MathWorks Expo – October 2015
Achieving Certification for Safety Critical Systems
John Russell – Head of Systems and Software Engineering – BAE Systems Electronics Systems (UK)
October 2015
Electronic Systems

Agenda

• Introduction to BAE Systems Electronic Systems
• The challenge of achieving certification for safety critical systems
• Application of Model Based Design – why is it right?
• What is next?
• Conclusions
Electronic Systems (UK) Overview

- Electronic Systems is part of BAE Systems and reports into the US arm of the business
- The ES UK business is located in Rochester, Kent, England

- The site has 1600 employees
- Civil customers
- Military customers
Helmet Mounted Displays
Electronic Systems

Electronic Systems (UK) Overview

Head Up Displays
Flight Control Computers
Electronic Systems

Electronic Systems (UK) Overview

Active Inceptors
Electronic Systems

Electronic Systems (UK) Overview

HybriDrive™ Systems
What is Safety Critical Software?

• Safety Critical Software: Failure may have catastrophic consequences that causes injury or loss of life. E.g. Flight Control, Primary Flight Display

• Verification activities must demonstrate that the software meets its requirements under all foreseeable operating conditions

SAE-ARP-4754A
DO-178C / DO-331
DO-254
Electronic Systems

The Challenge

- Increasing competition within the industry
- Increased focus on process adherence
- Evolving standards
- How can we meet these certification challenges and cost/schedule challenges
- The use of Model Based Design is one way
- Generation of a backup flight control system implemented purely in PLDs – no processor
- Developed to DO-254 DAL-A
Lifecycle Comparison – DAL A Software Development
Electronic Systems

Lifecycle Comparison – DAL A Software Development

- Define System Requirements
- Define System Model in MATLAB/SIMULINK
- Verify Model to Requirements
- Perform Coverage Analysis
- Auto Review
- Auto Generate Code
- Perform Robustness Analysis
DO-178C MBD Workflow – Simple Approach

1. Textual Requirements Review
2. System Software Model Review
3. Source Code
4. Object Code

- Requirements Based Tests
- Run tests on model to gain model coverage
- Repeat tests source coverage
- Repeat tests object coverage
Electronic Systems

What Is Next?

- We have embraced model based design across the development lifecycle for high integrity software development. What can we further improve?

- Overall tool performance
- Utilisation of parallel computing resources
- Improved integration with other tools
- Level of subset support for the Simulink Code Inspector
- Reusable libraries
- Increased use of hardware in the loop systems
Conclusions and Benefits

- Applicable to DO-178C and DO-254
- Cultural change
- Whole lifecycle view – an integrated workflow

- Cost
- Schedule
- Quality
- Customer satisfaction