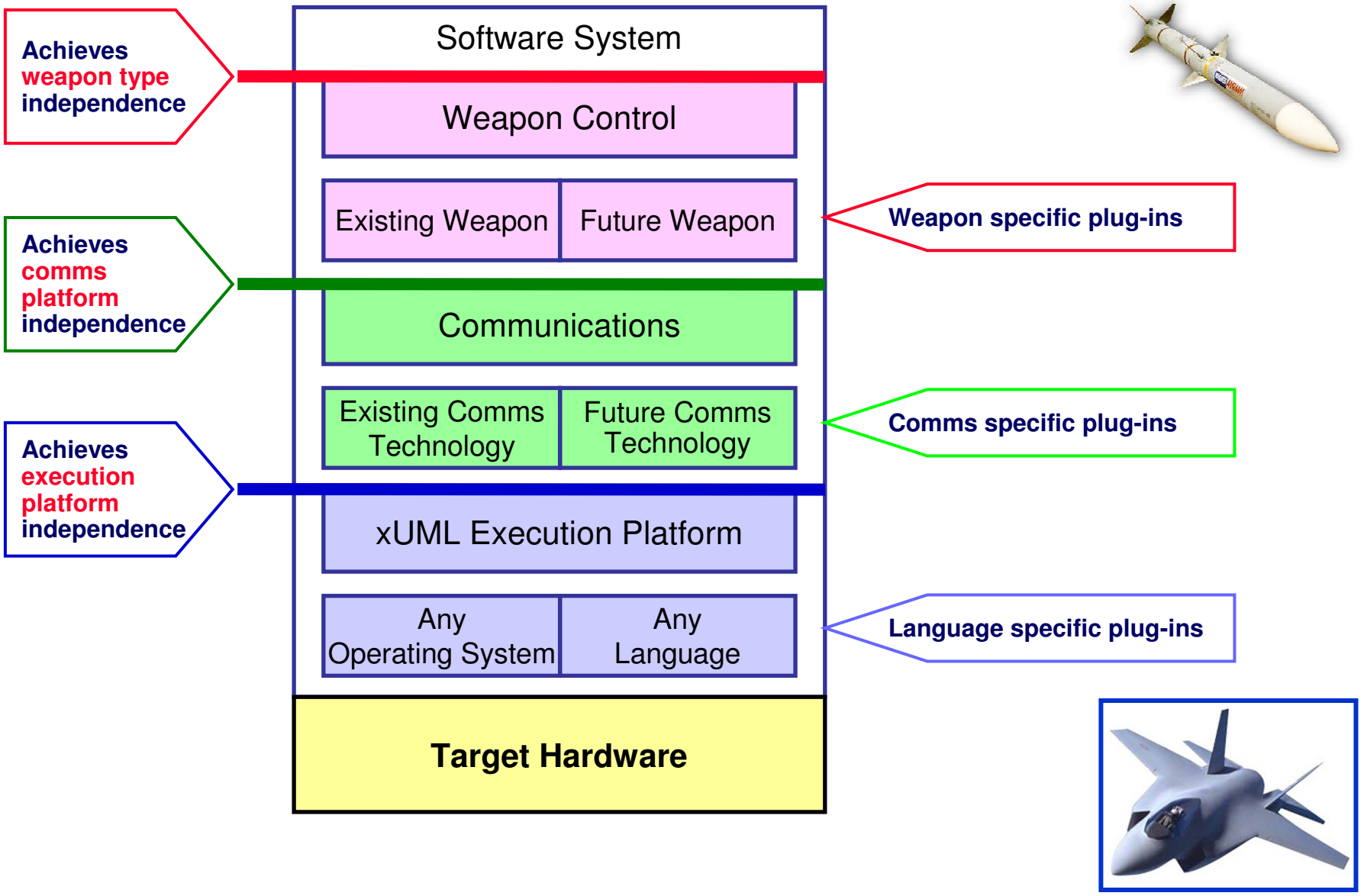
Plug and Play Weapons

In the real world…

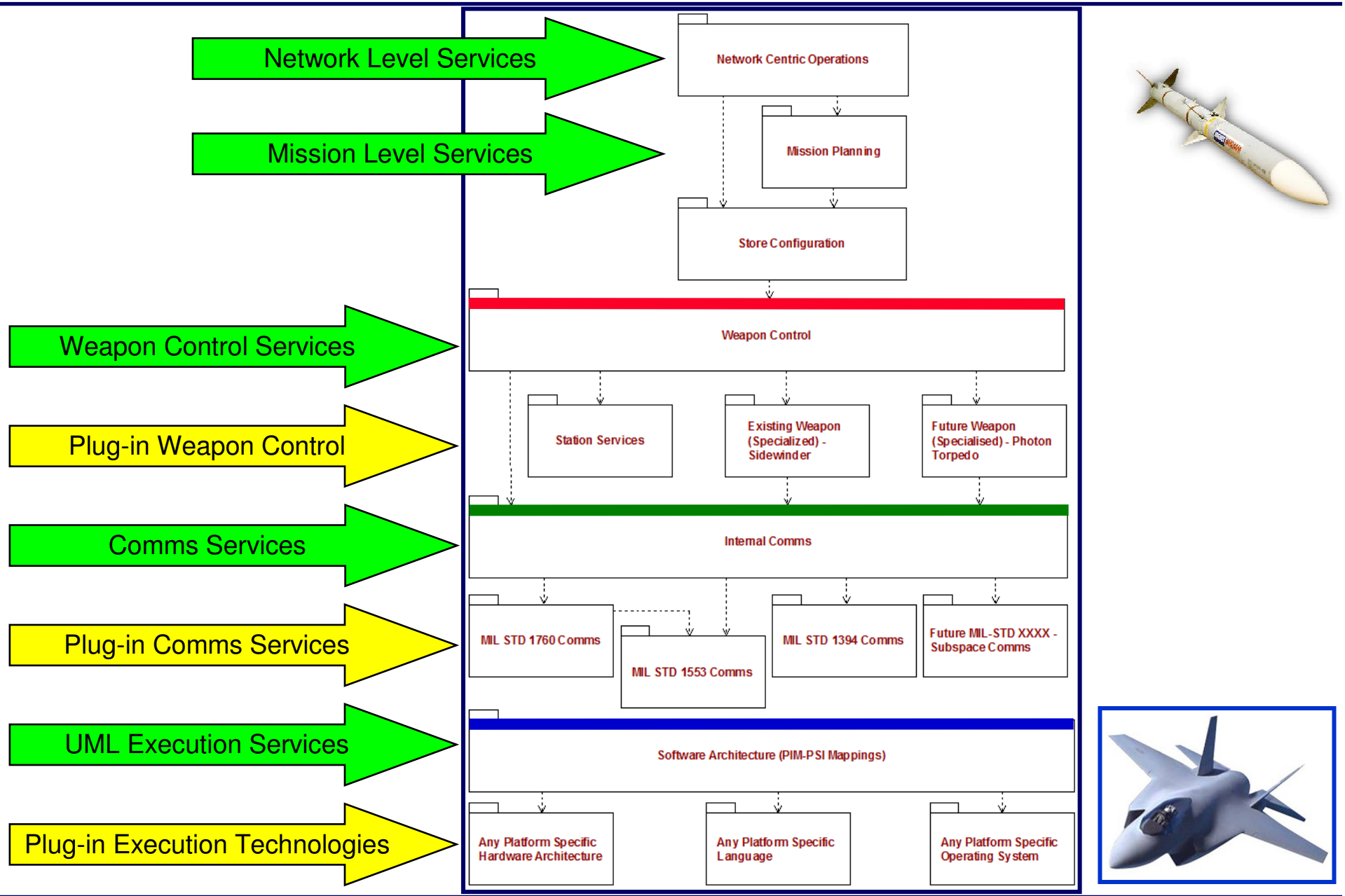
Weapons have differing capabilities and different comms interactions…

…and each aircraft has a different execution environment…

Plug and Play Domain Architecture



The ALWI Domain Architecture



The Data-Code Spectrum

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

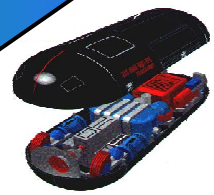
When adding a new weapon to an existing aircraft

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

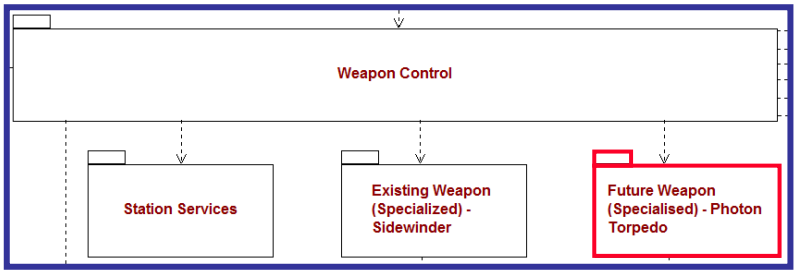


Data Tables (Objects)

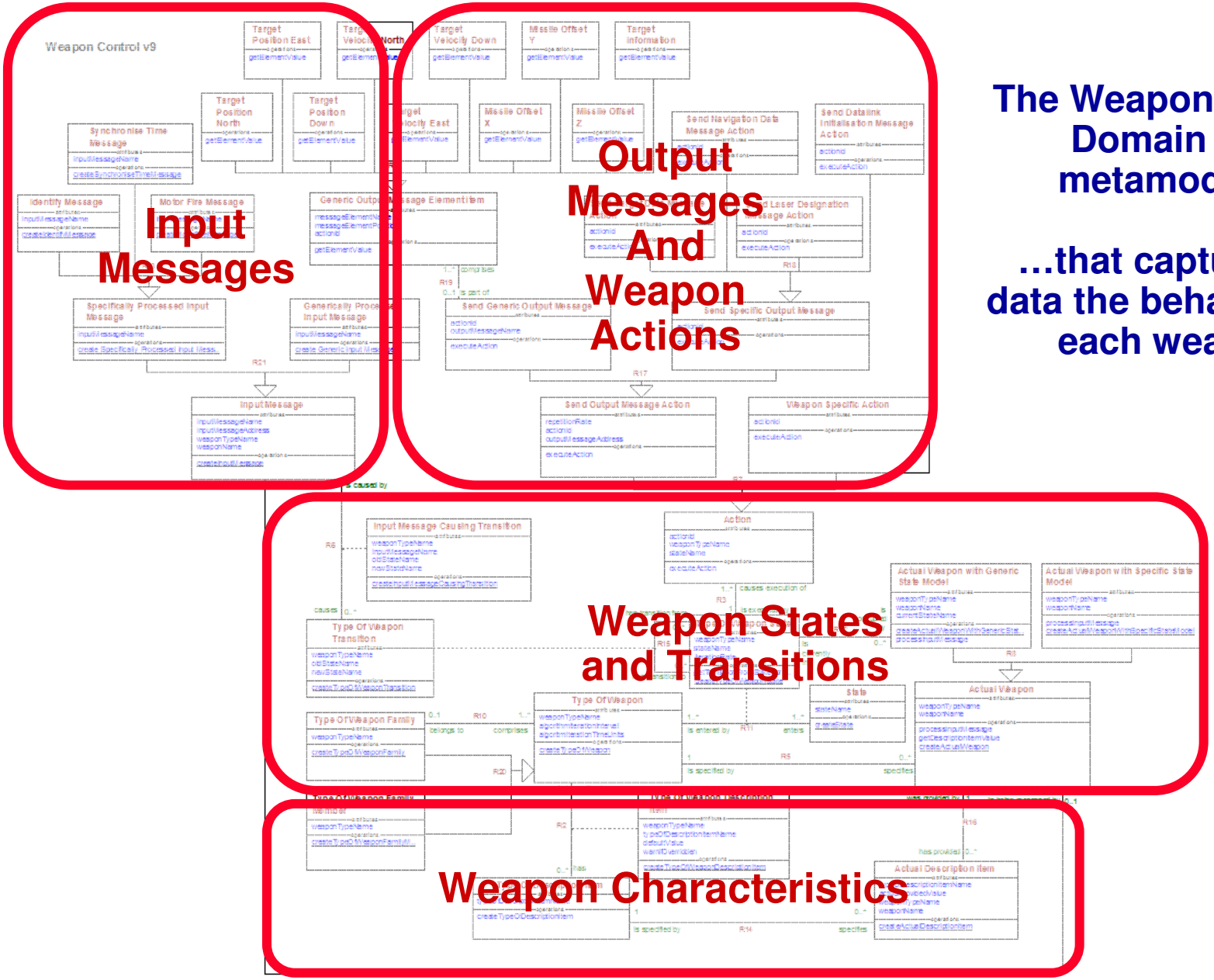
4: DataSet Table for Type Of Weapon State::AMRAAM Stat...

	weaponTypeName	stateName	iterationRate
2	AMRAAM	POWER_ON_BIT	0
3	AMRAAM	DETERMINING_BIT_RESULT	0
4	AMRAAM	OFF	0
5	AMRAAM	INITIALISING_EFFECTOR	0
6	AMRAAM	PREPARING_FOR_LAUNCH	10
7	AMRAAM	LAUNCHING	0
8	AMRAAM	POST_LAUNCH_GUIDANCE	50
9	AMRAAM	ENGAGEMENT_COMPLETE	0

Plug-in domains



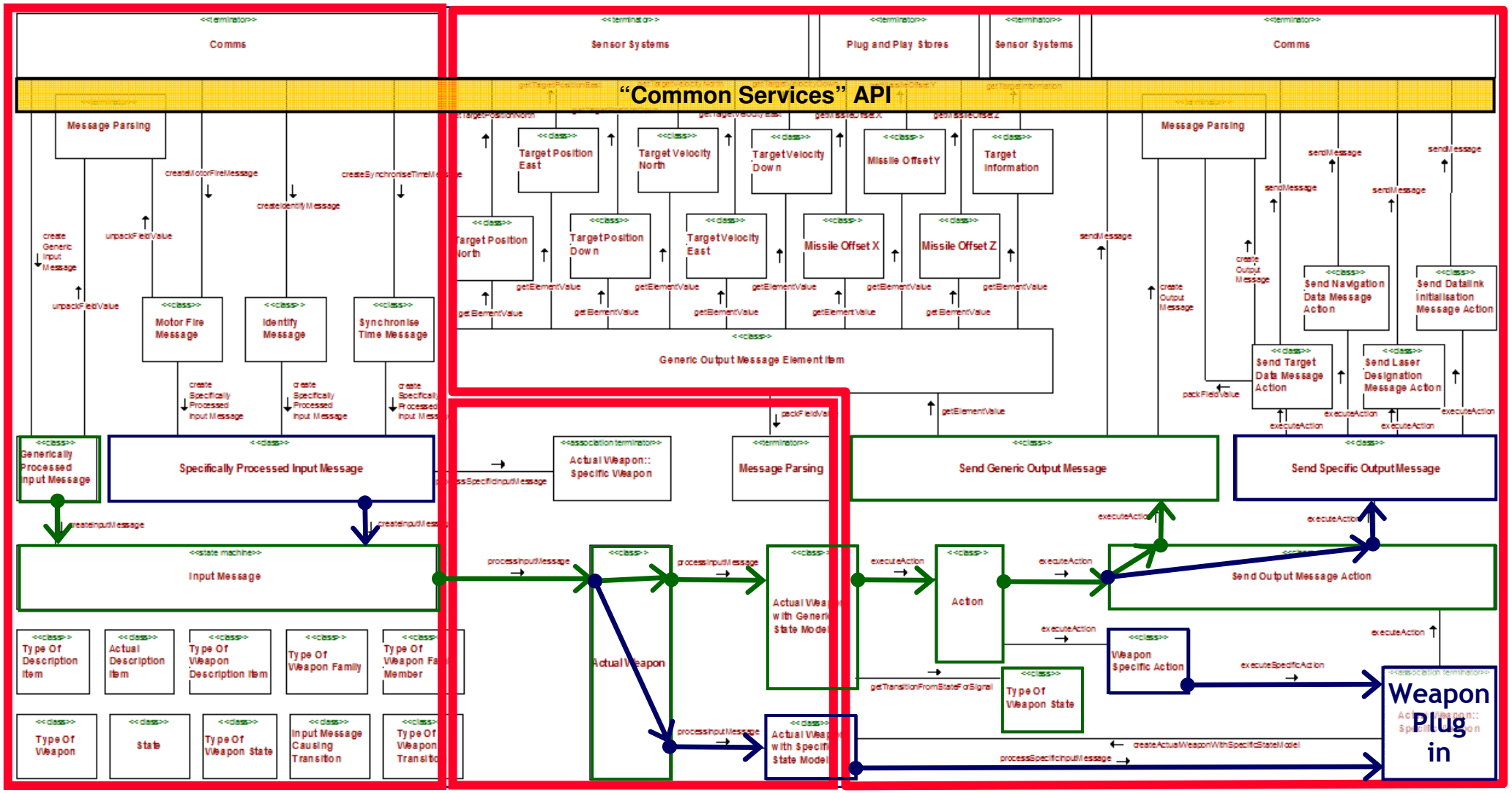
Weapon Control Domain Classes – Overview



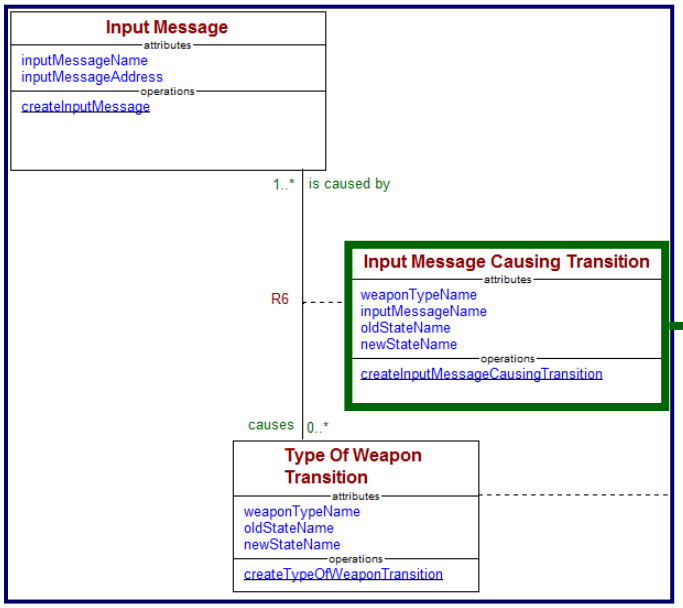
The Weapon Control Domain is a metamodel…

…that captures in data the behaviour of each weapon

Weapon Control Domain Regions

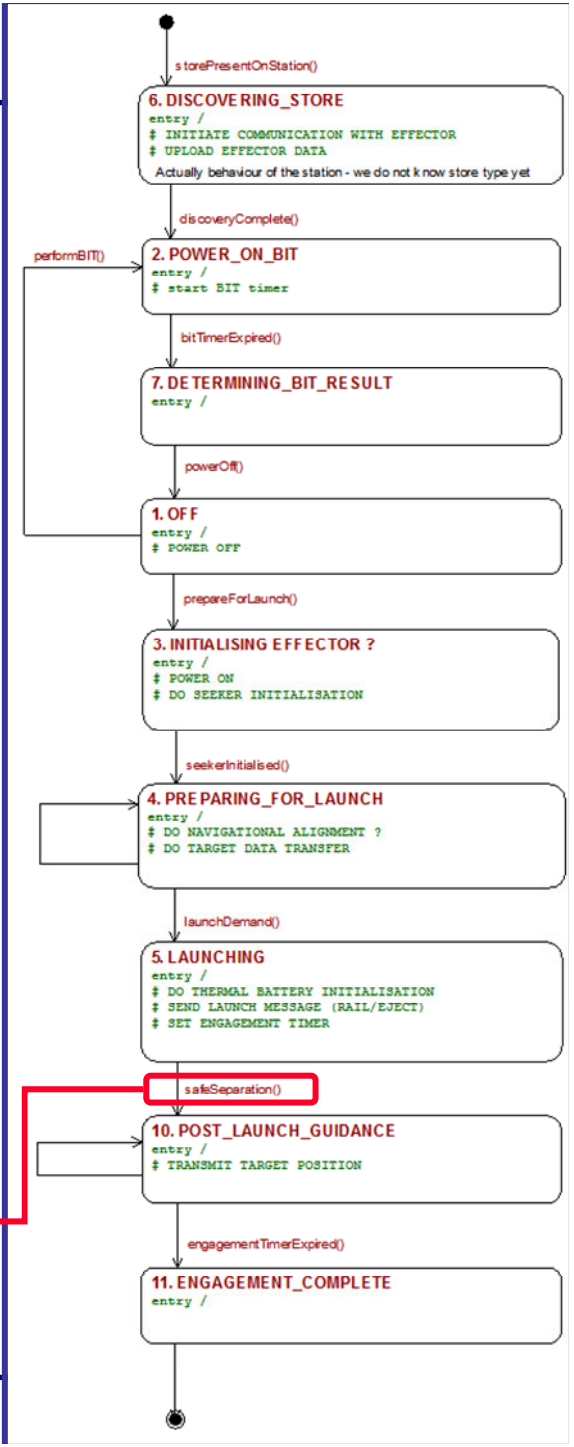


Generic State Machines and Messages

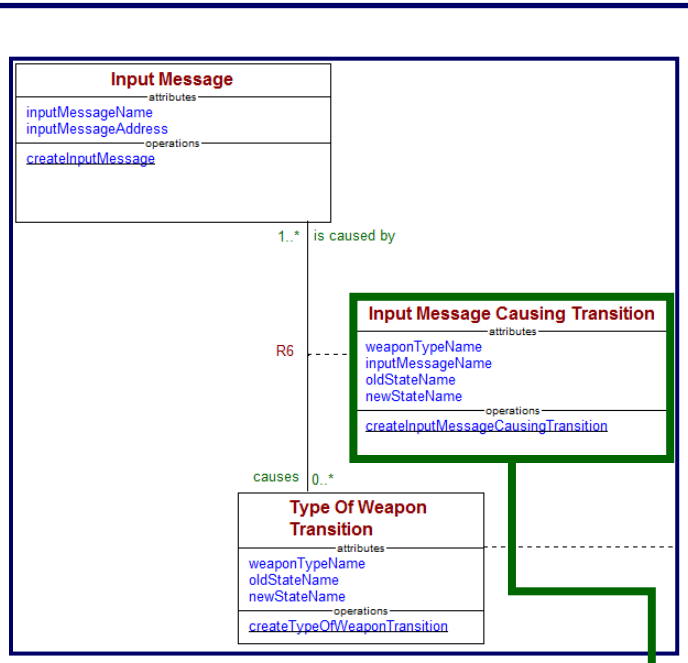


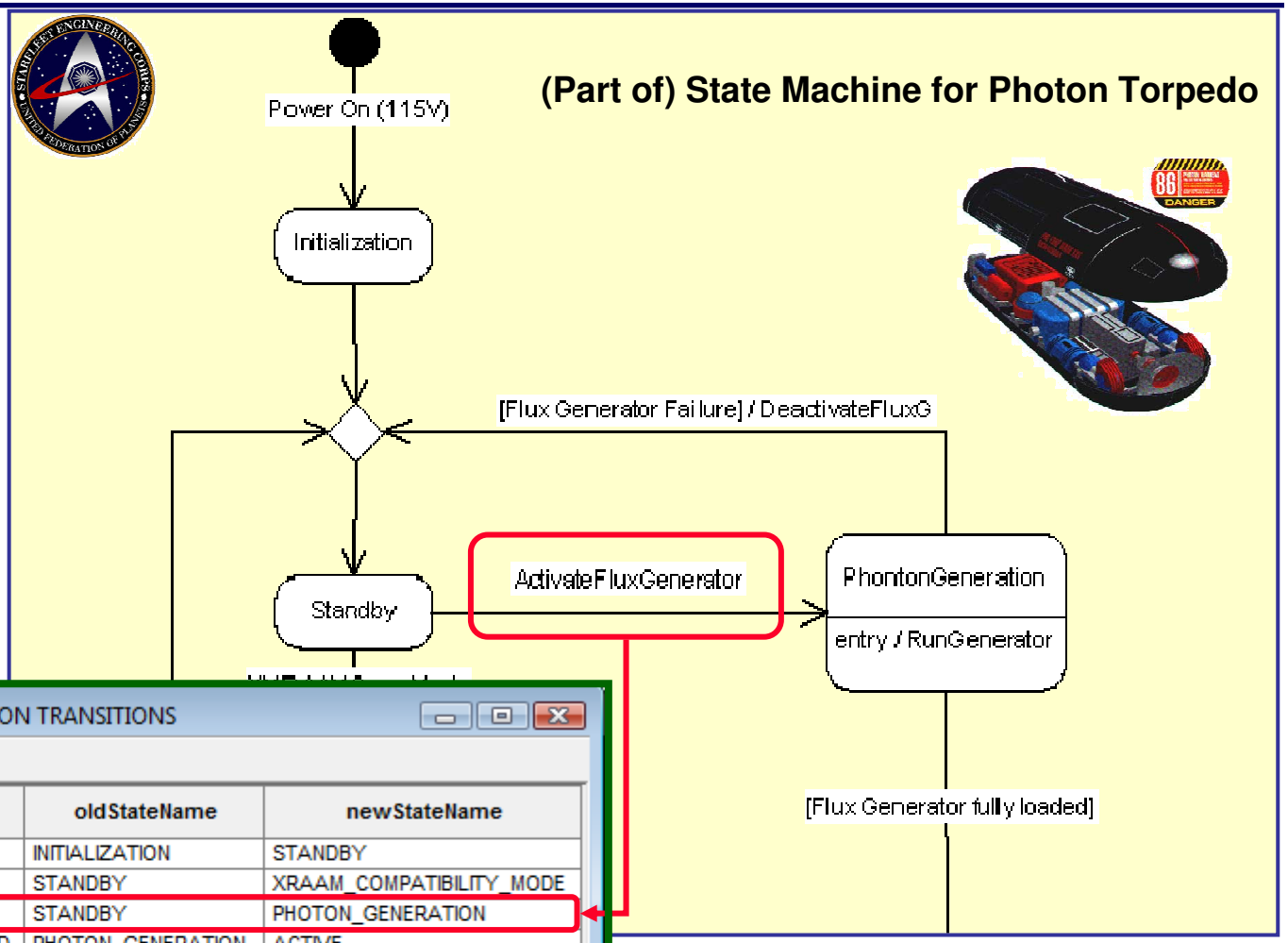
4: DataSet Table for Input Message Causing Transition::AMRAAM Transitions

	weaponTypeName	inputMessageName	oldStateName	newStateName
1	AMRAAM	DISCOVERY_COMPLETE	DISCOVERING_STORE	POWER_ON_BIT
2	AMRAAM	BIT_TIMER_EXPIRED	POWER_ON_BIT	DETERMINING_BIT_RESULT
3	AMRAAM	POWER_OFF	DETERMINING_BIT_RESULT	OFF
4	AMRAAM	PREPARE_FOR_LAUNCH	OFF	INITIALISING_EFFECTOR
5	AMRAAM	SEEKER_INITIALISED	INITIALISING_EFFECTOR	PREPARING_FOR_LAUNCH
6	AMRAAM	LAUNCH_DEMAND	PREPARING_FOR_LAUNCH	LAUNCHING
7	AMRAAM	SAFE_SEPARATION	LAUNCHING	POST_LAUNCH_GUIDANCE
8	AMRAAM	ENGAGEMENT_TIMER_EXPIRED	POST_LAUNCH_GUIDANCE	ENGAGEMENT_COMPLETE
9	AMRAAM	PERFORM_BIT	OFF	POWER_ON_BIT



Generic State Machines and Messages – Adding A New Effector





4: DataSet Table for Input Message Causing Transition::PHOTON TRANSITIONS

	weaponTypeName	inputMessageName	oldStateName	newStateName
1	PHOTON TORPEDO	INITIALIZATION_COMPLETE	INITIALIZATION	STANDBY
2	PHOTON TORPEDO	XRAAM_COMP_MODE	STANDBY	XRAAM_COMPATIBILITY_MODE
3	PHOTON TORPEDO	ACTIVATE_FLUX_GENERATOR	STANDBY	PHOTON_GENERATION
4	PHOTON TORPEDO	FLUX_gENERATOR_FULLY_LOADED	PHOTON_GENERATION	ACTIVE
5	PHOTON TORPEDO	ASSIGN_TARGET	ACTIVE	TARGET_ASSIGNED
6	PHOTON TORPEDO	DEACTIVATE	ACTIVE	STANDBY
7	PHOTON TORPEDO	TARGET_REJECT	TARGET_ASSIGNED	ACTIVE
8	PHOTON TORPEDO	NO_TARGET	TARGET_ASSIGNED	ACTIVE
9	PHOTON TORPEDO	LAUNCH_TORPEDO	TARGET_ASSIGNED	FREE_FLIGHT
10	PHOTON TORPEDO	FLUX_GENERATOR_FAILURE	PHOTON_GENERATION	STANDBY

Workshop on Distributed Object Computing for Real-time and Embedded Systems
July 14 – 16, 2008, Washington, DC, USA

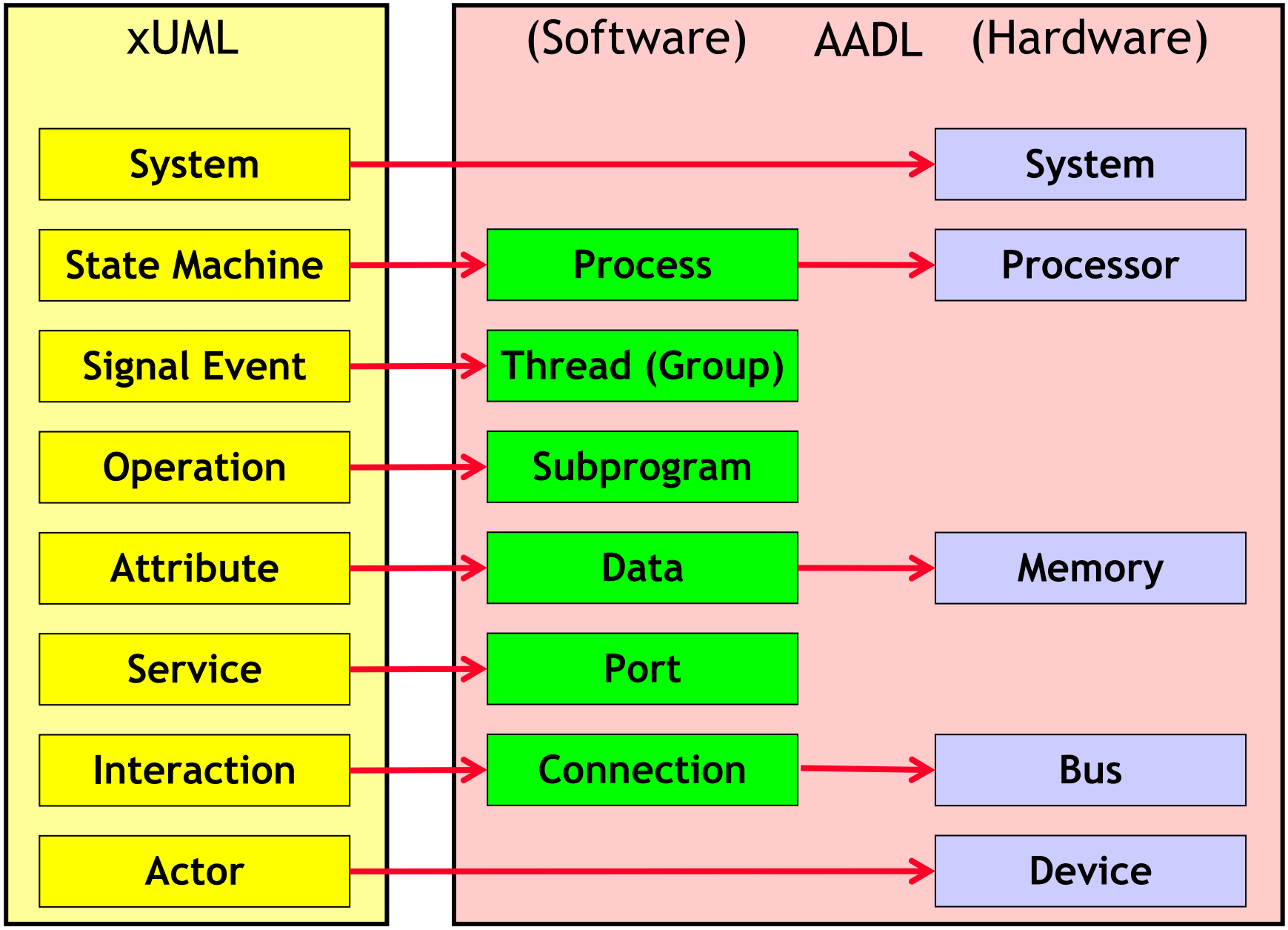
Plug and Play Implementation

KENNEDY CARTER

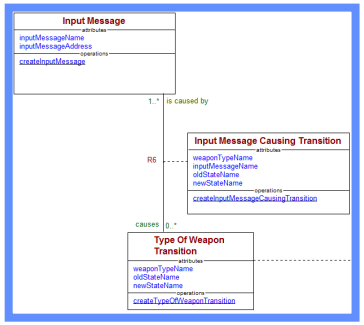
COMPLEXITY SIMPLIFIED

KC.COM

Mapping PIM (xUML) to PSI (AADL)

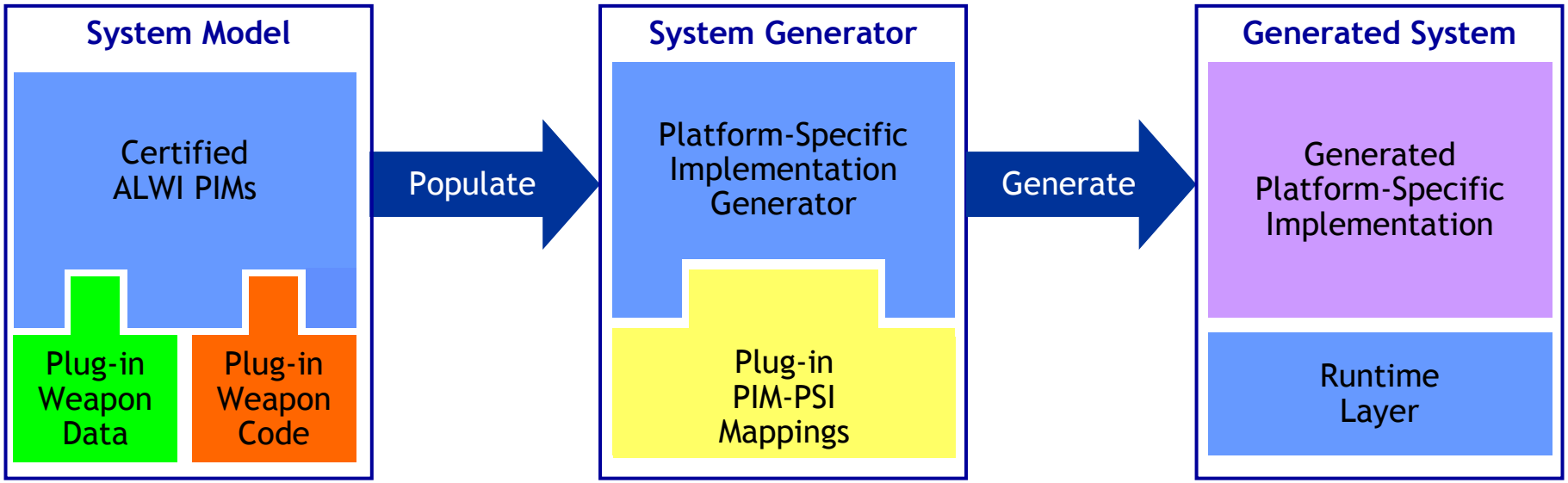


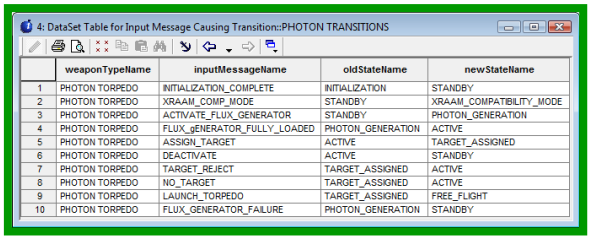
The ALWI Process



The ALWI process can be summarised as…

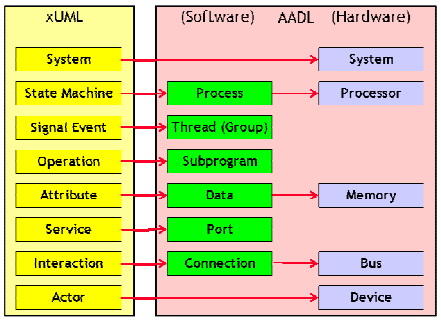
- Build and test a system model, using a precise modelling formalism
- Populate the code generator with the system model and platform-specific configuration, and generate the target system

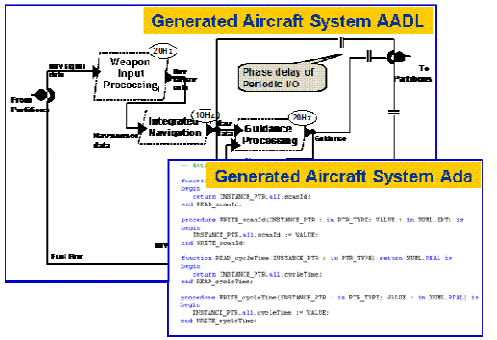




4: DataSet Table for Input Message Causing Transition::PHOTON TRANSITIONS

	weaponTypeName	inputMessageName	oldStateName	newStateName
1	PHOTON TORPEDO	INITIALIZATION_COMPLETE	INITIALIZATION	STANDBY
2	PHOTON TORPEDO	XRAAM_COMP_MODE	STANDBY	XRAAM_COMPATIBILITY_MODE
3	PHOTON TORPEDO	ACTIVATE_FLUX_GENERATOR	STANDBY	PHOTON_GENERATION
4	PHOTON TORPEDO	FLUX_GENERATOR_FULLY_LOADED	PHOTON_GENERATION	ACTIVE
5	PHOTON TORPEDO	ASSIGN_TARGET	ACTIVE	TARGET_ASSIGNED
6	PHOTON TORPEDO	DEACTIVATE	ACTIVE	STANDBY
7	PHOTON TORPEDO	TARGET_REJECT	TARGET_ASSIGNED	ACTIVE
8	PHOTON TORPEDO	NO_TARGET	TARGET_ASSIGNED	ACTIVE
9	PHOTON TORPEDO	LAUNCH_TORPEDO	TARGET_ASSIGNED	FREE_FLIGHT
10	PHOTON TORPEDO	FLUX_GENERATOR_FAILURE	PHOTON_GENERATION	STANDBY



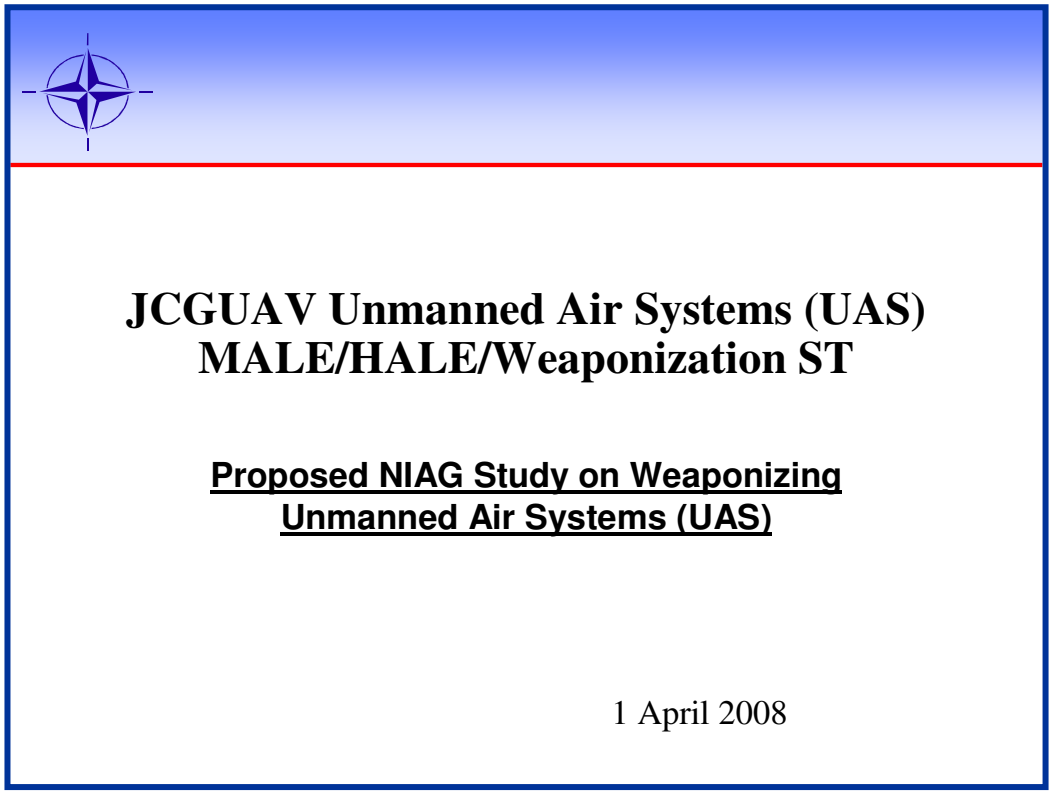


Summary

- 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…



Workshop on Distributed Object Computing for Real-time and Embedded Systems
July 14 – 16, 2008, Washington, DC, USA

The End

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

16 July 2008

Chris Raistrick, Kennedy Carter chris.raistrick@kc.com

KENNEDY CARTER

COMPLEXITY SIMPLIFIED

KC.COM