Rapid Deployment of Aerospace Flight Controls

Edward L. Burnett

Lockheed Martin Aeronautics Company - Palmdale
Lockheed Martin Senior Fellow
Modeling, Simulation and Controls

Presented to:
The MathWorks Aerospace and Defense Conference
14 June, 2006
Rapid Deployment of Aerospace Flight Controls

- The Historic Problem
- A Possible Solution
- A Case Study
- Summary
The Problem

- Augustine’s Laws
  - (Law XVI)
    - “In the year 2054, the entire defense budget will purchase just one aircraft”
  - Software’s Part (Law XVII)
    - “Like Entropy”
      - “Weight nothing”
      - “Obeys the 2nd Law of Thermodynamics; i.e., its always increasing”
The Problem

Major Acquisition Process

AMRAAM 1977 - 1991
ATF 1983
YF-22 1st Flt 1991
F-22A 1st Flt 1996 IOC 2005
A-12 1983-1991
JAST 1994-1996
AFX 1991-1994
JSF X-35 1st Flt 2001 F-35 IOC 2008

18 Years

Airframe Service Life

A-12 1983
JSF Selection 2001
IOC 2008
F-35 2001

B-52A 1950
B-52H 2001
U-2A 1960
U-2S 2008
XC-130 1970
C-130J 2001

Edward L. Burnett – MathWorks ADC
Lockheed Martin Aeronautics Company – Palmdale (PIRA#AER200605002)
The Problem

- However, We Live in a Rapidly Changing World

- 10/4/57
- 11/9/89
- 9/11/01
- 2010

The Problem

- Cost Growth of Flight Controls

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1940</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1960</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1970</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Future Possible Trends

Development Cost $
The Problem

Integration Issues

- Interface Control Documents (ICDs)
  - Variable Name
  - Data Type
  - Units
  - LSB, MSB
  - Update Rates
  - ...
Looking For Trends in History

Distribution of Severity 1&2 Problem Products
(Based on Origination Product Data)
The Problem

- Software Only
  - Exec, IOC, DSP
- Classical
  - CLAWS, Air Data
- MBD

Factor of 7 Times More Errors From Classical Process
The Problem

- Cost of Software and ICD Fixes

![Graph showing the cost of different phases of a program (SRR, PDR, CDR, TRR, FFRR, FF, IOC) with Flight Test and IFAT highlighted.](Image)
The Problem

• New Challenges –
  – FAA
    • Reliability
      – Space and Time Partitioning
    • UAVs in the NAS
      – See and Avoid
A Possible Solution

HELD FOR SOFTWARE
A Possible Solution

- Model Based Design +
  - Design
  - Analyze
  - Deploy
  - Integrate
  - Test
  - Document
A Possible Solution

• Classical VMS System Development Process
A Possible Solution

• Model Based Design VMS System Development Process

ACG = Automatic Code Generation
A Possible Solution

- Model Drawing Standards
  - The Enabler
    - Learning Curve
    - Model Reuse
    - Scripts
    - Tools
    - Automated Test
Automatic Code Generation:
Template Files, Scripts
Create C-code and wrappers
A Possible Solution

- Use of Buses and Property Tags
  - Model Based ICD
  - Script Testing of Interfaces
A Possible Solution

- Control of Input and Outputs of Sub-Systems
  - Allows for Automated Test Vector Creation
- Simulation (PIL)
- HIL

Iron Bird Testing
A Case Study

• Lockheed Martin – A Recent Example
  – *Independently Funded Concept Demonstrator*
  – *Rapid Design-Build-Fly Program*
  – *Very Small Team*
A Case Study

• Rapid Simulation and Flight Software Development
  – Developmental / Analysis Simulation – <1 Month
  – Real-Time Piloted Simulation – <2 Months
  – HIL & Engineering Test Stand – 7 Months
  – SCO’s – 9 Months
  – Taxi – 12 Months
  – 1st Flight – 13 Months
A Case Study

• Only 2 Flight Controls Software Changes
  – *Calibration Tables (Scheduled Update)*
    • During SCOs
  – *Flight Test Data Output Update*
    • Ethernet to RS422
Summary

- Model Based Design +
  - Reduces Design Process Delays
  - Model Drawing Standards
    - Reduces Learning Curve
    - Increases Model Reuse
  - Automatic Code Generation
    - Reduces Manpower Required
    - Reduces Errors Early
  - Embedded ICD in Model
    - Reduces Errors in Documentation
    - Reduces Errors in Integration
  - Built in Test and Data Pump
    - Increases Test Efficiency

Reduces Total Time and Cost to Deploy
Rapid Deployment of Aerospace Flight Controls

Questions?

Never Forget Who You are Working For!