Aerospace Domain Architectures

David Sharp
david.sharp@boeing.com
Senior Technical Fellow
Boeing Defense, Space, and Security

2014 National Science Foundation
Cyber Physical Systems Reference Architectures Workshop

26 March 2014

Outline

- Aerospace Architecture Domains
- Product Lines
- Reference Architectures
- Tools
- Summary / Takeaways

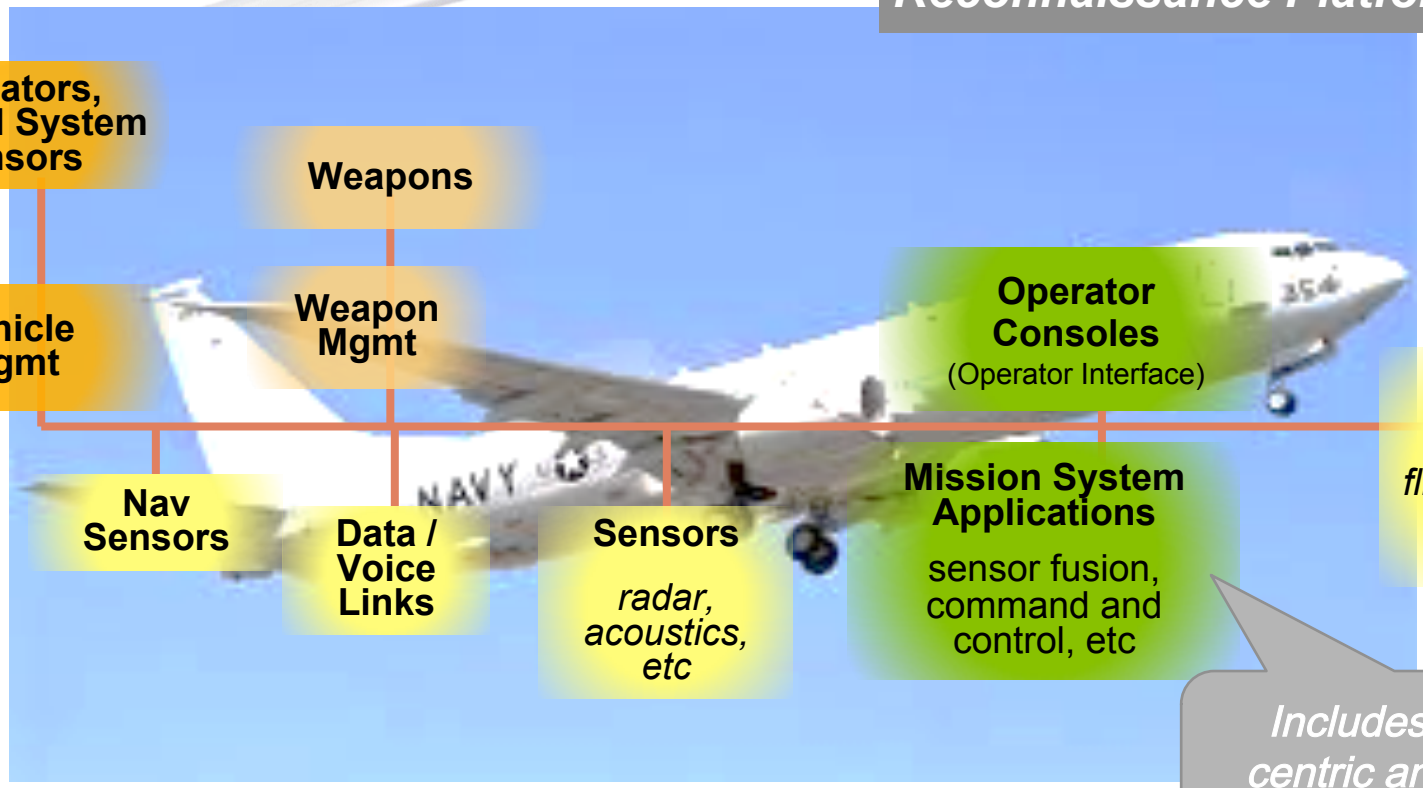
Key Aerospace Architecture Domains

Multiple Architectures Are Needed To Meet Varying Aerospace System Qualities

Arch. domain	**vehicle mgmt**	**cockpit / flight deck mission mgmt**	**crew station mission mgmt**	**non-real-time**
Exemplar Functionality	Flight controls	Flight Management	Reconnaissance	Mission Planning
User needs	*Smooth fault-transparent operation* (DO-178C, MIL-STD-882)	*Smooth fault-recovered operation*	*Up-to-date fault-recovered replayable operation*	*Up-to-date operation*
Driving architecture qualities	Safety Critical, Hard Real-Time, Single or Dual Fault Transparent	Mission Critical, Hard Real-Time, Sub second Frames and Fault Recovery	Mission Critical, Soft Real-Time, ~Seconds Fault Recovery, Training (e.g., pause, resume, record, replay)	Non-Real-Time
Reference architecture domain	Cyclic, synchronized multi-channel processing with redundant inputs/ outputs	Cyclic, hot standby/ backup-based fault recovery	Data-driven execution with recording, cold or hot fault recovery	Highly flexible (e.g. Web Services)

Architecture Domain Application Example

Reconnaissance Platform Example



Different Architecture Domains & Architectures Apply To Different Subsystems, And Sometimes Within Them

Sample Aerospace Product Line: Boeing Bold Stroke

Object-Oriented Reusable Application Framework

Architecture Specific Service Software

- Configurable to variable hardware configurations
- Supportive of reusable applications

- COTS-based real-time middleware services

Standard Service Software

Infrastructure Services

Operating System

Board Support Package

Hardware (CPU, Memory, I/O)

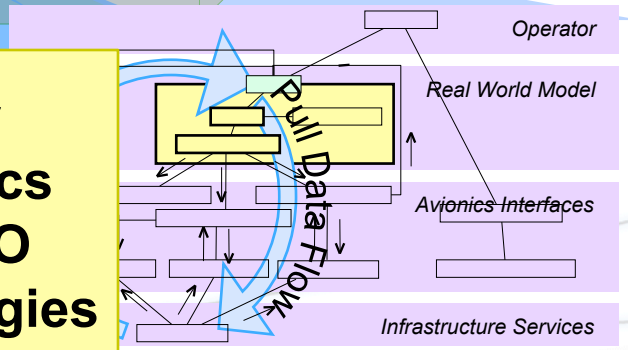
Commercial and Standar Platform

Product Line Component Model

Single Component Development Policies

- Configurable for Product Specific Functionality and Execution Environment
- Configurable for Product Specific Component Selection and Distribution Environment

Component Integration Policies

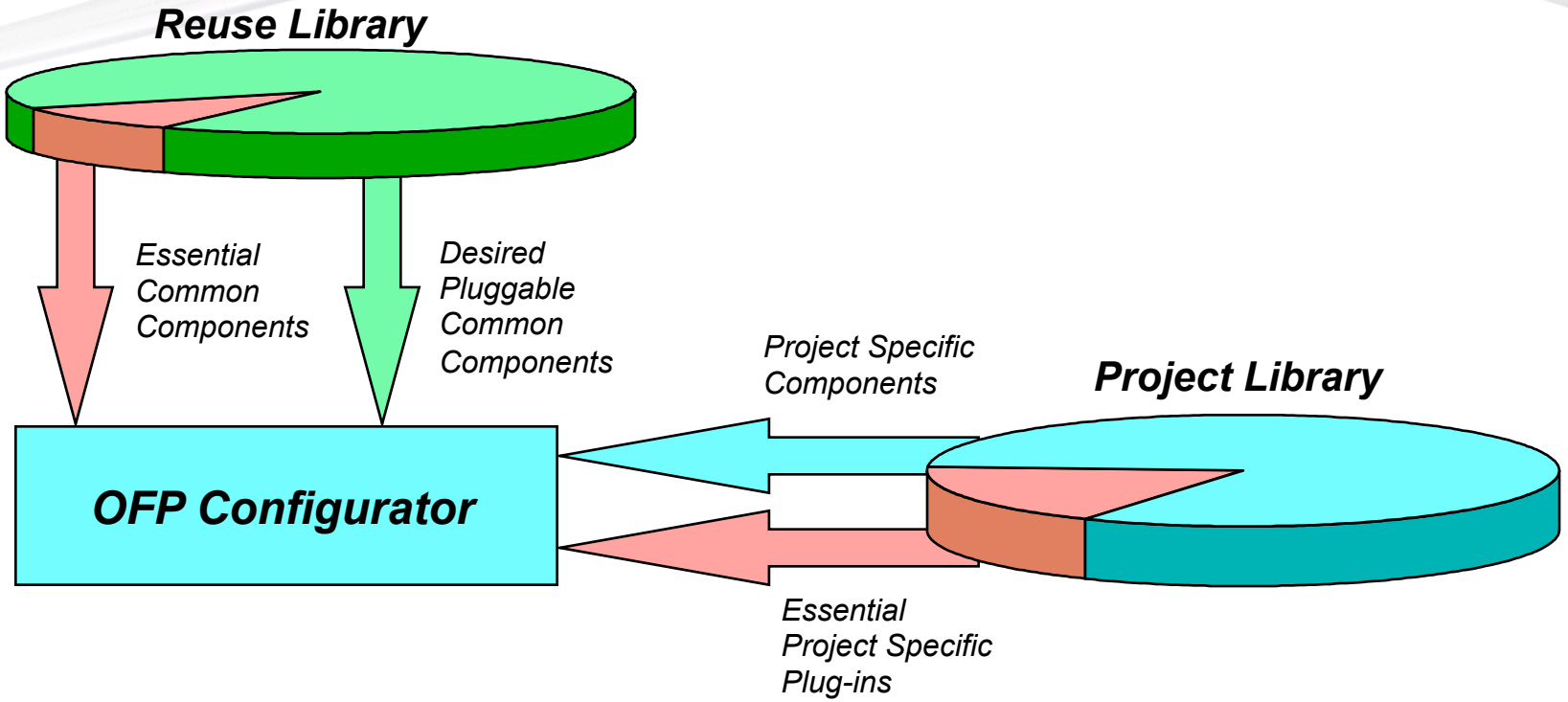


Component Integration Model

Creation of Highly Configurable Avionics Product Line Via OO Framework Technologies

Component-Based Product Line Production System

Bold Stroke



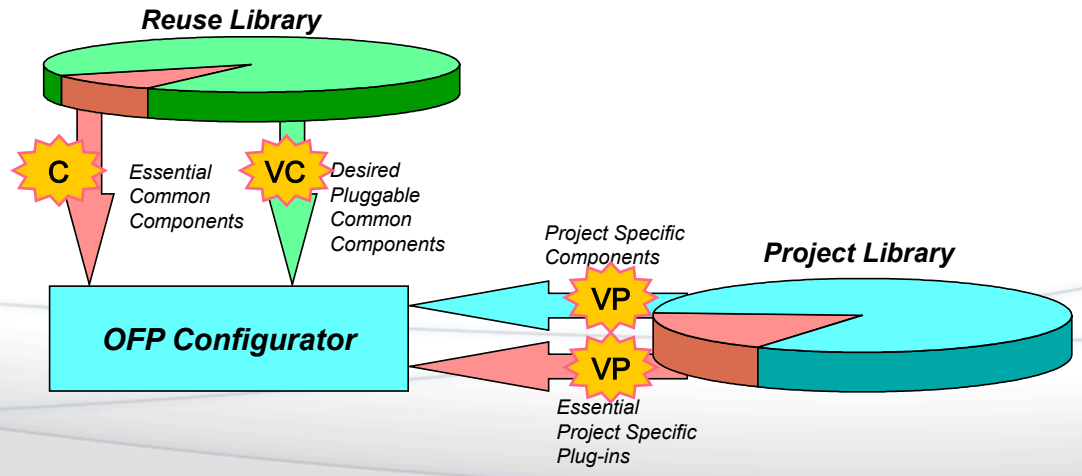
Flight Programs Created Via Component Selection and Wiring

Reference Architectures For Product Lines

- **In A Product Line Context, Reference Architectures Include:**
  - C – Identification of specific common invariant elements
  - – Identification of representative variable product specific elements, e.g.,
    - VC • product specific configurations of common elements, or
    - VP • product specific elements

What elements do all compliant systems require?

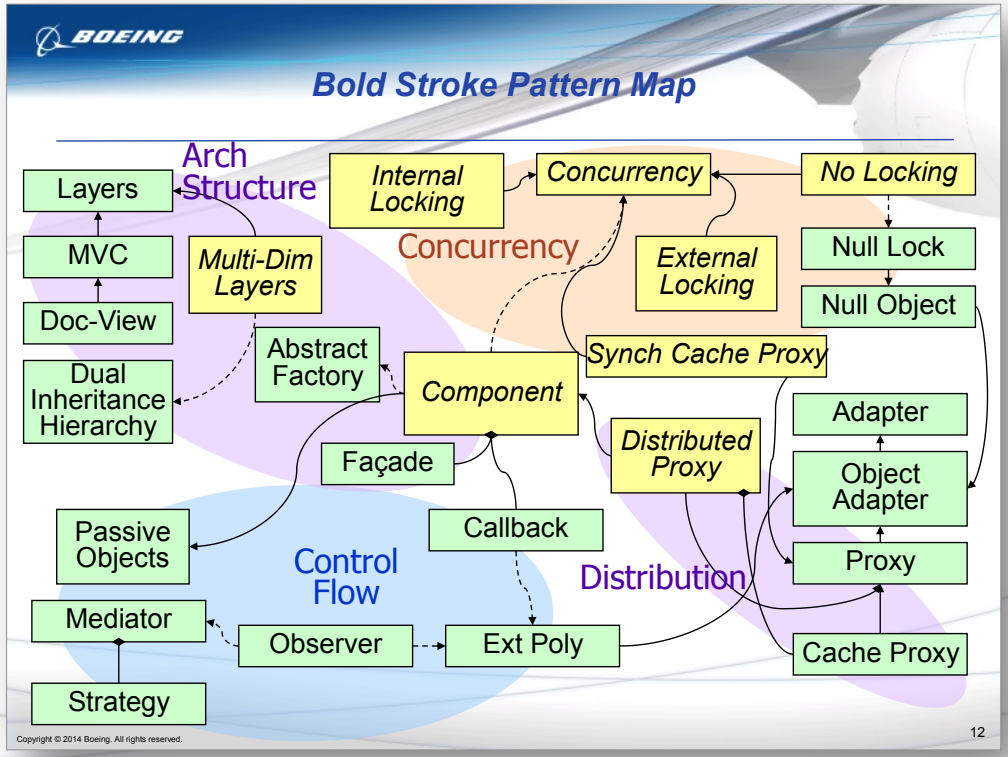
What are representative elements which differ between systems?



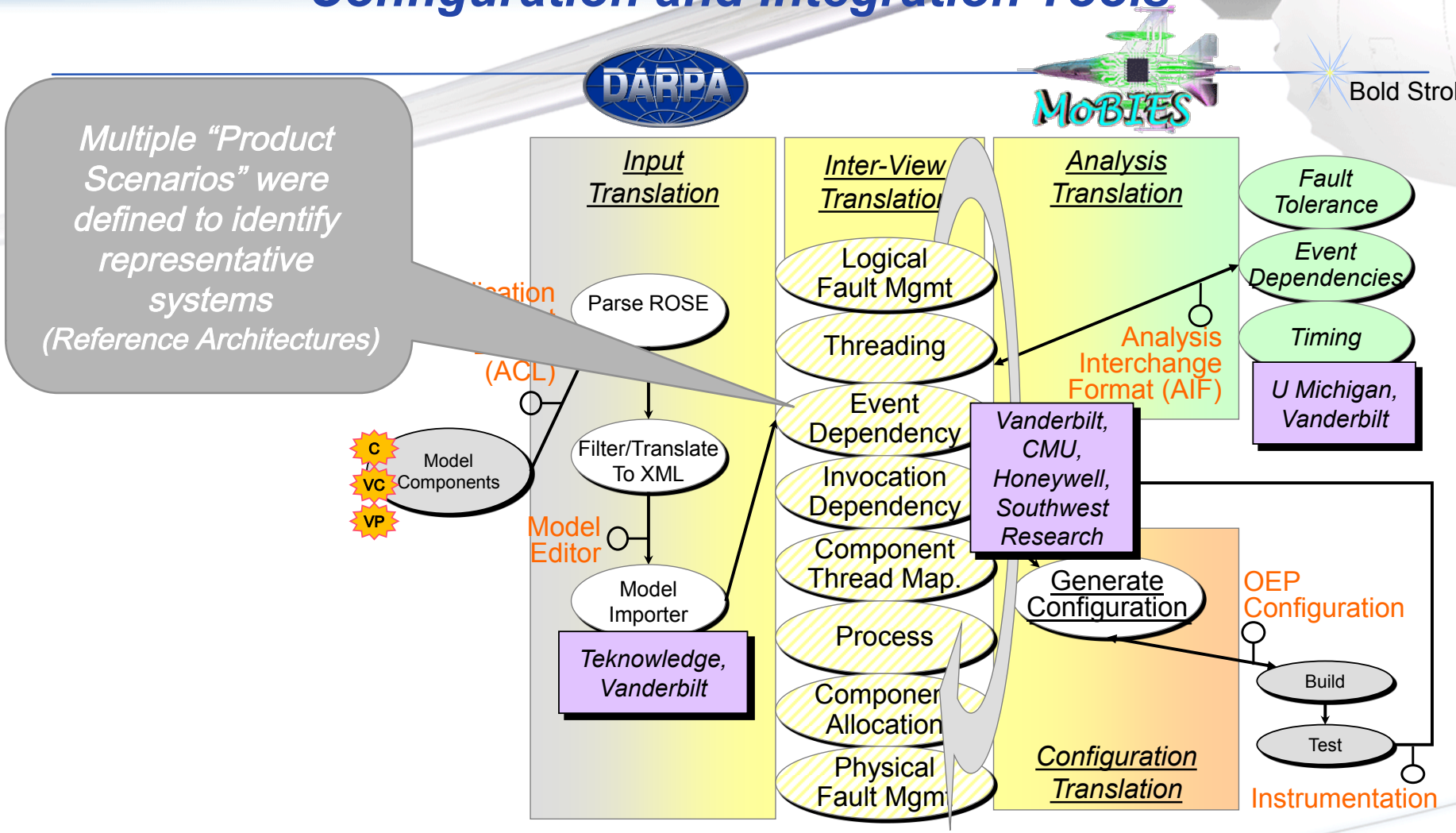
Reference Architectures For Product Lines (cont)

Bold Stroke

- **Beyond Hardware/Software Elements, Product Line Reference Architectures Should Identify Common And Variable:**
  - Product-focused items:
    - Architecture rules
    - Patterns
    - Interfaces
    - …
  - Process-focused items:
    - Tools
    - Config management
    - ...



Configuration and Integration Tools



Configuration & Integration Tools Can Support More Precise Component-Based Reference Architecture Definitions

Summary / Takeaways

- **Aerospace Systems Incorporate A Wide Range Of Architecture Domains And Architectures**
  - A family of – potentially interoperable – CPS Reference Architectures will be needed
- **Recommend Using Product Line Perspective For CPS Reference Architecture Definitions To, e.g.,**
  - Increase potential reuse
  - Specificity /clarity of reference architecture description
  - Reduce integration risk
- **Recommend Inclusion Of Process-Related Concerns In CPS Reference Architecture Definitions**
  - E.g., tools