FM@COLLINS

FM@SCALE 25 SEPTEMBER 2019 ARLINGTON VA

DR. DARREN COFER TRUSTED SYSTEMS DARREN.COFER@COLLINS.COM



© 2019 Collins Aerospace, a United Technologies company. All rights reserved.



WHY SHOULD COLLINS AEROSPACE CARE ABOUT FORMAL METHODS?

- Satisfy certification objectives?
 - Maybe (see DO-333)
- Eliminate testing?
 - Some, but not all

By analyzing requirements, models, and software early in the development process, we can eliminate defects, reducing costly rework and even more costly escapes.

- Be the most trusted source of aviation and high-integrity solutions in the world?
 - Definitely!







THE REAL REASON WE WILL USE FORMAL METHODS

- To reduce costs
 - Automation of tedious reviews and testing that humans are terrible at anyway
 - Reduce cost of rework
 - Reduce cost of escapes







THE REAL REASON WE WILL USE قامد FORMAL METHODS

- To reduce costs
 - Automation of tedious reviews and testing that humans are terrible at anyway
 - Reduce cost of rework
 - Reduce cost of escapes

(Undetected design error that compromises safety)

The New York Times

Boeing Says Charges Tied to 737 Max Grounding to Reach \$8 Billion



Boeing 737 Max planes parked at the municipal airport in Renton, Wash. The Max planes have been grounded after two were involved in deadly crashes. Lindsey Wasson for The New York Times

By David Gelles

July 18, 2019



The financial fallout from the troubled <u>737 Max</u> jetliner continues to swell for Boeing, which on Thursday announced \$7.3 billion in costs that will hit its bottom line.



FM@COLLINS : SUCCESSES



DO IT ONCE

TURNSTILE / AAMP7 / SEL4

- Large, sustained effort by formal methods experts to produce a product or other reusable artifact
- AAMP7G microprocessor with intrinsic partitioning
- Turnstile cross-domain guard
- GH Integrity certification
- seL4 microkernel (Data61, FoC)





DO IT LOTS

GRYPHON / FSV

Model Layer Translation Model Parsin Intermediate representation FM tools usable by SW developers Intermediate processed by Formal ٠ Model Formulation Formal Problem Formulation (Model Model Checkers: Transformation Transforma NuSMV, Prover, Layer Intermediate Formal Input Model 2 Formal Problem BAT, Kind, SAL Model Optimization Simulink SCADE Simulink Mappings Gateway Formal Probler Optimization Full traceability between **Theorem Provers:** source and formal Reactis Lustre model ACL2, PVS Optimized Mos (XMI) Safe State Simulink StateFlow Machines Gateway Programming Model2Tex Model Checking State-Of-The-Art ransformatio Languages: Optimized Portfolio of Layer (nuXmv) SPARK (Ada), C Verification Engines ABC (AIG) Model nuXmv Model Rockwell Collins/U of Minnesota Design Esterel Technologies Verifier SRI International 787Electric Power Distribution LGEN ExtPwr APU RGEN MathWorks FT DU AVAILABLE \bigcirc • • ∿ LEFT_DU_AVAILABLE Reactive Systems 3-LEFT SWITCHBANK EICAS LEFT_DU_APPLICATION LEFT_DU_APPLICAT 吉 阜 宫 T SWITCHBANK PED LEFT_SWITCHBANK_PFD EFT SWITCHBANK MAI LEFT_SWITCHBANK_MAP 2 GHT DU AVAILABLE AC Bus L AC Bus R RIGHT_DU_AVAILABLE **O**-GHT SWITCHBANK EICA RIGHT_SWITCHBANK_EICAS RIGHT_DU_APPLICAT RIGHT_DU_APPLICATION TRU TRU SWITCHBANK PEE RIGHT_SWITCHBANK_PFD (B)-GHT SWITCHBANK MAP WM WM RIGHT_SWITCHBANK_MAP 阜 DU_APPLICATION_SELECTION _ DC Bus R DC Bus L 宫 EICAS MAP EICAS MAP PFD LEFT_DU_APPLICATION PFD RIGHT_DU_APPLICATION Batt Bus L Batt Bus F CURSOR_LOCATION ▶(3) 9 CURSOR_LOCATIO EFT MANUAL REQUEST Display Application Window Manager Batt (10)-GHT MANUAL REQUEST RIGHT_MANUAL_REQUEST CURSOR_PLACEMENT



© 2019 Collins Aerospace, a United Technologies company. All rights reserved.

UTAS / UTRC Formal Specs Verifier (FSV) for Formal Verification

Full MATLAB API-

based Model Parsing

Simulink/STATEFLOW

Verification Model

Specification Patterns

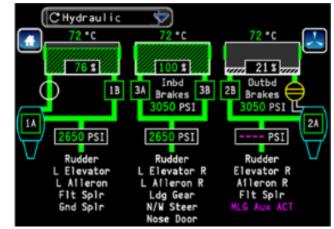
Environment Model

STATEFLOW Design Model

DO IT LOTS

CERTIFICATION TOOLS FOR COMMERCIAL AVIONICS





Custom domain-specific tools using FM to automate:

- extensive peer reviews of requirements, designs, and code
- manually generated structural tests
- process documentation

New methods use automated reasoning to improve today's labor-intensive methods



AUTOMATED CERTIFICATION TOOLS

Crew Alerting

- Automated design review and test generation with structural coverage of requirements
- First cert use Sept 2018 ٠
- Saved \$1.5M on M170 program

Flight Controls

- Custom IDE automates design review and test generation
- First expected cert use 2020

Aircraft Health Monitoring

- Automated review of generated • code using CBMC
- First expected cert use 2020



Property	Result
Inhibits	6 Invalid, 4 Valid
🕨 🔣 Global	2 Invalid, 24 Valid
Boolean_Out	1 Invalid. 34 Valid
🕨 🛷 Debug	3 Valid
🔻 💭 Avionics	1 Working. 66 Waiting, 4 Invalid, 3 Unknown, 89 Valid
ACP_1_FAIL_ADV_CAS	2 Valid
ACP_2_FAIL_ADV_CAS	2 Valid
ACP_3_FAIL_ADV_CAS	2 Valid
AURAL_INHIB_STAT_CAS	2 Valid
ADC_1_FAIL_ADV_CAS	1 Valid
♦ ✓ ADC 2 FAIL ADV CAS	1 Valid



 ∇

Inbd

0 PSI

0 PSI

Rudder

L Elevator R

L Alleron R

Ldg Gear

N/W Steer

38

3A Brakes

72 °C

21 5

PS1

-- PSI

Elevator R

Atleron R

Fit Spir

NLG AUX AC

Outbo

2B Brakes

CHydraulic

O PSI

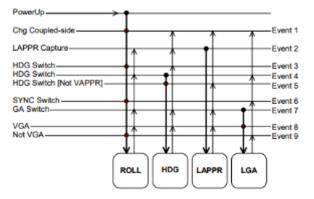
. Elevator

L Aileron

Fit Spir

Gnd Solr

1B



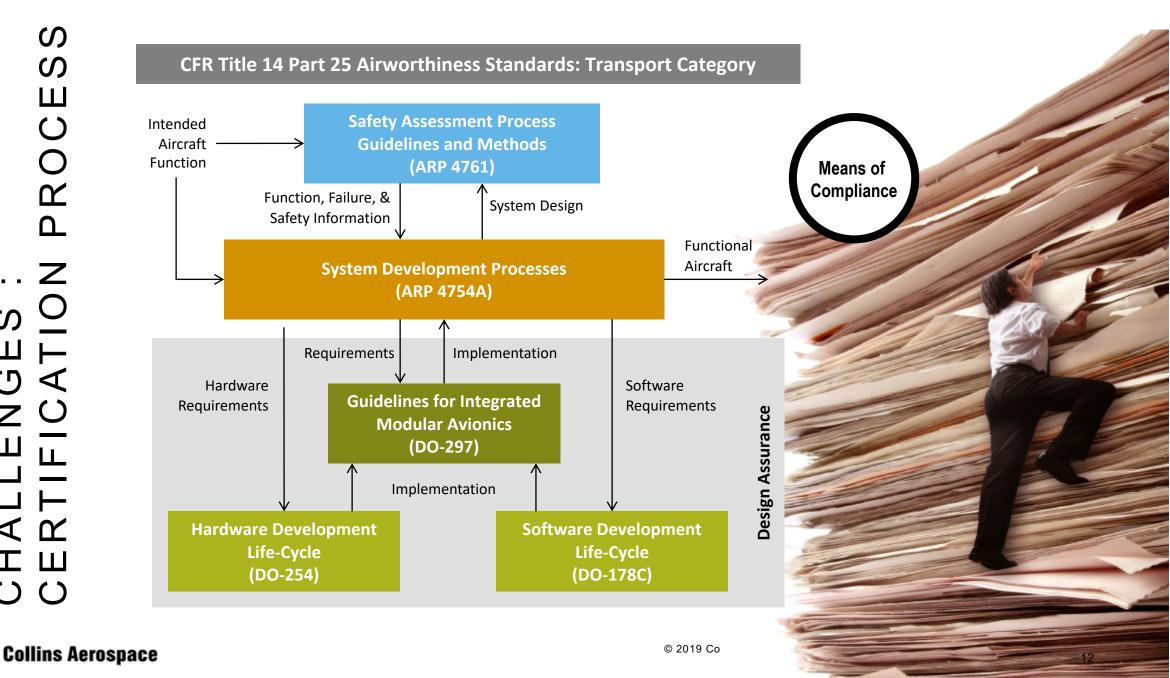




FM@COLLINS : CHALLENGES



© 2019 Collins Aerospace, a United Technologies company. All rights reserved. 11





FM@COLLINS : FUTURE

HOW CAN WE SCALE UP APPLICATION OF FORMAL METHODS FOR AVIONICS?



1 : DO IT LOTS MORE

DEVELOP MORE DOMAIN-SPECIFIC TOOLS

- Get really good at rapid customization of a standard set of analysis engines for new/different application areas
 - Maximum automation (low-hanging fruit)
 - Target existing domain-specific specifications
 - Evidence generation for certification objectives



2 : PROVABLE SYSTEM SAFETY/SECURITY

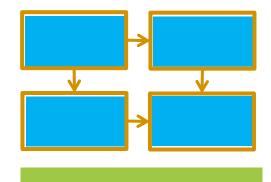
- Standardize on a system architecture language for integrating
 - System design
 - System safety
 - Cybersecurity
- Identify standard properties (requirements) needed for all aircraft/systems/components
 - With some configuration
- Architecture Analysis and Design Language (AADL)
 - Analogous to "infrastructure as code"



ARCHITECTURE-DRIVEN ASSURANCE

HACMS / CASE / AMASE APPROACH

- Architecture model is correct (AADL)
 - Properties, structure, behavior, interaction of components, interfaces, contracts
 - Verify system safety/security properties (in the presence of faults/threats)
- Components are correct
 - Consistent/realizable contracts
 - Components verified to implement contracts
- System does what the model says
 - Verified kernel (seL4)
 - No other information flows (memory safety, isolation)
 - OS executes model correctly (incl. timing)
- System implementation corresponds to model
 - Automatic build from component and architecture models with proof of compliance to architecture specification









Code, papers, videos available at:

Loonwerks.com

github.com/Loonwerks



