GLOBAL PRODUCT DATA INTEROPERABILITY

SUMMIT 2018

ELYSIUM Parker Aerospace NORTHROP GRUMMAN BOEING

Better Defined Software Requirements with Model-based Systems Engineering

Functional Safety Analysis and Control Tuning

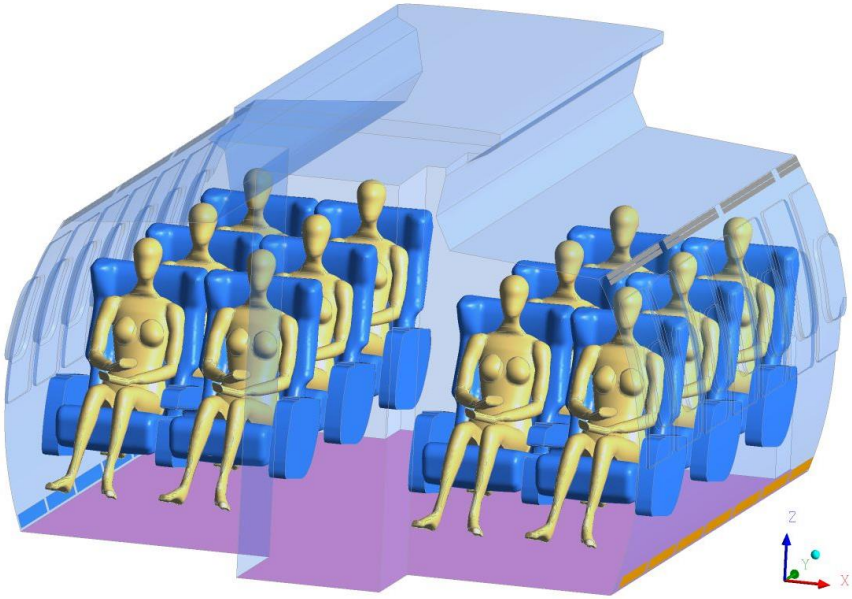
Rachel Knutson – Senior Field Application Engineer

- **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

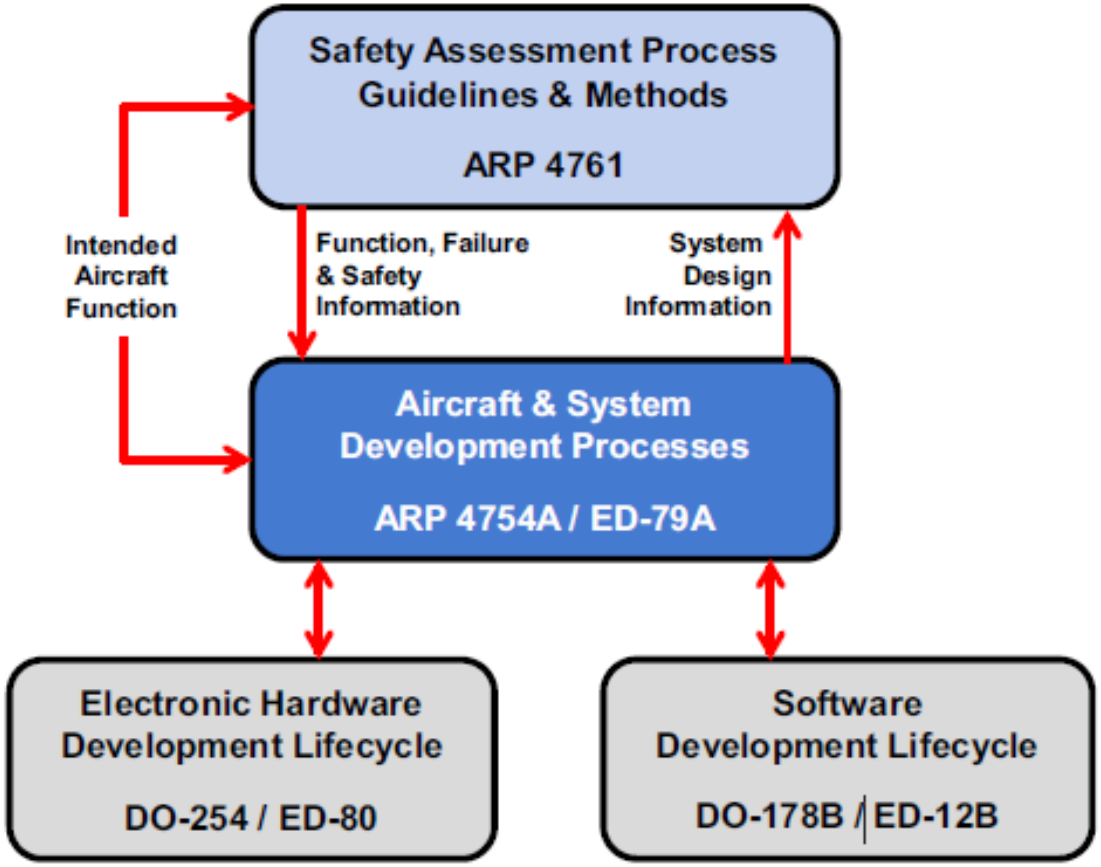
Project Scope

- **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.**

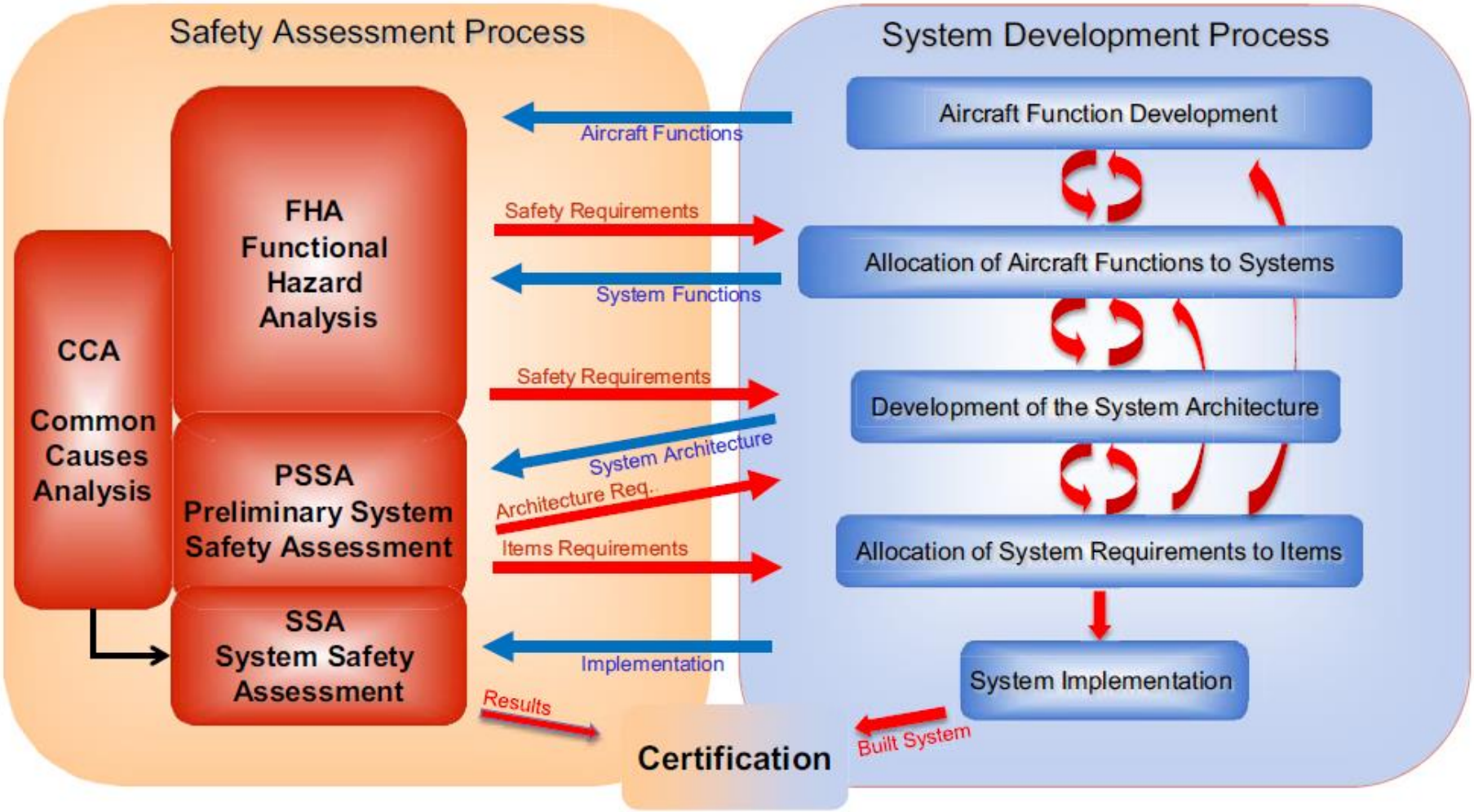


ARP4754A, ARP476,1 DO-254, DO-178C Standards



System Safety and Engineering Processes in ARP4761 and ARP4754A

Global Product Data Interoperability Summit | **2018**

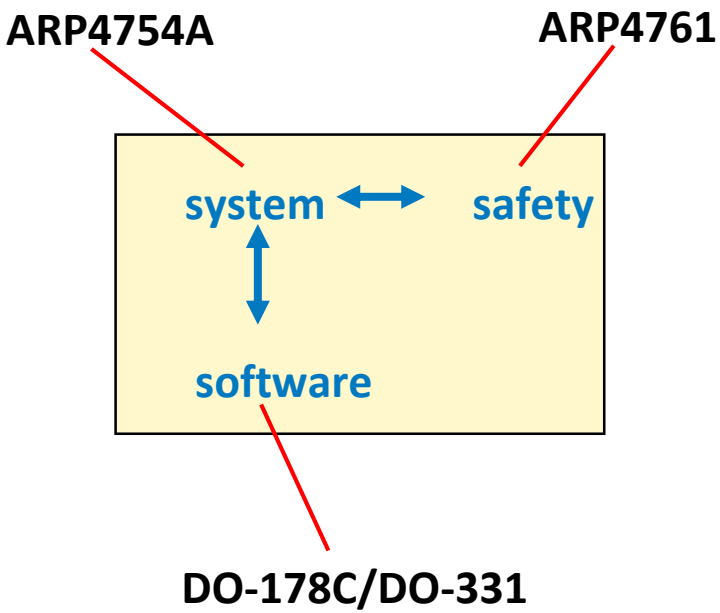


DO-178C/DO-331 Model Usage Examples

Process that generates the life-cycle data	MB Example 1	MB Example 2	MB Example 3	MB Example 4 (See Note 1)	MB Example 5 (See Note 1)
System Requirement and System Design Processes	Requirements allocated to software	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed
					Design Model
Software Requirement and Software Design Processes	Requirements from which the Model is developed	Specification Model (See Note 2)	Specification Model	Design Model	
	Design Model	Design Model	Textual description (See Note 3)		
Software Coding Process	Source Code	Source Code	Source Code	Source Code	Source Code

Objectives of the Approach

- **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**



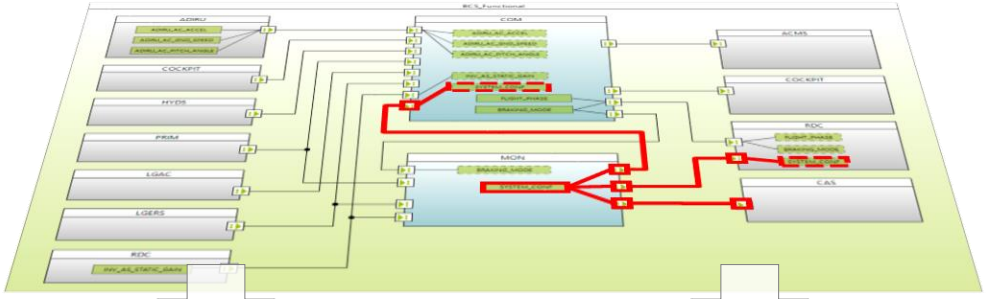
Systems Requirements Definition

- **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

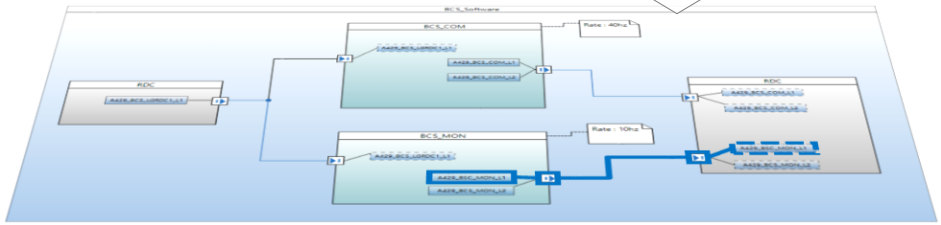
Requirements Examples from DOORS

ID	Requirement	
CPCS-SYS-3	False	**3 System Operating Requirements**
CPCS-SYS-4	False	**3.1 Normal Automatic Mode Requirements**
CPCS-SYS-5	True	Test The Normal Automatic Mode shall be the active mode of the system upon system start-up.
CPCS-SYS-7	True	The Normal Automatic Control Mode shall be controlled by the IASC primary channel (Channel A).
CPCS-SYS-9		**3.2 Manual Mode Requirements**
CPCS-SYS-12	True	When the MANUAL ON command is sent from the Control Panel, the Manual Control Mode shall be active.
CPCS-SYS-13	True	Manual Control Mode shall be controlled by the IASC back up channel (Channel B).
CPCS-SYS-14	True	While the Manual Control Mode is active, the IASC primary channel (Channel A) shall be inoperative for control commands.
CPCS-SYS-15	True	While the Manual Control Mode is active, the IASC primary channel (Channel A) shall be active for data monitoring.
CPCS-SYS-16	True	While in Manual Control Mode, the system shall allow manual control of the cabin altitude rate of change.

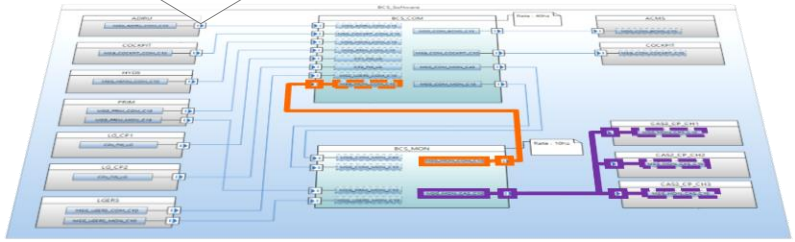
Allocation of Functions to Software and Platform Components



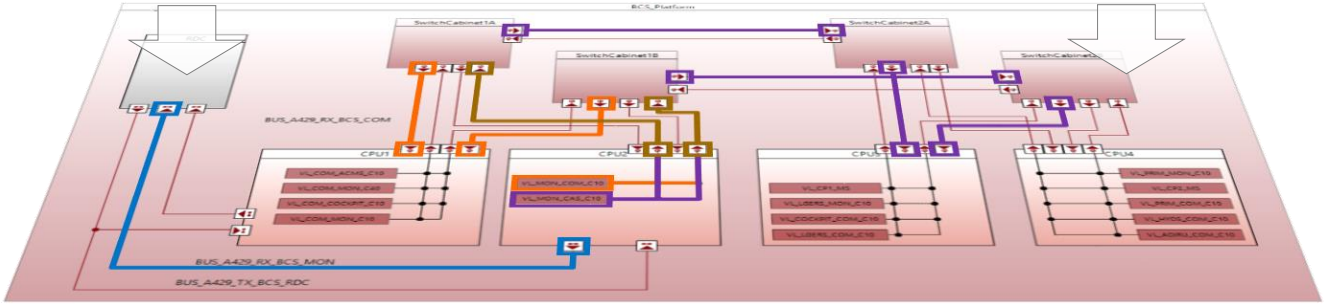
Functional Architecture



Software Architecture (ARINC 429)



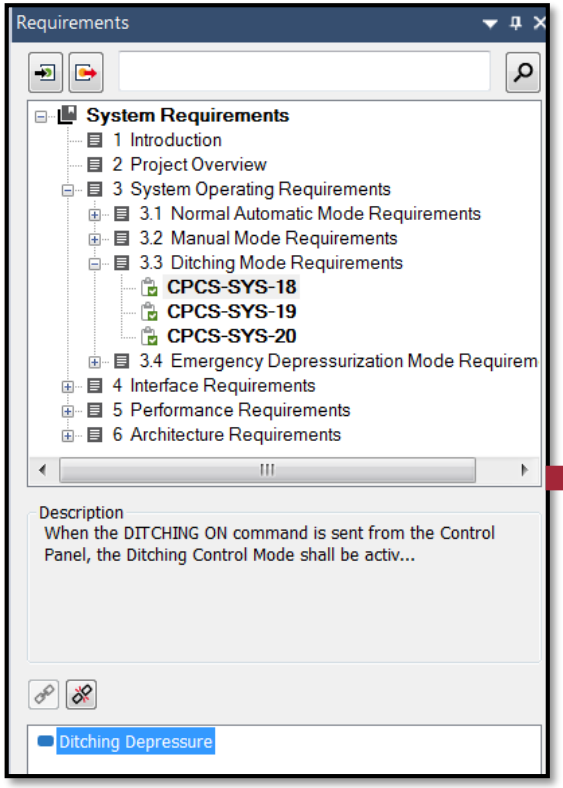
Software Architecture (ARINC 664-P7 / AFDX)

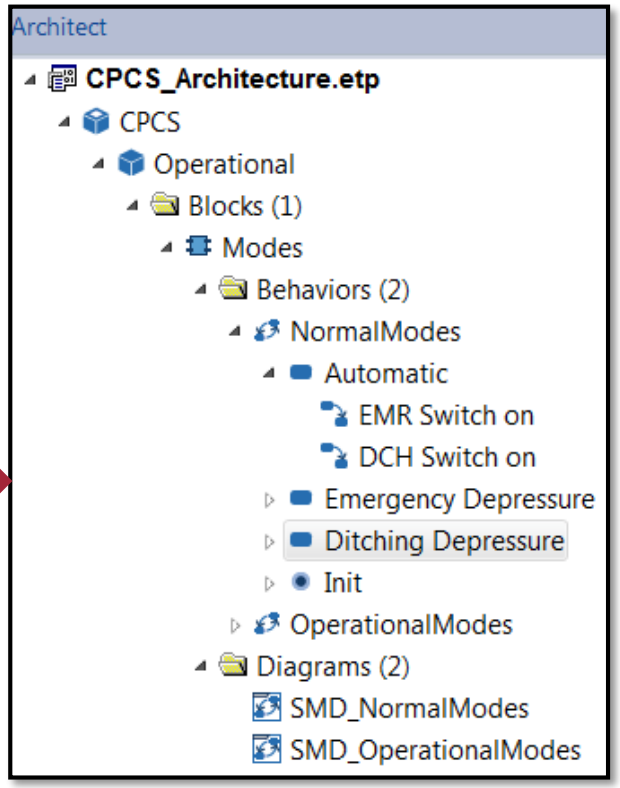


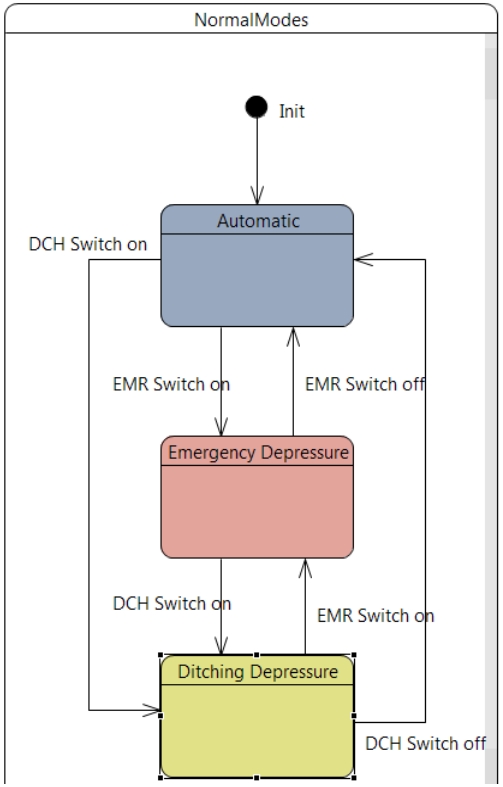
Platform (ARINC 429 and ARINC 664-P7 / AFDX)

Architecture Definition

From system requirements to architecture and operational representation

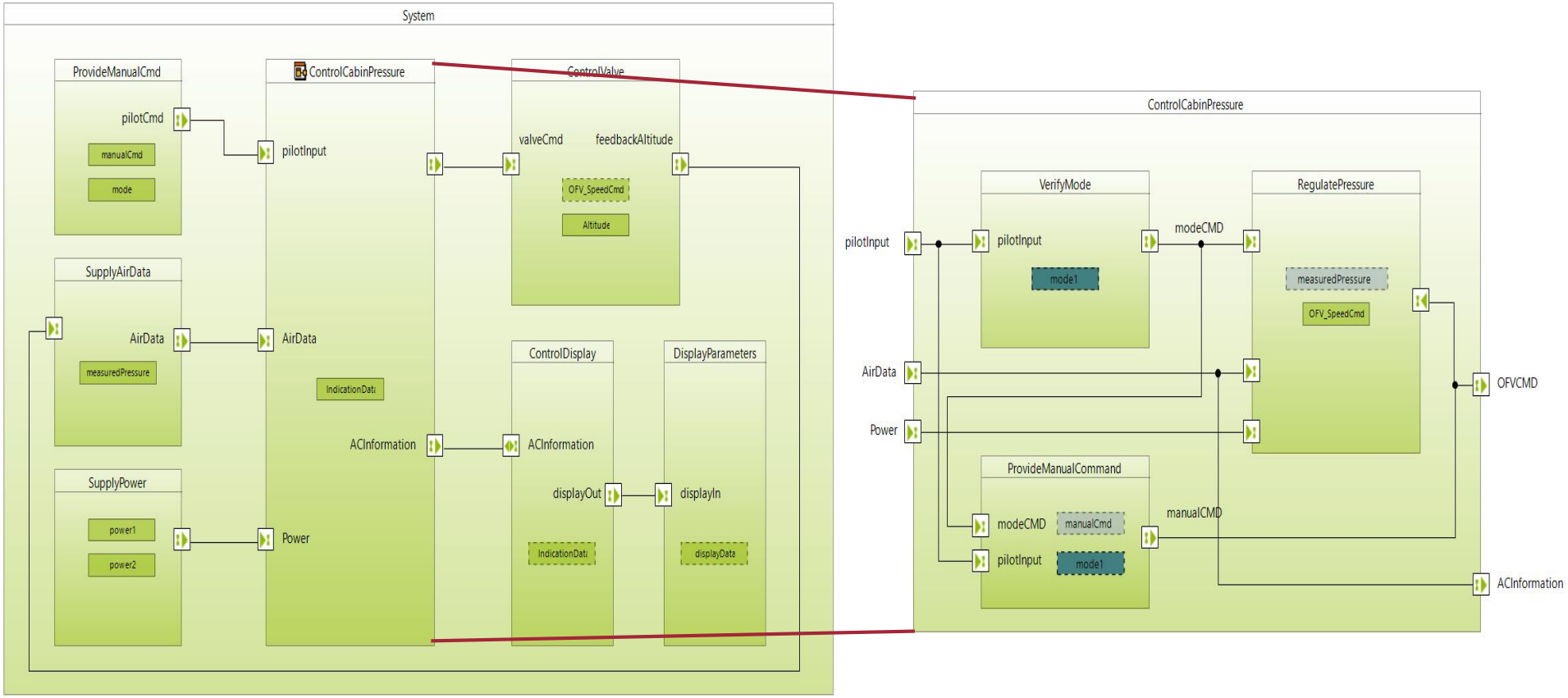






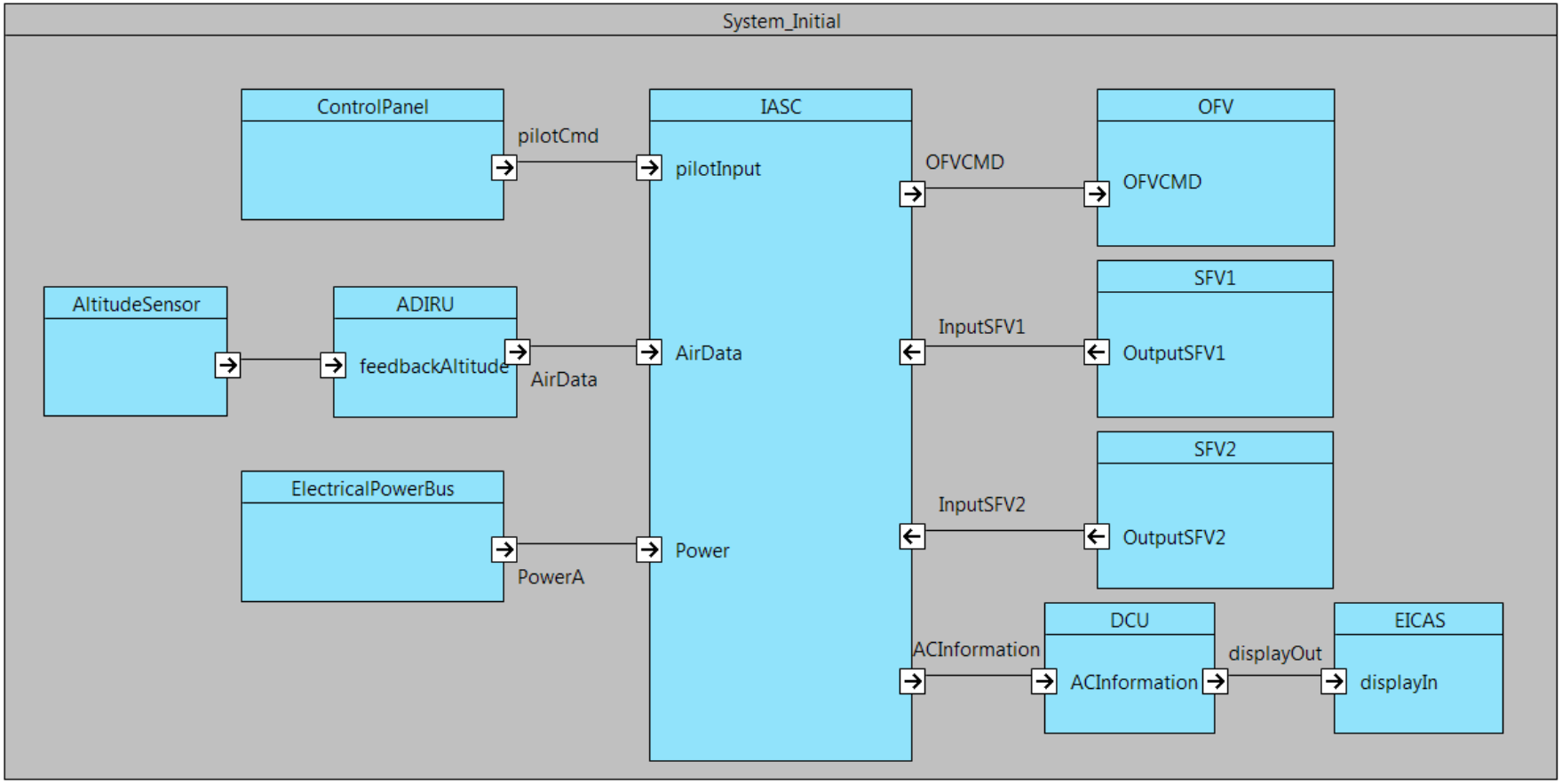
Functional Decomposition of Operational Requirements

Global Product Data Interoperability Summit | **2018**

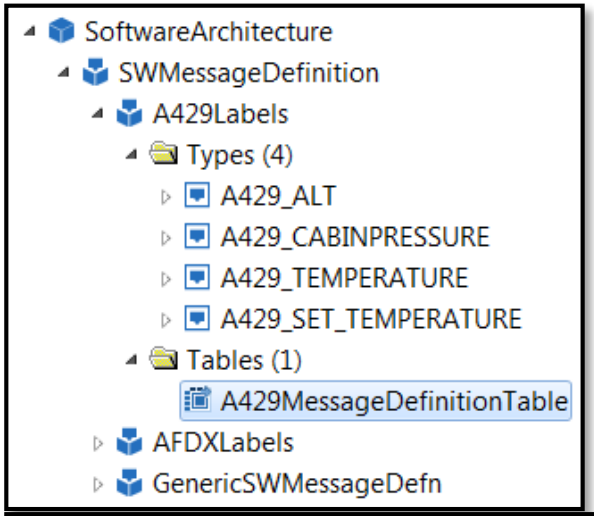


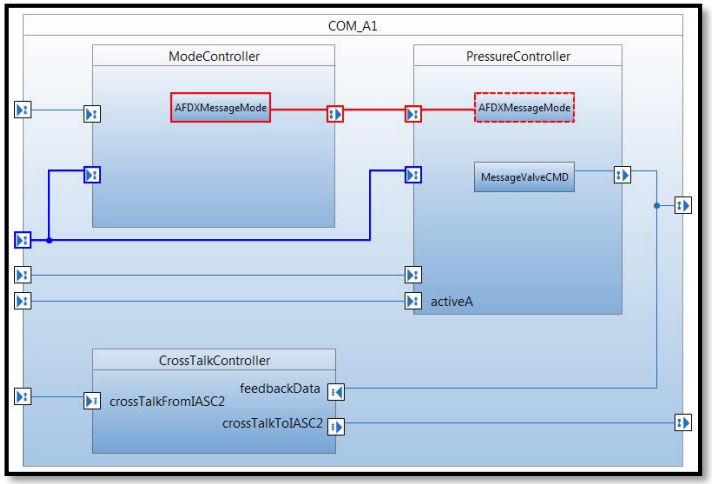
Architecture Definition (Initial)

Global Product Data Interoperability Summit | **2018**



Software Architecture Perspective – ICD Generation

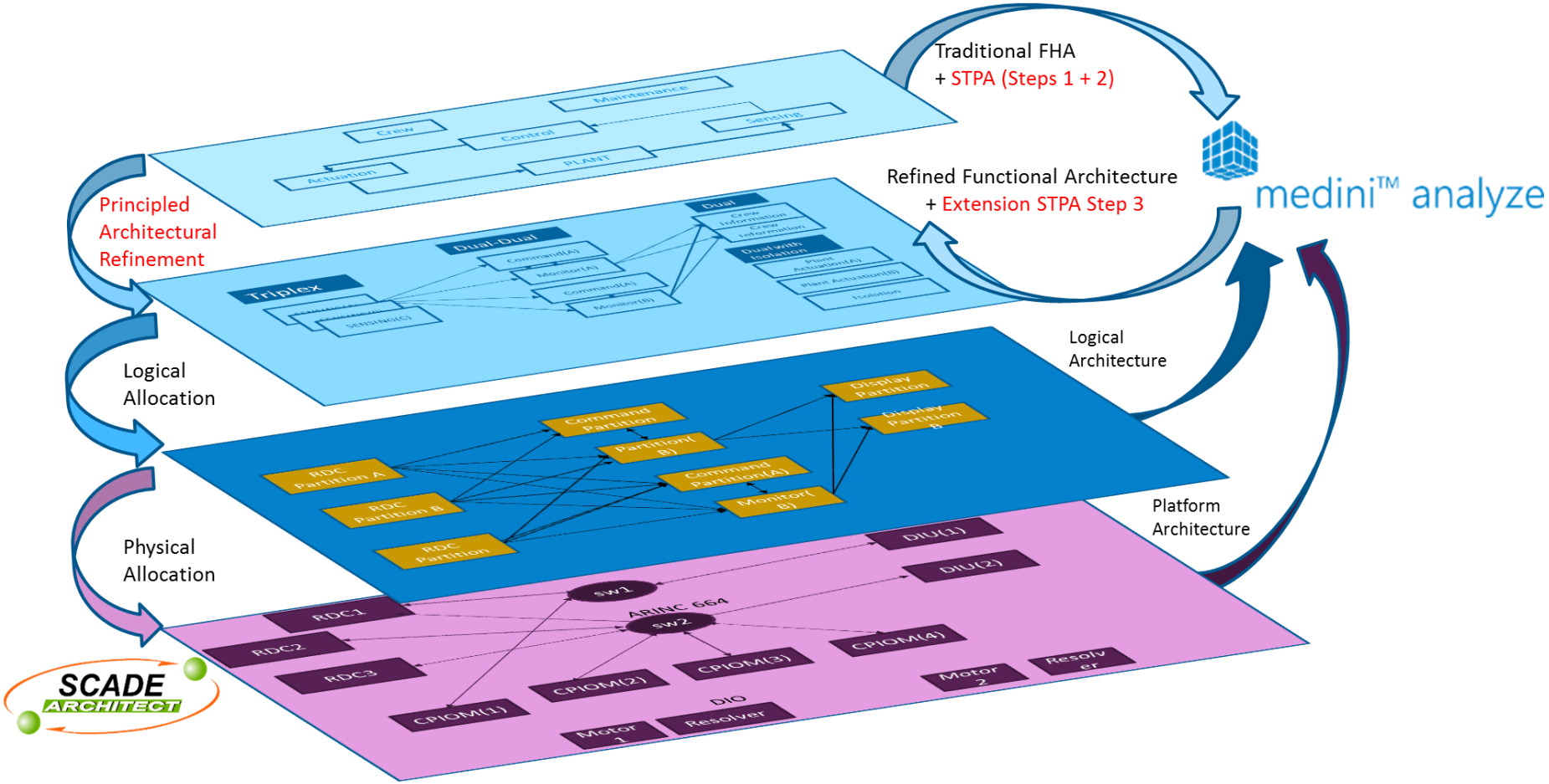




		A	B	C	D	E	F	G	H	I	J
		LabelID	Field Type	Position	Size	Bit29mean0	Bit29mean1	Encoding	EquipmentID	SDIRole	SSMatrixLength
1	A429_ALT	203				positive sign	negative sign	BNR = 1	260	data = 0	2
2	MessageFields										
3	ALTITUDE		real	11	18						
4	A429_CABINPRESSURE	152				positive sign	negative sign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

Architecture and Safety Workflow (Iterations)

Global Product Data Interoperability Summit | **2018**



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**

Checklist

type filter text

Task/Requirement	Checked	Related Artifacts	Checked By	Date of Check
Safety responsibilities	☐			
Aircraft safety group	☐			
Establish and communicate the safety requirements at all tiers of definition	☑		ihoffman	01.03.18 17:07
Identify aircraftlsystem functions required for continued safe flight and landing	☑		ihoffman	01.03.18 17:07
Develop an Aircraft-Level Functional Hazard Assessment (FHA) [if at the OEM level]	☑		ihoffman	01.03.18 17:07
Develop System FHA manual and ensure System FHAs are performed consistently in accordance with this guidelines [if at the OEM level]	☑	System FHA cabin pressure	hoffman	23.02.18 11:54
Conduct thorough, integrated survivability assessments [if at the OEM level]	☐			

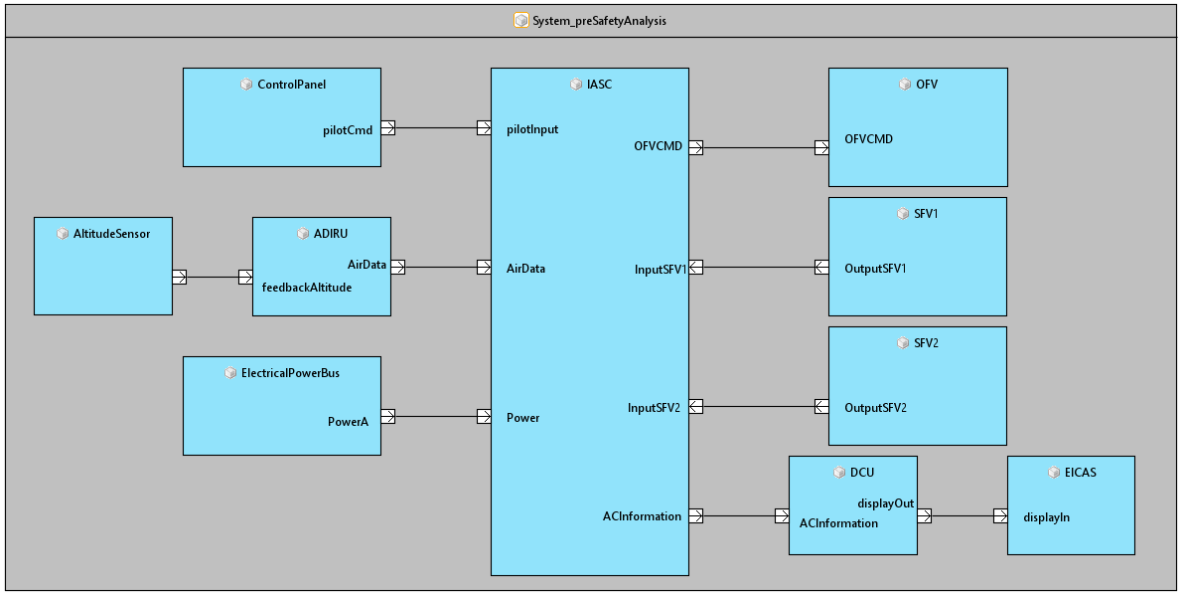
ID	Function	HAZOP Guide Word	Operational /Environmental condition	Phase	Failure Condition (Hazard description)	Effect of failure condition on Aircraft/Crew	Failure Condition Classification	DAL	Safety Objective
IPS-1-02	[F001] Provide cabin pressure and air exchange	NOT	Cruise altitude	In Flight	[IPS-1-02] Unannunciated loss of cabin pressure	Significant reduction of oxygen flow and pressure inside cabin decreases. Significant increase on crew workload.	Hazardous	B	[SO_01] The EICAS has to be available.
IPS-1-04	[F001] Provide cabin pressure and air exchange	NOT	Take off	In Flight	[IPS-1-03] Annunciated loss of cabin pressure	Significant reduction of oxygen flow and pressure inside cabin decreases. Significant increase on crew workload.	Major	C	[SO_01] The EICAS has to be available.

Safety Analysis

Global Product Data Interoperability Summit | **2018**

- 21 AIR CONDITIONING AND PRESSURIZATION
  - Aircraft Level
    - 21Air conditioning and pressurization
  - Functional architecture design
  - Safety program plan [7/74 checked]
- Functional hazard assessment
  - System FHA
    - SFHA
      - System FHA cabin pressure
        - [IPS-1-02] Unannunciated loss of cabin pressure
        - [IPS-1-04] Annunciated loss of cabin pressure
      - System Failure condition at system level
- Preliminary safety assessment
  - PSSA cabin pressure
    - Cabin pressure
      - Cabin pressure control system
      - [E1] Unannunciated loss of cabin pressure
        - [I1] EICAS failure
          - Logical Gate [1 trace]
            - [E1] IASC2 failure
            - [E2] IASC1 failure
              - Logical Gate
- System safety assessment
  - System Architectur
    - CPCS
      - Functional
      - Software
        - IBD_IASC_1
        - IBD_IASC_2
        - SWMessageDefinition
        - IASC2
        - IASC1
      - Hardware
      - Package
    - Library
  - FMEA
  - FMES
- CCA
- Specifications
  - Safety objectives
  - Safety requirements from the PSSA

- **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**

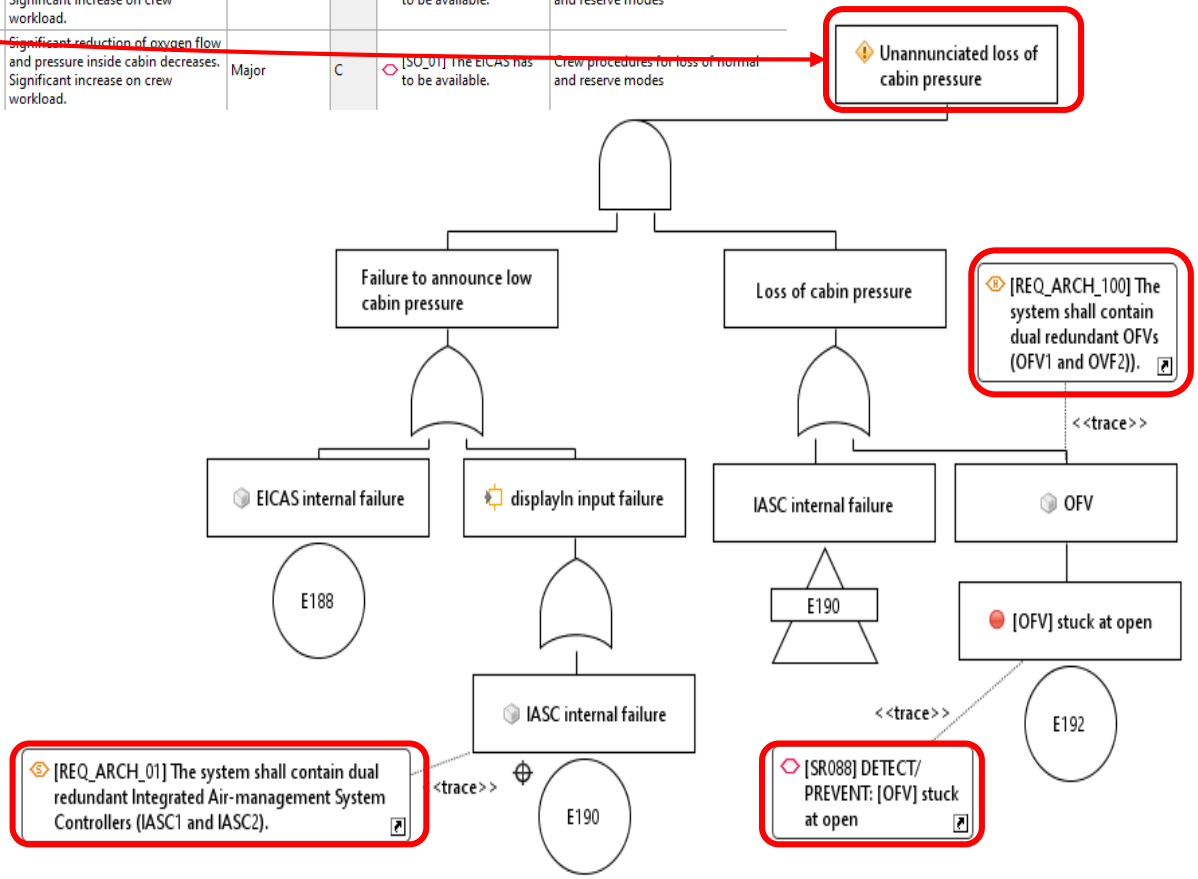


Functional Safety Analysis

Global Product Data Interoperability Summit | **2018**

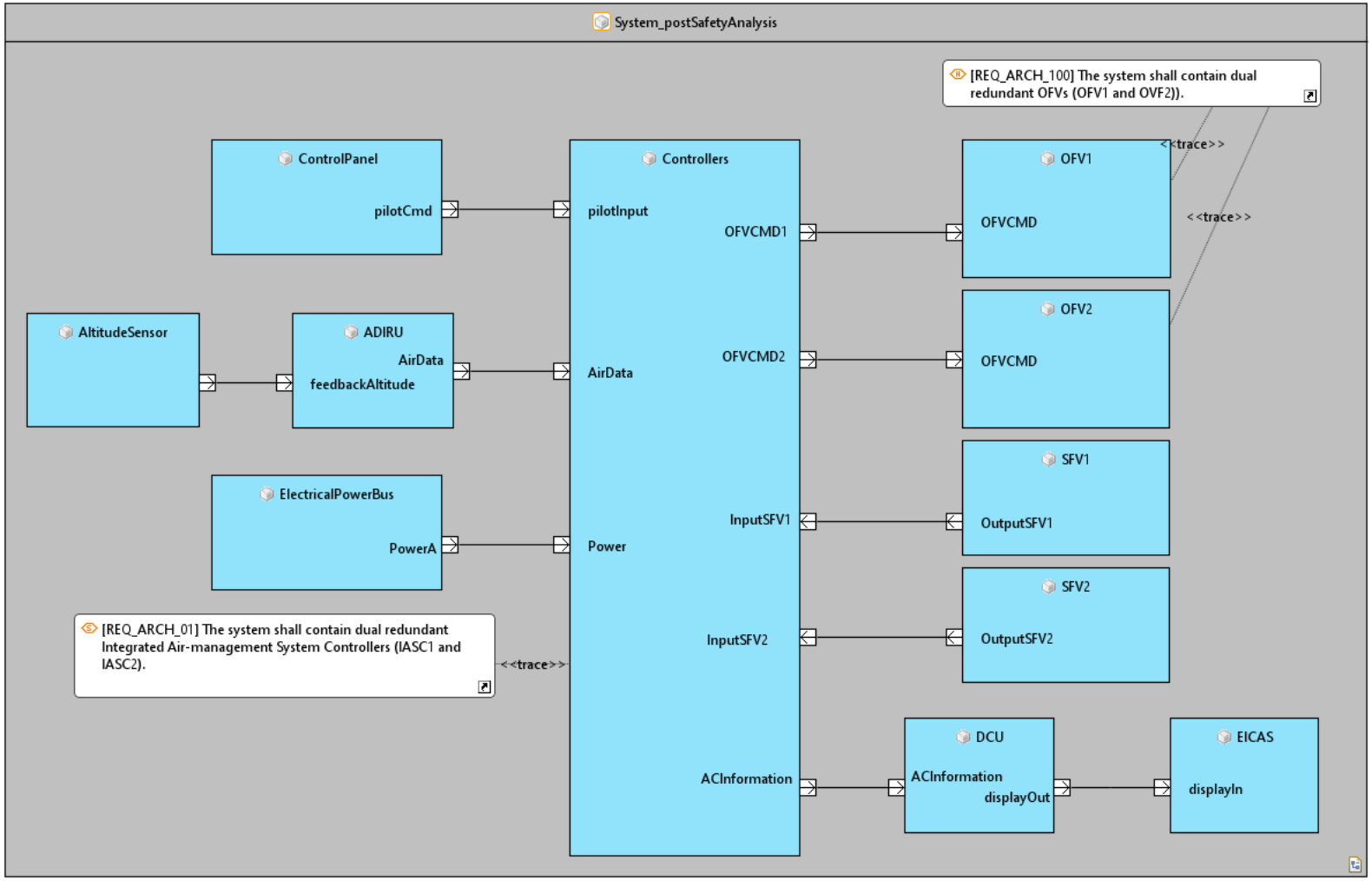
ID	Function	HAZOP Guide Word	Operational/Environmental condition	Phase	Failure Condition (Hazard description)	Effect of failure condition on Aircraft/Crew	Failure Condition Classification	DAL	Safety Objective	Reference to supporting Material
IPS-1-02	[F001] Provide cabin pressure and air exchange	NOT	Cruise altitude	In Flight	[IPS-1-02] Unannunciated loss of cabin pressure	Significant reduction of oxygen flow and pressure inside cabin decreases. Significant increase on crew workload.	Hazardous	B	[SO_01] The EICAS has to be available.	Crew procedures for loss of normal and reserve modes
IPS-1-04	[F001] Provide cabin pressure and air exchange	NOT	Take off	In Flight	[IPS-1-03] Annunciated loss of cabin pressure	Significant reduction of oxygen flow and pressure inside cabin decreases. Significant increase on crew workload.	Major	C	[SO_01] The EICAS has to be available.	Crew procedures for loss of normal and reserve modes

- **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

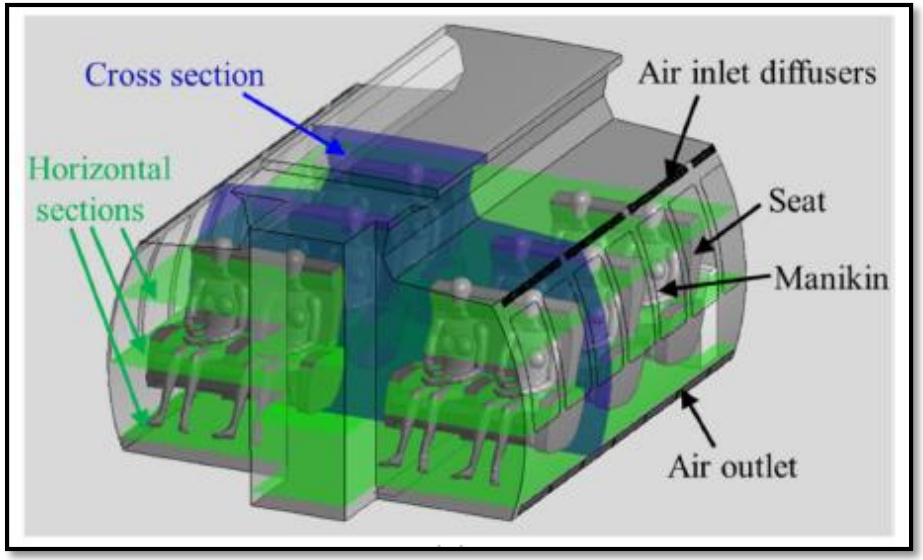
Global Product Data Interoperability Summit | **2018**

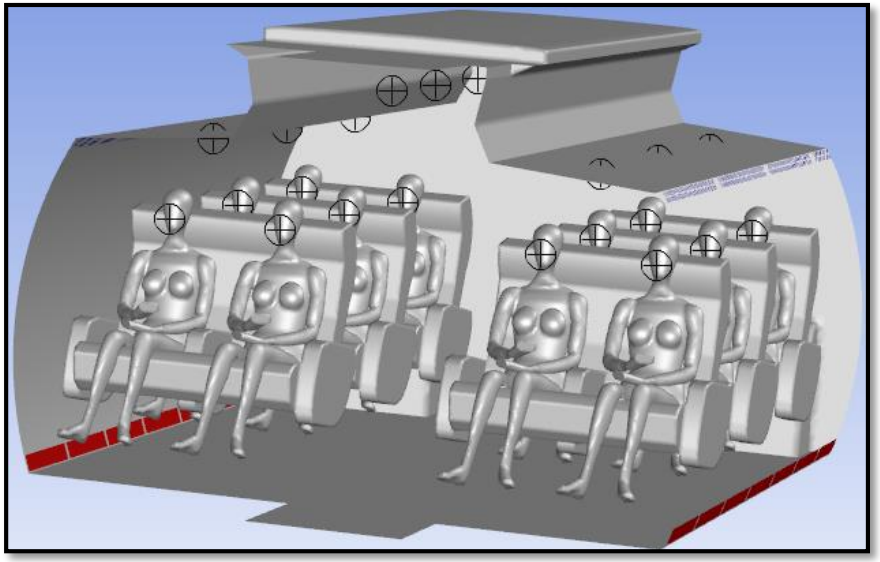


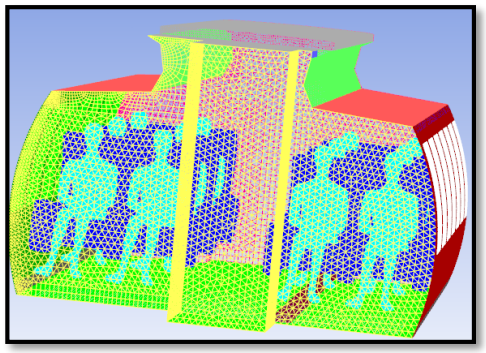
Control Laws Design

- **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



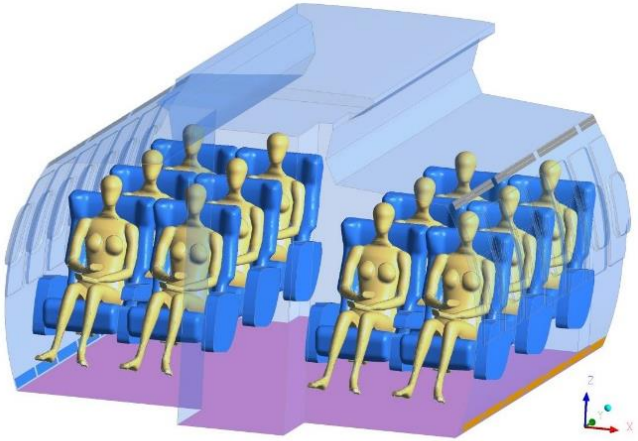


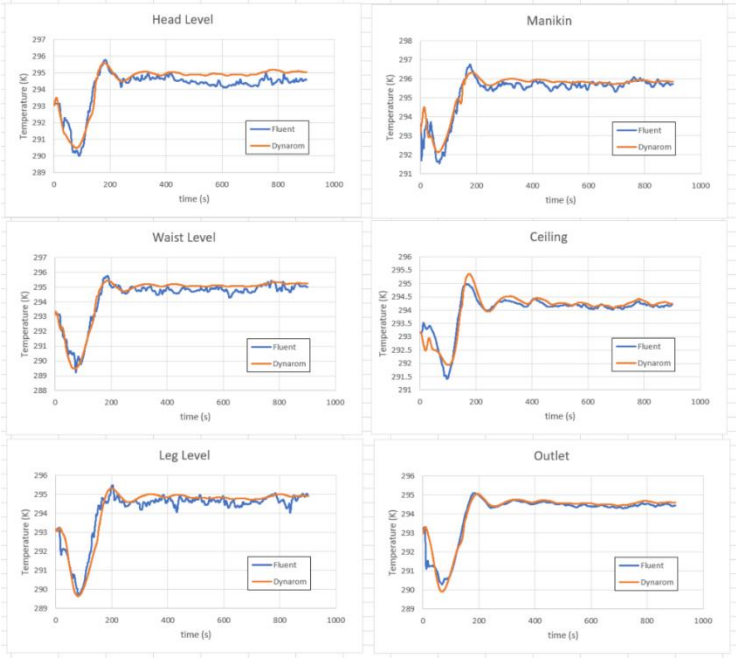


- **ANSYS Fluent captures the 3D air flow within a MD-82 cabin**
- **Virtual sensors distributed spatially in the cabin are used to monitor temperature and pressure**
- **CFD analysis is generated for the entire cabin**
  - **computation time = several days**

Generate Reduced Order Model

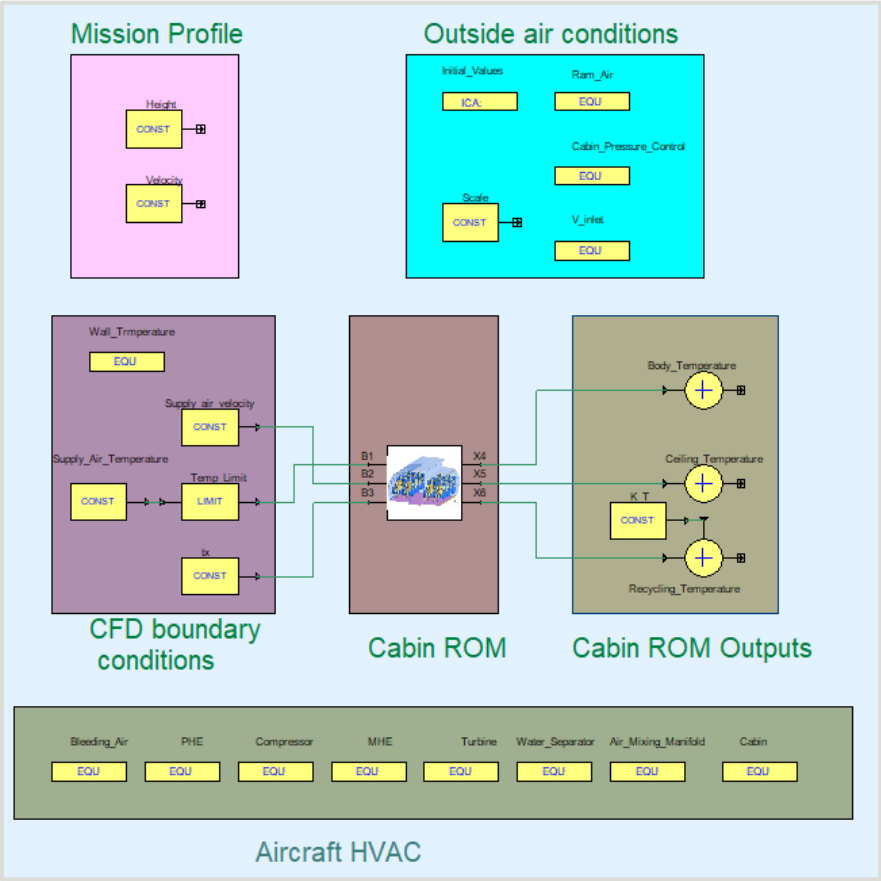
- **3D models are extremely precise but require large CPU time**
- **ANSYS unique technology (deep learning) enables the creation of a Reduced Order Model for transient non-linear 3D models**
- **Resulting Performance improvement:**
  - **Fluent - 40 hours on 48 CPU**
  - **ROM - 4 seconds on 1 CPU**





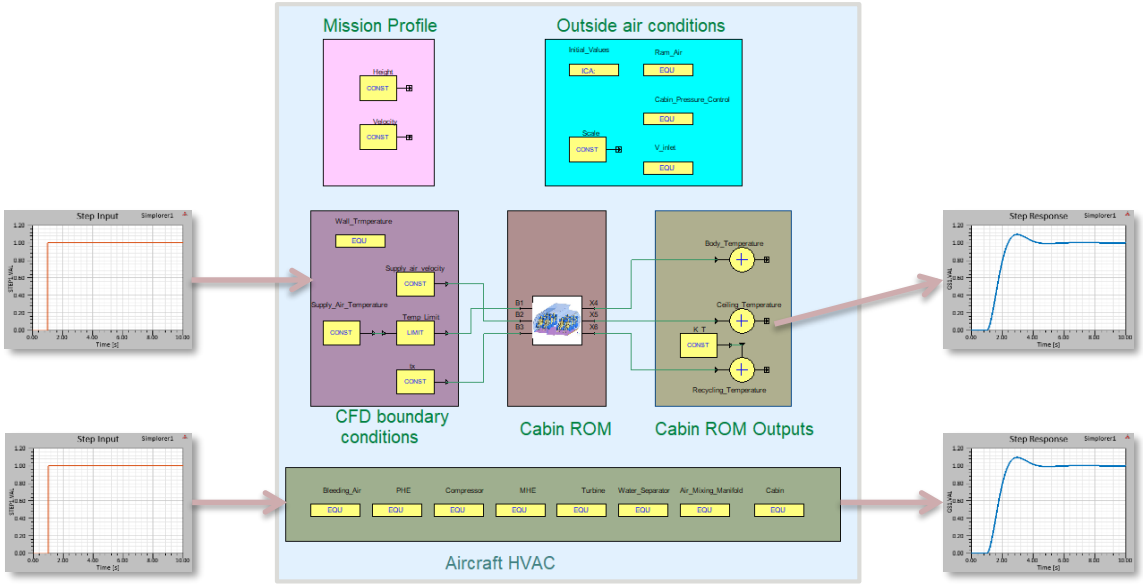
Construct a High-Fidelity System Model with Twin Builder

- **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**



Extract State Space Equations to Design the Controller

- **Twin Builder can automatically extract a continuous linear State-Space model from any arbitrarily complex system model**



Step Inputs

System Step Responses

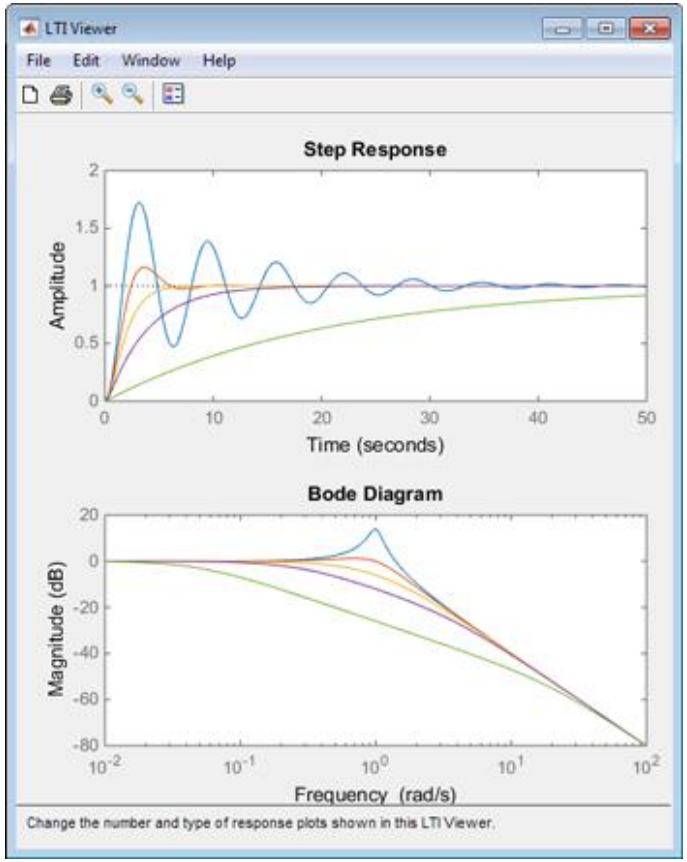
Generate LTI matrices:

$$\dot{x}_{(t)} = A * x_{(t)} + B * u_{(t)}$$

$$y_{(t)} = C * x_{(t)} + D * u_{(t)}$$

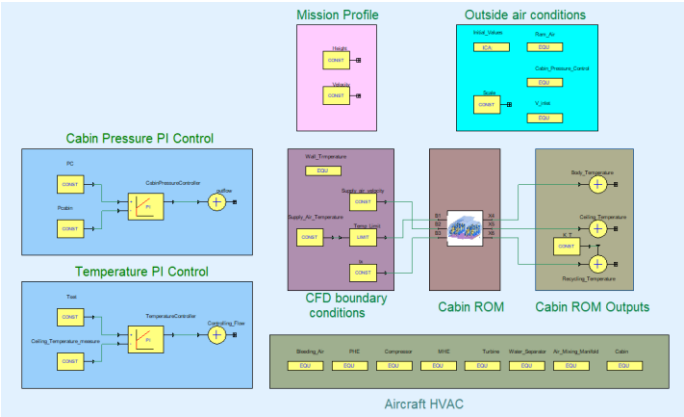
Perform Initial Controller Tuning

- **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**

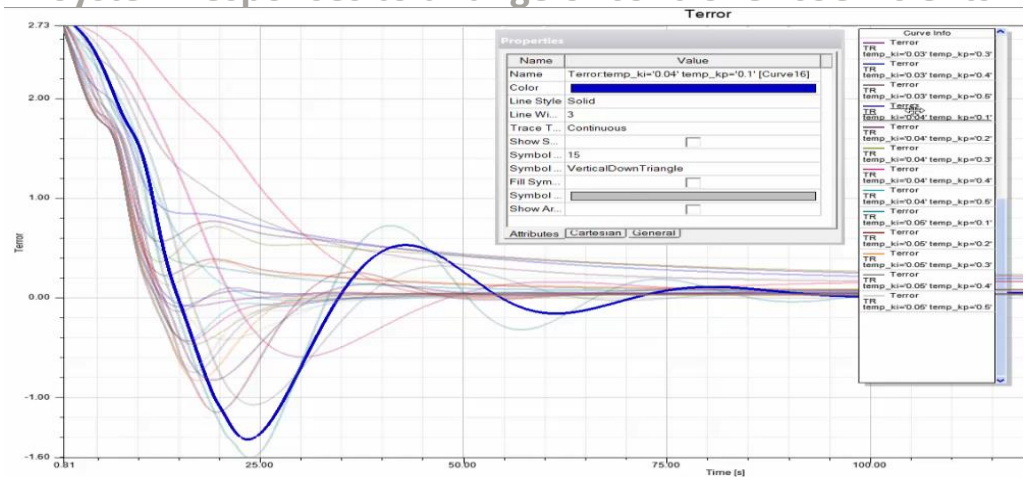


Validate Controller Requirements within the System Model

- **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**

5 Performance Requirements

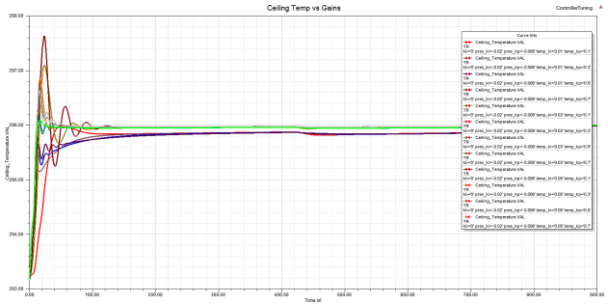
While in Normal_Operational_Mode, the cabin pressure system shall regulate the cabin pressure to maintain a cabin_pressure_level within 0.2% of the cabin_pressure_target_set point.

While in Normal_Operational_mode, when a 100Hpa change of the cabin_pressure_target_set_point is requested, the cabin pressure system shall regulate the cabin pressure, to achieve a cabin_pressure_level within 0.2% of cabin_pressure_target_set_point within 20 seconds.

While in Normal_Operational_Mode, the cabin pressure shall monitor the cabin_pressure_command_panel, and set the cabin_pressure_target_set_point to reflect the hardware_seleced value with 0.1 seconds.

While in Normal_Operational_Mode, the cabin_environmental_control_system shall regulate the cabin temperate to maintain a cabin_temperature_level within 0.3° of the cabin_temperature_target_set point.

While in Normal_Operational_mode, when a 3°C change of the cabin_temperature_target_set_point is requested, the cabin pressure system shall regulate the cabin temperature, to achieve a cabin_temperature_ level within 0.3° of cabin_temperatire _target_set_point within 20 seconds.



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*ft)^-1

Ki = -0.02 (s^2 * ft)^-1

Kd = 0.0 (ft)^-1

dT = 100 ms

loLimit = OFV_NORMAL_SPEED_CMD deg/s

Definition of Software High Level Requirements (HLRs)

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

- **Example Software HLRs:**

CPCS-SW-3	**1.2 Pressure Controller**
CPCS-SW-9	The Pressure Controller shall use the following constants when determining the targeted cabin altitude: ALT1: 20,000 ft CAB_ALT1: 4,000 ft OFV_DITCHING_SPEED_CMD: -20 deg/s OVF_EMERGENCY_SPEED_CMD: 25 deg/s OFV_NORMAL_SPEED_CMD: 10 deg/s
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*ft)^-1 Ki = -0.02 (s^2 * ft)^-1 Kd = 0.0 (ft)^-1 dT = 100 ms loLimit = 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*ft)^-1 Ki = -0.02 (s^2 * ft)^-1 Kd = 0.0 (ft)^-1 dT = 100 ms loLimit = OFV_NORMAL_SPEED_CMD deg/s highLimit = - OFV_NORMAL_SPEED_CMD deg/s

Implementation of HLR

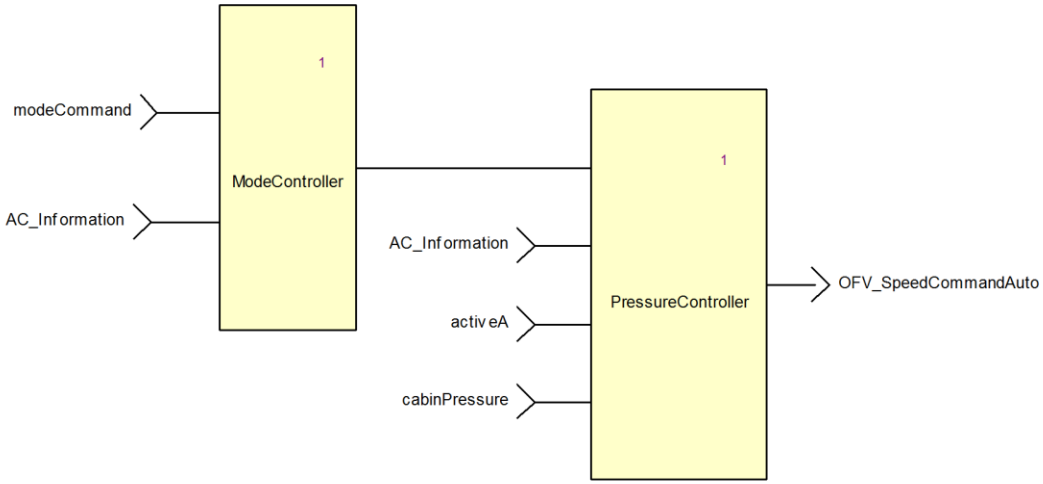
Our Example requirement has two parts:

- **Part 1: Mode constraint**

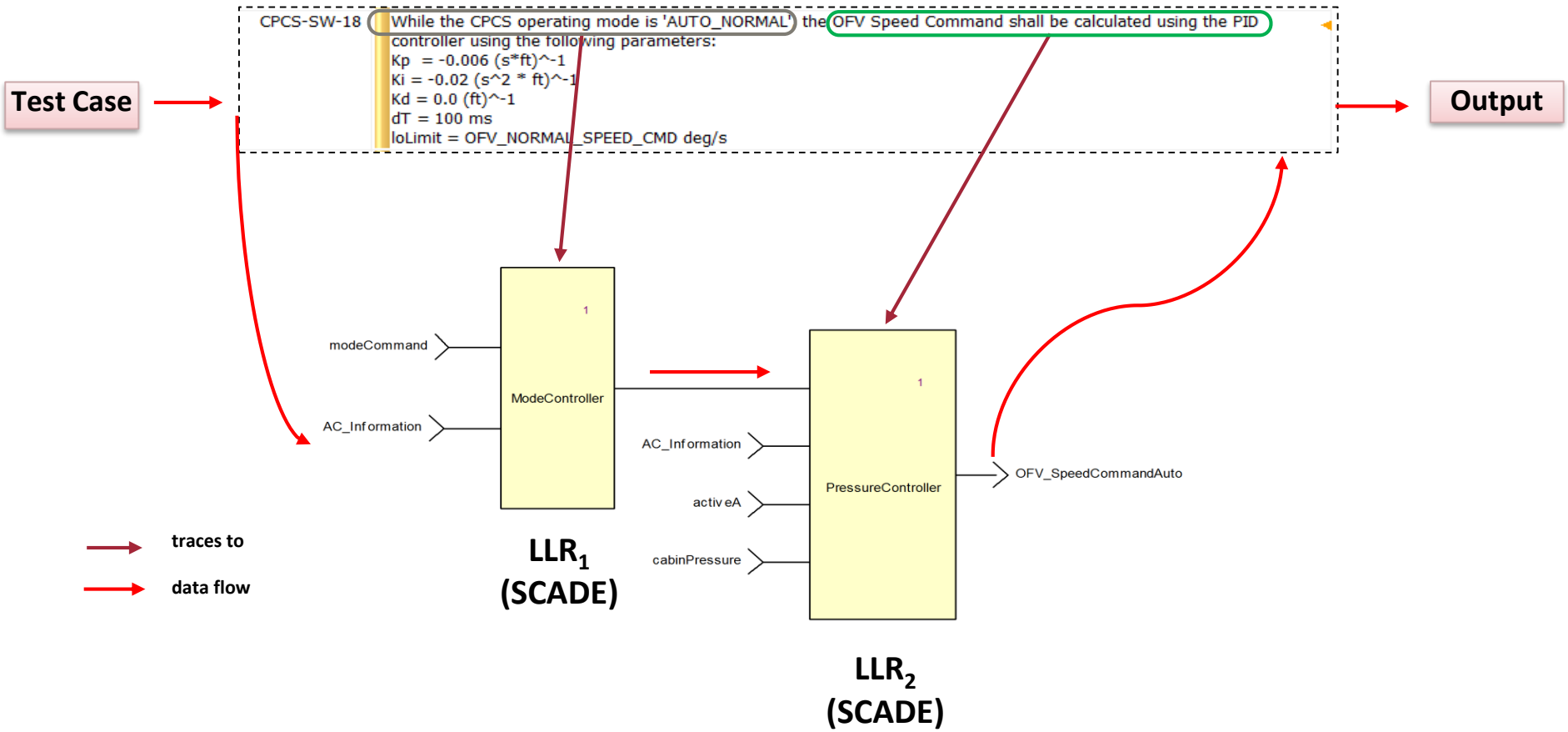
  " While the CPCS operating mode is 'AUTO_NORMAL'…"

- **Part 2: Directive for the output**

  "…the OFV Speed Command shall be calculated using the PID controller…"

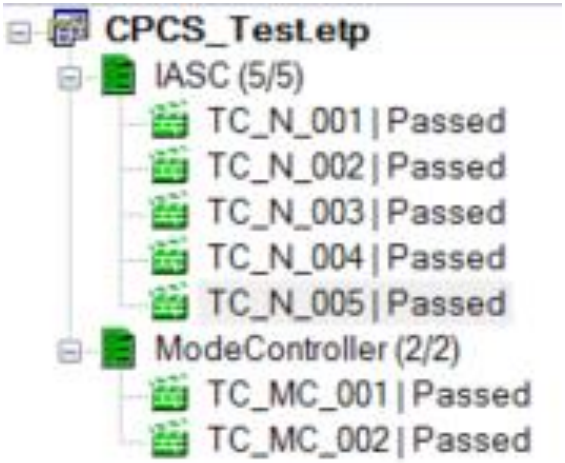


Software Verification: Example HLR



Software Verification Summary

- **Model-Based Testing on Host**
  - **100% conformance testing achieved**
  - **100% model coverage achieved**



2. Coverage Information

Model Test Coverage

2.1. Overview

Coverage percentage of the project: **100.00 %**

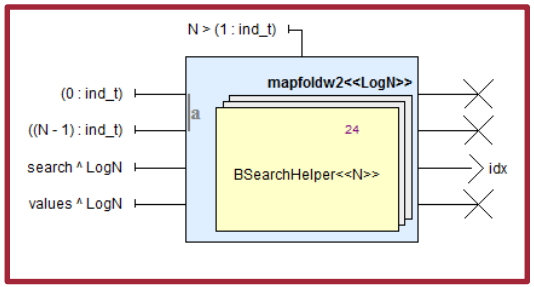
2.2. Covered Operators

Entity Path	Coverage
IASC::IASC/	8/8
IASC::ModeController/	8/8
IASC::PC_Cabin_GetTargetCabAlt/	4/4
IASC::PC_OFV_Automatic/	4/4
IASC::PC_OFV_SpeedCommand/	17/17
IASC::PressureController/	12/12

Qualified/Certified Code Generation

- **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**




```
#ifndef _BSearch_P_int8_10_4_H_
#define _BSearch_P_int8_10_4_H_

#include "kcg_types.h"

/* P::BSearch/ */
extern void BSearch_P_int8_10_4(
  /* search/ */
  kcg_int8 search_int8_10_4,
  /* values/ */
  array_int8_10 *values_int8_10_4,
  /* IfBlock1:, _L1/, out_of_range/ *
  kcg_bool *out_of_range_int8_10_4,
  /* idx/ */
  ind_t_P *idx_int8_10_4);


#endif /* _BSearch_P_int8_10_4_H_ */
```

Standard ANSI C code

```
with Kcg_Types;
with Kcg_Config;

--# inherit Kcg_Types, Kcg_Config;
package P
-- P::
is

  -- P::ind_t/
  subtype ind_t is Kcg_Config.Kcg_Int32;

  -- P::BSearch/
  procedure BSearch_int8_10_4(
    -- search/
    search_int8_10_4 : in Kcg_Config.Kcg_Int8;
    -- values/
    values_int8_10_4 : in Kcg_Types.Int8_Range_0_9;
    -- IfBlock1:, _L1/, out_of_range/
    out_of_range_int8_10_4 : out Boolean;
    -- idx/
    idx_int8_10_4 : out ind_t);


end P;
```

SPARK 95 Ada code

Use the System Model as a Twin

Global Product Data Interoperability Summit | **2018**