Better Defined Software

Requirements with

Model-based Systems

Engineering

Functional Safety Analysis and Control Tuning
• **Background**
  • 12 years control software design for DO-178B/C certified programs

• **Current assignments**
  • Consult a variety of clients on best practices for software architecture and verification strategies to satisfy DO-178C requirements using SCADE system and software solutions
Project Scope

The Cabin Pressure Control System (CPCS) is an avionics system designed to minimize the rate of change of cabin pressure.

The purpose of the CPCS is to ensure the safety of the airframe and passengers while maximizing comfort for aircrew and passengers during all phases of flight.
ARP4754A, ARP476,1 DO-254, DO-178C Standards
System Safety and Engineering Processes in ARP4761 and ARP4754A

Safety Assessment Process:
- FHA Functional Hazard Analysis
- CCA Common Causes Analysis
- PSSA Preliminary System Safety Assessment
- SSA System Safety Assessment

System Development Process:
- Aircraft Functions
- Allocation of Aircraft Functions to Systems
- Development of the System Architecture
- Allocation of System Requirements to Items
- System Implementation

Certification

Results

Implementation

Architecture Req.

Items Requirements

System Functions

Safety Requirements

Aircraft Functions
## DO-178C/DO-331 Model Usage Examples

<table>
<thead>
<tr>
<th>Process that generates the life-cycle data</th>
<th>MB Example 1</th>
<th>MB Example 2</th>
<th>MB Example 3</th>
<th>MB Example 4 (See Note 1)</th>
<th>MB Example 5 (See Note 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Requirement and System Design Processes</td>
<td>Requirements allocated to software</td>
<td>Requirements from which the Model is developed</td>
<td>Requirements from which the Model is developed</td>
<td>Requirements from which the Model is developed</td>
<td>Requirements from which the Model is developed</td>
</tr>
<tr>
<td>Software Requirement and Software Design Processes</td>
<td>Requirements from which the Model is developed</td>
<td>Specification Model (See Note 2)</td>
<td>Specification Model</td>
<td>Design Model</td>
<td>Design Model</td>
</tr>
</tbody>
</table>

**Note 1:** Design Model

**Note 2:** Design Model

**Note 3:** Textual description (See Note 3)
Objectives of the Approach

Global Product Data Interoperability Summit | 2018

• Take benefits from simulation at all levels:
  • System models for system analyses
  • Software models for software analyses

• Eliminate most of the low-level software verification activities

• Integrate the ARP4754A/ARP47161 and DO-178C/DO-331 modeling contexts for achieving the certification objectives in an efficient manner
Systems Requirements Definition

Global Product Data Interoperability Summit | 2018

- Systems Requirements were created in compliance to EARS (Easy Approach to Requirements Syntax) methodology.
  - The EARS methodology is an effective way of expressing requirements between five types (or patterns) to avoid defining poor requirements which can propagate to lower levels.

- System level requirements were defined for:
  - System Operating Modes
  - User Interface requirements
  - System Performance requirements
  - System Architecture requirements
### Requirements Examples from DOORS

#### 3 System Operating Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPCS-SYS-3</td>
<td>False</td>
</tr>
<tr>
<td>CPCS-SYS-4</td>
<td>False</td>
</tr>
<tr>
<td>CPCS-SYS-5</td>
<td>True, Test The Normal Automatic Mode shall be the active mode of the system upon system start-up.</td>
</tr>
<tr>
<td>CPCS-SYS-7</td>
<td>True, The Normal Automatic Control Mode shall be controlled by the IASC primary channel (Channel A).</td>
</tr>
<tr>
<td>CPCS-SYS-9</td>
<td></td>
</tr>
<tr>
<td>CPCS-SYS-12</td>
<td>True, When the MANUAL ON command is sent from the Control Panel, the Manual Control Mode shall be active.</td>
</tr>
<tr>
<td>CPCS-SYS-13</td>
<td>True, Manual Control Mode shall be controlled by the IASC back up channel (Channel B).</td>
</tr>
<tr>
<td>CPCS-SYS-14</td>
<td>True, While the Manual Control Mode is active, the IASC primary channel (Channel A) shall be inoperative for control commands.</td>
</tr>
<tr>
<td>CPCS-SYS-15</td>
<td>True, While the Manual Control Mode is active, the IASC primary channel (Channel A) shall be active for data monitoring.</td>
</tr>
<tr>
<td>CPCS-SYS-16</td>
<td>True, While in Manual Control Mode, the system shall allow manual control of the cabin altitude rate of change.</td>
</tr>
</tbody>
</table>
Architecture Definition

Global Product Data Interoperability Summit | 2018

Allocation of Functions to Software and Platform Components

**Functional Architecture**

**Software Architecture (ARINC 429)**

**Software Architecture (ARINC 664-P7 / AFDX)**

**Platform (ARINC 429 and ARINC 664-P7 / AFDX)**
Architecture Definition

From system requirements to architecture and operational representation
Functional Decomposition of Operational Requirements
Architecture Definition (Initial)
Architecture Definition

Global Product Data Interoperability Summit | 2018

Software Architecture Perspective – ICD Generation

- SoftwareArchitecture
  - SWMessageDefinition
    - A429Labels
      - Types (4)
        - A429_ALT
        - A429_CABINPRESSURE
        - A429_TEMPERATURE
        - A429_SET_TEMPERATURE
      - Tables (1)
        - A429MessageDefinitionTable
  - AFDXLabels
  - GenericSWMessageDefn

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>LabelID</td>
<td>Field Type</td>
<td>Position</td>
<td>Size</td>
<td>Bit29mean0</td>
<td>Bit29mean1</td>
<td>Encoding</td>
<td>EquipmentID</td>
<td>SDIRole</td>
<td>SSMatrixLength</td>
</tr>
<tr>
<td>1</td>
<td>A429_ALT</td>
<td>203</td>
<td></td>
<td>positive sign</td>
<td>negative sign</td>
<td>BNR = 1</td>
<td>260</td>
<td>data = 0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>MessageFields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ALTITUDE</td>
<td>real</td>
<td>11</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A429_CABINPRESSURE</td>
<td>152</td>
<td></td>
<td>positive sign</td>
<td>negative sign</td>
<td>BCD = 0</td>
<td>98</td>
<td>data = 0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>MessageFields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CABINPRESSURE</td>
<td>real</td>
<td>11</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A429_TEMPERATURE</td>
<td>187</td>
<td></td>
<td>positive sign</td>
<td>negative sign</td>
<td>BCD = 0</td>
<td>88</td>
<td>data = 0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>MessageFields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TEMPERATURE</td>
<td>real</td>
<td>4</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A429_SET_TEMPERATURE</td>
<td>187</td>
<td></td>
<td>positive sign</td>
<td>negative sign</td>
<td>BCD = 0</td>
<td>88</td>
<td>data = 0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>MessageFields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TEMPERATURE</td>
<td>real</td>
<td>4</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Functional Safety Analysis with medini

medini provides an integrated model-based approach to functional safety of aerospace systems

Development and safety assessment according to ARP4754A and ARP4761 are supported

Major safety analysis methods are provided: FHA, FTA, FMEA, FMES, CCA

Reliability prediction and quantitative analysis are supported
Safety Analysis

Global Product Data Interoperability Summit | 2018

- The Architecture model from SCADE Architect is imported into medini for safety analysis
- The safety analysis follows the ARP4754A phases/activities
- It is done iteratively and integrated with the system development
Functional Safety Analysis

• The System FHA leads to Failure Conditions

• The Failure Conditions are analyzed with FTAs

• Safety Requirements are derived

• Preliminary Architecture is updated

• Traces are established
Functional Safety Analysis – System Architecture Update
Global Product Data Interoperability Summit | 2018
Control Laws Design

Objectives
- Define requirements allocated to Software: DO-331

Workflow
- Model the Physical System
- Generate a Reduced Order Model of the Physical System
- Construct the System Model with Twin Builder
- Extract State Space Equations to design the Controller (Linearization)
- Validate & Optimize Controller requirements within the system model
- Allocate system requirements to Software
Model the Physical System

- ANSYS Fluent captures the 3D air flow within a MD-82 cabin
- Virtual sensors distributed spatially in the cabin are used to monitor temperature and pressure
- CFD analysis is generated for the entire cabin
  - computation time = several days
Generate Reduced Order Model

Global Product Data Interoperability Summit | 2018

- 3D models are extremely precise but require large CPU time
- ANSYS unique technology (deep learning) enables the creation of a Reduced Order Model for transient non-linear 3D models

- Resulting Performance improvement:
  - Fluent - 40 hours on 48 CPU
  - ROM - 4 seconds on 1 CPU
Construct a High-Fidelity System Model with Twin Builder

- ANSYS Twin Builder models the system components (actuators, sensors, etc.) with the ROM for a complete system simulation.

- This enables optimization and validation of component choices with the system response.
- Twin Builder can automatically extract a continuous linear State-Space model from any arbitrarily complex system model.

\[
\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \\
y(t) = C \cdot x(t) + D \cdot u(t)
\]
Perform Initial Controller Tuning

Global Product Data Interoperability Summit | 2018

• A Control Toolbox is used for the basic control law design
  • Analyze properties of the linear system identified in the previous step
  • Design and tune the controller
    – Select the form of the controller based on properties of the system
    – Discretize the controller
    – Determine initial parameters values for the controller
Validate Controller Requirements within the System Model

- Integrate the controller in the high fidelity system model
- Twin Builder enables parametric analysis
  - On the basis of the controller definition designed from the state space equation, perform a sweep on specific controller parameters with closed-loop physics simulation
  - Select parameters for best performances
  - Check robustness to mission profiles and/or system changes

System responses to a range of controller coefficients
Allocate System Requirements to SW

- The System Modeling workflow converts our initial performance requirements into requirements for the software controller.
**Definition of Software High Level Requirements (HLRs)**

**Global Product Data Interoperability Summit | 2018**

- HLRs were created in compliance to EARS (Easy Approach to Requirements Syntax) methodology.

- **Example Software HLRs:**

<table>
<thead>
<tr>
<th>CPCS-SW-3</th>
<th>1.2 Pressure Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPCS-SW-9</td>
<td>The Pressure Controller shall use the following constants when determining the targeted cabin altitude:</td>
</tr>
<tr>
<td></td>
<td>ALT1: 20,000 ft</td>
</tr>
<tr>
<td></td>
<td>CAB_ALT1: 4,000 ft</td>
</tr>
<tr>
<td></td>
<td>OFV_DITCHING_SPEED_CMD: -20 deg/s</td>
</tr>
<tr>
<td></td>
<td>OFV_EMERGENCY_SPEED_CMD: 25 deg/s</td>
</tr>
<tr>
<td></td>
<td>OFV_NORMAL_SPEED_CMD: 10 deg/s</td>
</tr>
</tbody>
</table>

| CPCS-SW-18 | While the CPCs operating mode is 'AUTO_NORMAL', the OFV Speed Command shall be calculated using the PID controller using the following parameters: |
|            | Kp = -0.006 (s^-1)     |
|            | Ki = -0.02 (s^-2 ft^-1) |
|            | Kd = 0.0 (ft^-1)       |
|            | dT = 100 ms            |
|            | IoLimit = OFV_NORMAL_SPEED_CMD deg/s |

| CPCS-SW-19 | While the CPCs operating mode is 'MANUAL', the OFV Speed Command shall be calculated using the PID controller using the following parameters: |
|            | Kp = -0.006 (s^-1)     |
|            | Ki = -0.02 (s^-2 ft^-1) |
|            | Kd = 0.0 (ft^-1)       |
|            | dT = 100 ms            |
|            | IoLimit = OFV_NORMAL_SPEED_CMD deg/s |
|            | highLimit = - OFV_NORMAL_SPEED_CMD deg/s |
Our Example requirement has two parts:

• Part 1: Mode constraint
  "While the CPCS operating mode is ‘AUTO_NORMAL’…"

• Part 2: Directive for the output
  “…the OFV Speed Command shall be calculated using the PID controller…"
Software Verification: Example HLR

Test Case

LLR₁ (SCADE)

modeCommand

LLR₂ (SCADE)

AC Information

traces to

data flow

PressureController

OFV.SpeedCommandAuto

OFV_Speed Command shall be calculated using the PID

While the CPCs operating mode is 'AUTO_NORMAL', the controller using the following parameters:

Kp = -0.006 (s*ft)⁻¹
Ki = -0.02 (s⁻² * ft)⁻¹
Kd = 0.0 (ft)⁻¹
DT = 100 ms

 thoLimit = OFV_NORMAL_SPEED_CND deg/s
Software Verification Summary

Model-Based Testing on Host
- 100% conformance testing achieved
- 100% model coverage achieved
- SCADE Suite KCG is developed according to DO-178C/DO-330 TQL-1 objectives

- Complete traceability from model to code
  - C Code Generation
  - Ada Code Generation

Qualified/Certified Code Generation

Global Product Data Interoperability Summit | 2018
Use the System Model as a Twin

Model-Based Systems Engineering

System Safety Analysis

System Simulation & Digital Twins

System Architecture

System/SW Architecture

SW Components (FMI)

Physics Simulation

Model-Based Software Engineering

medini™ analyze

Twin Builder

BOEING is a trademark of Boeing Management Company
Copyright © 2018 Boeing. All rights reserved.

Copyright © 2018 Northrop Grumman Corporation. All rights reserved.

Global Product Data Interoperability Summit | 2018