A Product Line Architecture for Army Aviation Diagnostics and Maintenance: Views and Evolution

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA  15213

Sholom Cohen
Ken Capolongo, CE-LCMC
May 2007

Contents / Agenda

Advanced Multiplex Test System
Product Line Concepts
  • Definition
  • Application to AMTS
Architecture Views
  • System Context
  • Software
Evolution
  • Product line growth
  • Tele-maintenance
  • Condition-Based Maintenance
AMTS Product Description

A line of off-board (non-embedded) diagnostic products to:

- support Army and Joint Aviation weapon platform maintenance
- assist maintainers
- support all maintenance levels (flight-line, intermediate, and depot)
- diagnose and repair avionics data bus networks faults

1553B System for AH-64A
AMTS Business Goals

Improve productivity of maintenance activities and eliminate false diagnoses
  ✓ Increases aircraft readiness
  ✓ Reduces aircraft operation and sustainment (O&S) costs
Meet anticipated product demand with current resources
Reduce product development time and cost
Minimize customer’s cost of entry
Provide products for various applications /maintenance levels
  • Operational: Aviation or Vehicle Platforms
  • Intermediate: Line Replaceable Unit (LRU)
  • Depot: from LRU to card level
Support collaborative (i.e., tele-) maintenance

What’s a Product Line?

A software product line is defined as
  • “A set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.”
Product Line Features and Variations

Platforms supported
- Helo
  - Apache A
  - Apache D
  - UH-60
- Aircraft
  - F-16
  - C-17 (under discussion)
- Ground vehicles
  - M1A1 Abrams
Variations within each by tail # or other vehicle feature

Bus
- Type
  - 1553 (1553A, 1553B)
  - Ethernet
  - 1773
  - ARINC (commercial aircraft)
  - CAN? (automotive)
- Architecture
  - Single bus
  - Multiple bus
- Single type
- Mixed type
- Nested (i.e., bus within bus; e.g., JTRS)

Message handling
Message analysis
- From text
- From XML
Message DB
XML translator
- Read
- Write (for sim/stim tool)

Test level
- Operational
- Intermediate
- Depot
- Development

Variation Example – Maintainer Interface
How AMTS Was Created

Leveraged years of experience supporting avionics maintenance and developing software/hardware tools
Exploited MIL-STD-1553 commonality
Established management support to develop prototype
Mined existing assets
Incorporated iterative/reactive Product Line engineering approach

AMTS Decomposition View

Core Environment - common modules across the product line
Product Configuration - modules that tailor core environment for specific platforms

Key:
- System
- Subsystem
- Module
Core Environment Decomposition View

Common assets for building testing systems for specific aviation platforms, systems, and subsystems

Core Environment (CE)

Consists of Console and Common Interface Driver (CID) modules

- **Console**: data driven software utilizes
  - Non-specific data and graphics to provide a universal data bus analyzer
  - Specific data, graphics, and software modules to provide platform data bus specific diagnostics
- **CID**: software adaptor utilizes
  - Vendor libraries to provide device communication independent of manufacturer

Common assets include a 1553 data bus generic monitoring system without platform specific data (Bus Tester Tool Kit).

- Every possible LRU on data bus, up to 32
- Specific LRU’s unknown. Where located on bus unknown
- Configuration capability to be added in future with tools to build plug-ins
Product Configuration (PC)

Product core assets and product specifics
- employ reverse engineering of platform system to understand behavior for database system information and message repository
- Use Multiplex ICD to identify product-specific messages
- Use Bus controller SRS to understand behavior

Product modules
- Core environment asset instances (console and CID)
- Product core (PCore) & PCore instances
- Product specifics (e.g., Read Codes for AH-64A)

AMTS Layered Context

Core Environment

Product

Any Aviation platform (generic)

Aviation platform (specific)

Cabling

AMTS Core Environment

AMTS Instance (contains CE, product core, and product specifics)
Layer View – Core and Product Configuration

Core Environment

Behavior view on next Slide

Sequence Diagram for Get Message Stack

User
Monitor
Server
Generic Bus Data
Device Adapter
Vendor Library
Device Buffer

Click
Start Snapshot
Start Timer
Stop Snapshot
Buffer data
Formatted data for user

User
Monitor
Server
Generic Bus Data
Device Adapter
Vendor Library
Device Buffer

Get buffer
Return buffer

© 2007 Carnegie Mellon University
Evolution – Product Line Growth

AMTS Product Line

- Console team builds assets:
  - Console assets

Product Configuration Team builds products:

- Product Core
- Product Specific

Uses:
- CID assets

SLV Product Line

- Builds loader/verifiers for:
  - IDM ARC220
  - EGI

Uses:
- AMAT Product Line

- Builds LRU tester for:
  - IDM ARC220
  - EGI

CID Team

- Builds assets:
  - CID Assets

Evolution – Tele-maintenance

Remote Client

- Tablet PC

Core Environment (CE):

- Console
- Monitor
- Server
- Multi-File-ID

- CDU
- Controller
- Configuration

- Multi-Media
- Multimedia

Common Interface Driver (CID):

- Device Adapter
- Vendor Library

Key:
- Subsystem
- Module
- Sub-Module
- Repository

Vendor

Collaboration Application (SME)
Tele-maintenance Support

Evolution – Condition Based Maintenance

Need expressed by maintainers in field
Aviation Responsive Maintenance System (ARMS) to consolidate and provide situational awareness picture
• of aircraft readiness for use in CBM
• of current data state across the fleet
• for playback on AMTS (collect data (one-hour’s worth) and the SME who is providing assistance has it for analysis.

Use existing built-in analysis capabilities as model to determine how to retrofit this in off-board for all aircraft

Product line approach is an enabler
Evolution for CBM

Change Device Adapter: add streaming output for record/playback to current buffered output for snapshot.

Evolution – Changes to Behavior for CBM

Click
Start Snapshot
Start Timer
Stop Snapshot
Buffer data
Formatted data for user

Start Streaming
End Streaming

Get buffer
Return buffer

Click
Start Snapshot
Start Timer
Stop Snapshot
Buffer data
Formatted data for user
Summary

AMTS architecture supports multiple and growing list of diagnostic/maintenance products

Changes managed through architecture evolution

- Identification of new goals (e.g., adding CBM capability)
- Technology changes (e.g., secure collaboration support)

Variation mechanisms not covered in this presentation – possible topic for working group discussions