30th Annual INCOSE
international symposium

Virtual Event
July 20 - 22, 2020

# Towards a Common Systems Engineering Method to Cover a Complete System Development Process

# Authors

**Aiste ALEKSANDRAVICIENE**

Cyber Systems Industry Business Consultant (EMEAR) at Dassault Systèmes

- BS and MS in Information Systems Engineering
- 16 years in Software and Systems Engineering
- Member of INCOSE
- Author of MagicGrid BoK and a member of MagicGrid development team

**Aurelijus MORKEVIČIUS, PhD**

Cyber Systems, EMEAR Industry Process Expert Senior Manager (EMEAR) at Dassault Systèmes

- PhD, MS, and BS in Software Systems Engineering
- 15 years experience in Software and Systems Engineering
- UAF co-chairman in OMG, member of INCOSE and NATO ACaT
- Author of MagicGrid Method

**Andrius ARMONAS, PhD**

CATIA Systems Modeling Application Director, MagicDraw Product Manager at Dassault Systèmes

- PhD, MS, and BS in Software Engineering
- 20 years in Software Engineering
- Member of INCOSE, received Best Paper awards at INCOSE 2018 and 2019
- Member of OMG Safety and Reliability working group

**Gauthier FANMUY**

Systems Engineering Role Portfolio Director at Dassault Systemes

- Engineer's degree, in Physics, Computer Science, Optics
- 30 years in Systems Engineering
- Member of INCOSE and AFIS (French chapter of INCOSE)

# Outline

- Motivation
- Introduction to MBSE Grid
- Proposed Modifications and Additions
  - Core modifications
  - Addition of Safety & Reliability pillar
  - Bridging the gap between MBSE and MBD
- Conclusions

# Motivation

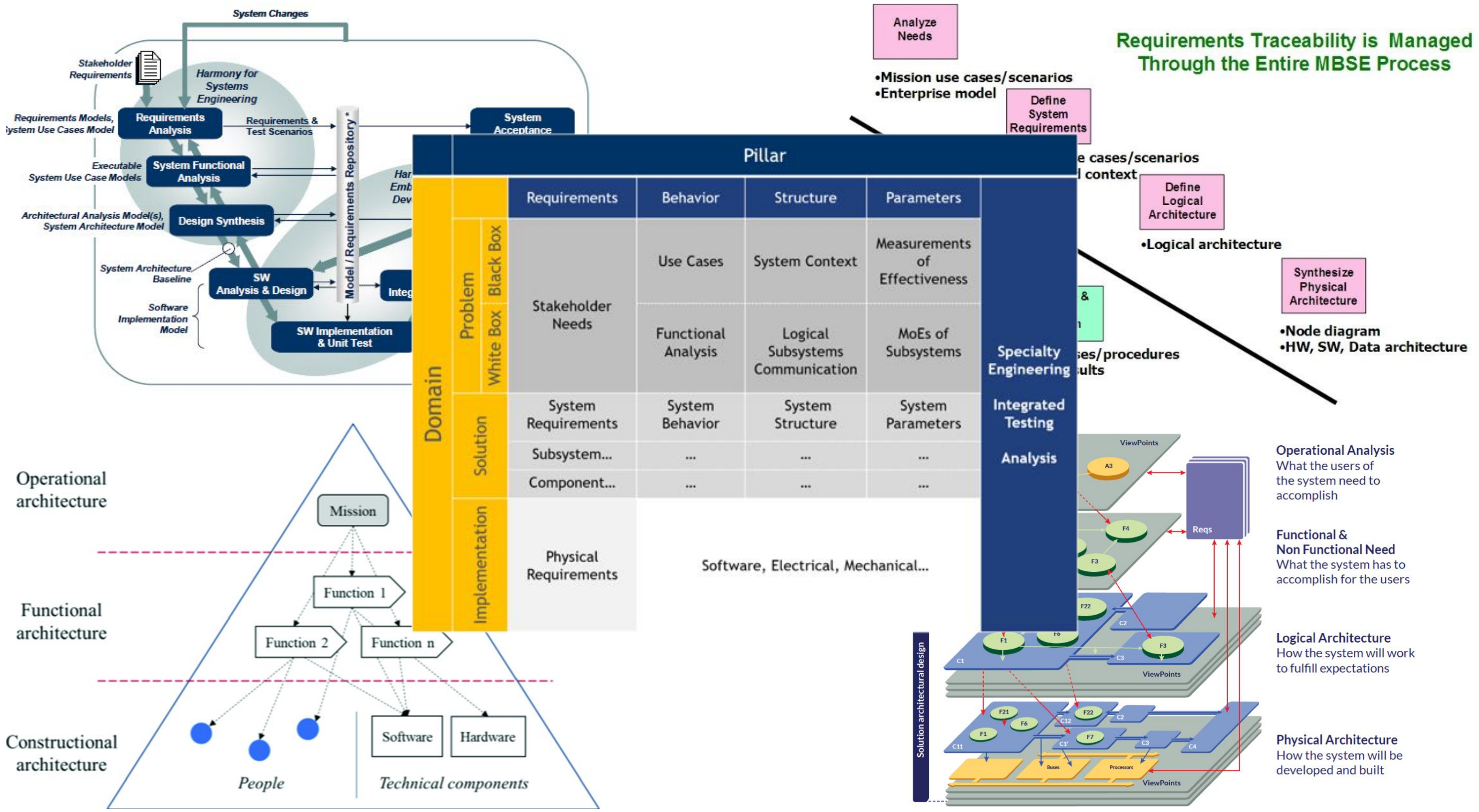
- SysML as a standardized modeling language is a key MBSE enabler for modeling large and complex systems
- SysML is neither a framework nor a methodology
  - SysML does not provide any guidelines or recommendations for the modeling process
- MBSE methodology is needed to address the following questions
  - how to begin modeling the system,
  - what views need to be built,
  - which artefacts will be delivered,
  - what are the engineering activities, etc.
- MBSE Grid modeling method has been introduced In 2017

# Purpose

Revisit and Update MBSE Grid methodology in accordance to:

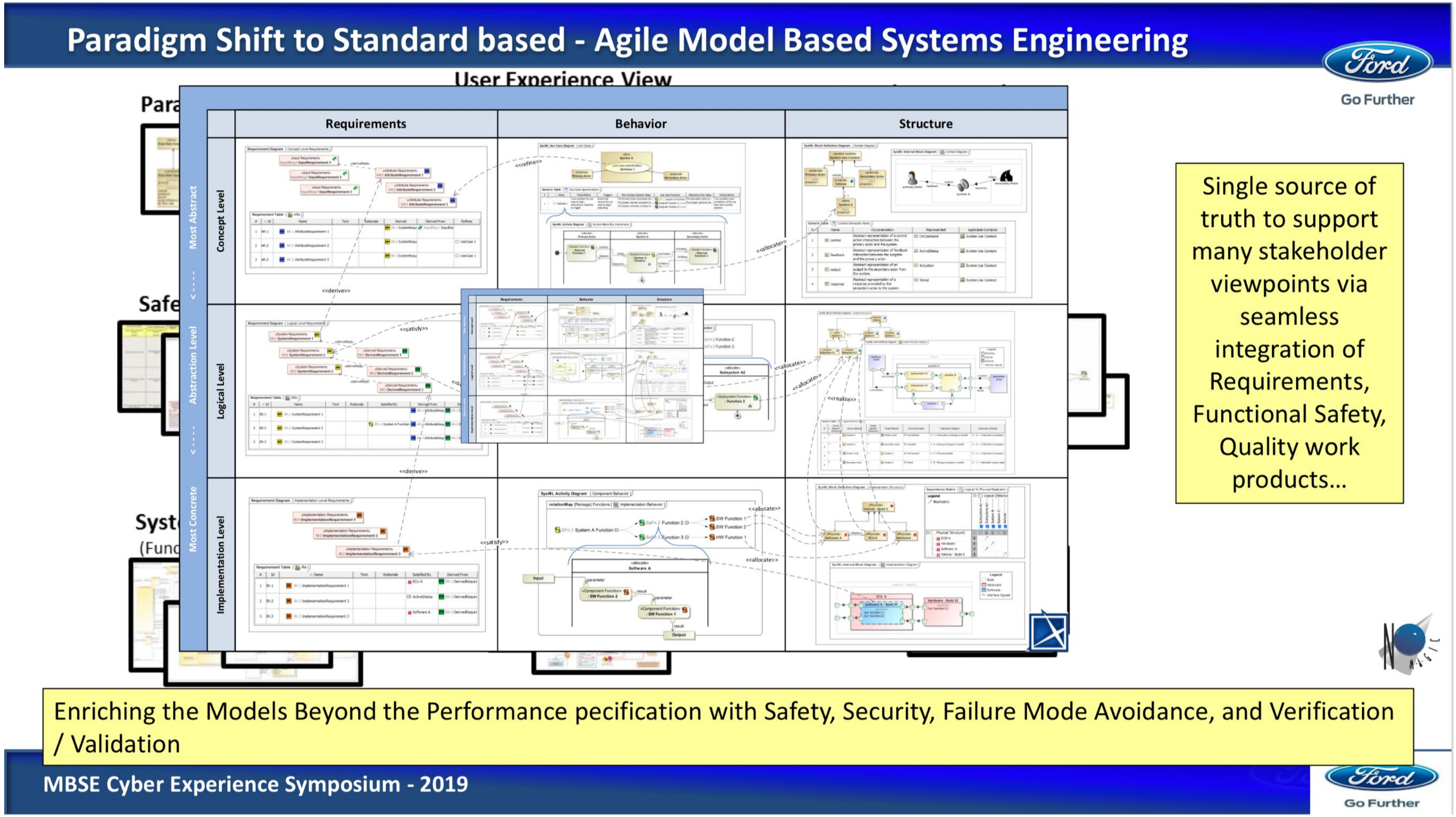
- **Industry Evolution**
  - Following the publication of an MBSE grid, there is a need to revisit existing systems engineering methodologies,
    - to assess MBSE grid applicability to current MBSE projects
    - discover major industry trends
    - evolve the grid in accordance with the engineering community needs.
- **User feedback**
  - Implementing organizations had many questions and requests for improvements. This feedback has been analyzed and MBSE Grid has been updated accordingly.

# Industry Evolution


System Changes
Stakeholder Requirements
Harmony for Systems Engineering
Requirements Models, System Use Cases Model
Requirements Analysis
Requirements & Test Scenarios
Model / Requirements Repository*
System Acceptance
Executable System Use Case Models
System Functional Analysis
Architectural Analysis Model(s), System Architecture Model
Design Synthesis
Har
Emb
Dev
System Architecture Baseline
SW Analysis & Design
Integ
Software Implementation Model
SW Implementation & Unit Test
Operational architecture
Mission
Functional architecture
Function 1
Function 2
Function n
Constructional architecture
Software
Hardware
People
Technical components
Analyze Needs
•Mission use cases/scenarios
•Enterprise model
Define System Requirements
e cases/scenarios
l context
Define Logical Architecture
•Logical architecture
&
n
es/procedures
ults
Synthesize Physical Architecture
•Node diagram
•HW, SW, Data architecture
Requirements Traceability is Managed Through the Entire MBSE Process
ViewPoints
A3
F4
F3
F22
F1
F6
C1
C2
C3
Reqs
F21
C11
C12
C1'
F7
C4
Buses
Processors
Solution architectural design
Operational Analysis
What the users of the system need to accomplish
Functional & Non Functional Need
What the system has to accomplish for the users
Logical Architecture
How the system will work to fulfill expectations
Physical Architecture
How the system will be developed and built

| | | | Pillar | | | | |
|---|---|---|---|---|---|---|---|
| | | | Requirements | Behavior | Structure | Parameters | |
| Domain | Problem | Black Box | Stakeholder Needs | Use Cases | System Context | Measurements of Effectiveness | Specialty Engineering<br>Integrated Testing<br>Analysis |
| | | White Box | | Functional Analysis | Logical Subsystems Communication | MoEs of Subsystems | |
| | Solution | | System Requirements | System Behavior | System Structure | System Parameters | |
| | | | Subsystem... | ... | ... | ... | |
| | | | Component... | ... | ... | ... | |
| | Implementation | | Physical Requirements | Software, Electrical, Mechanical... | | | |

# User Feedback



## Paradigm Shift to Standard based - Agile Model Based Systems Engineering

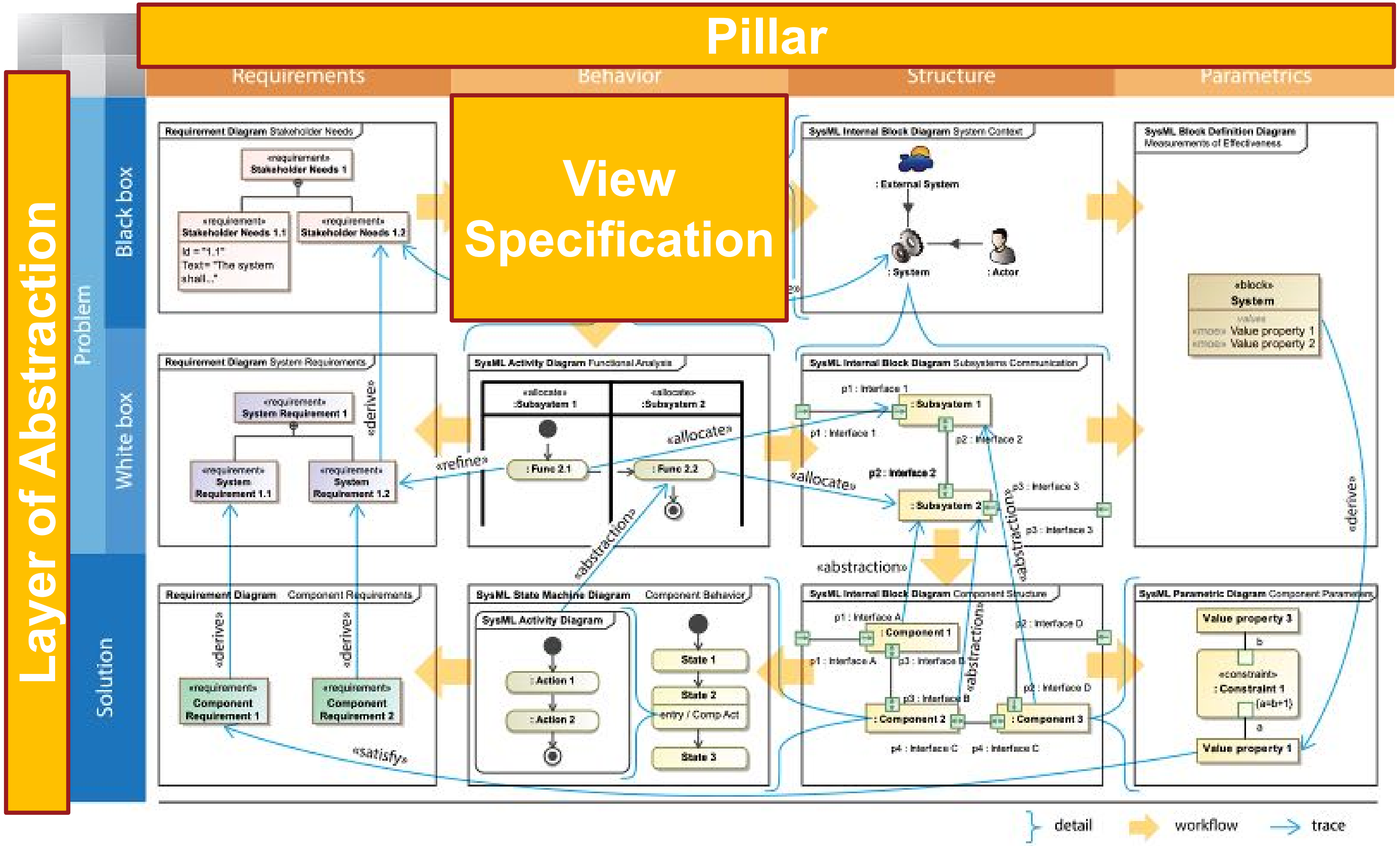
Single source of truth to support many stakeholder viewpoints via seamless integration of Requirements, Functional Safety, Quality work products…

Enriching the Models Beyond the Performance pecification with Safety, Security, Failure Mode Avoidance, and Verification / Validation

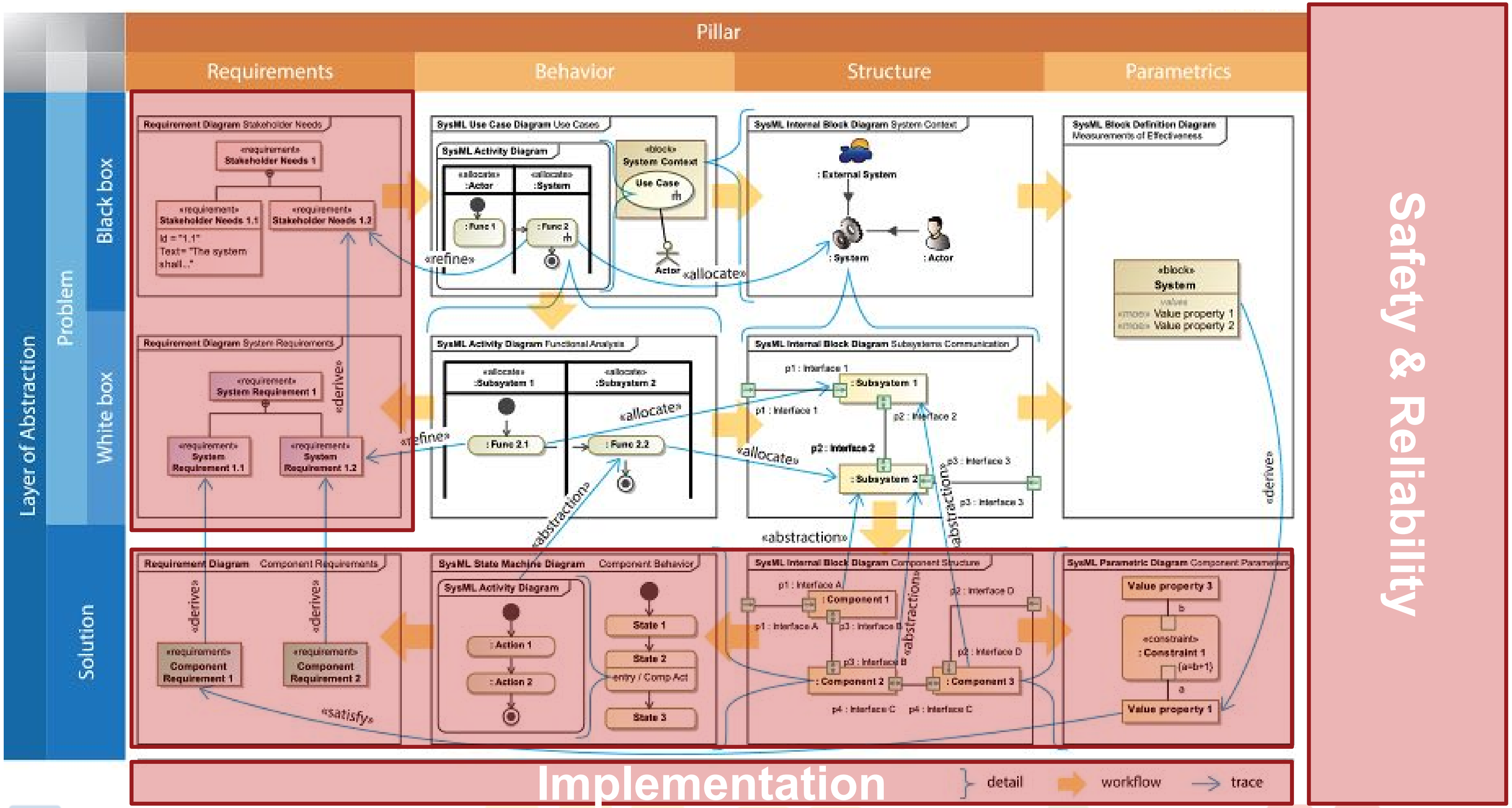
MBSE Cyber Experience Symposium - 2019

- BAE Systems
- Boeing
- Bombardier
- Honda
- Jaxa
- John Deere
- Kongsberg Defence and Aerospace
- Thyssenkrupp

# Introduction to MBSE Grid



# MBSE Grid: Key Areas of Changes

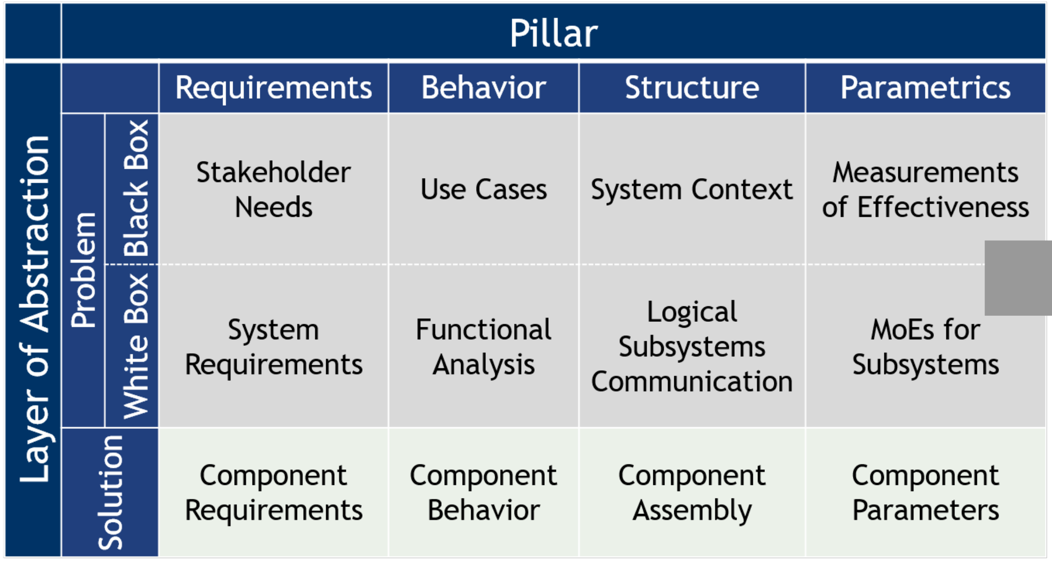


Safety & Reliability

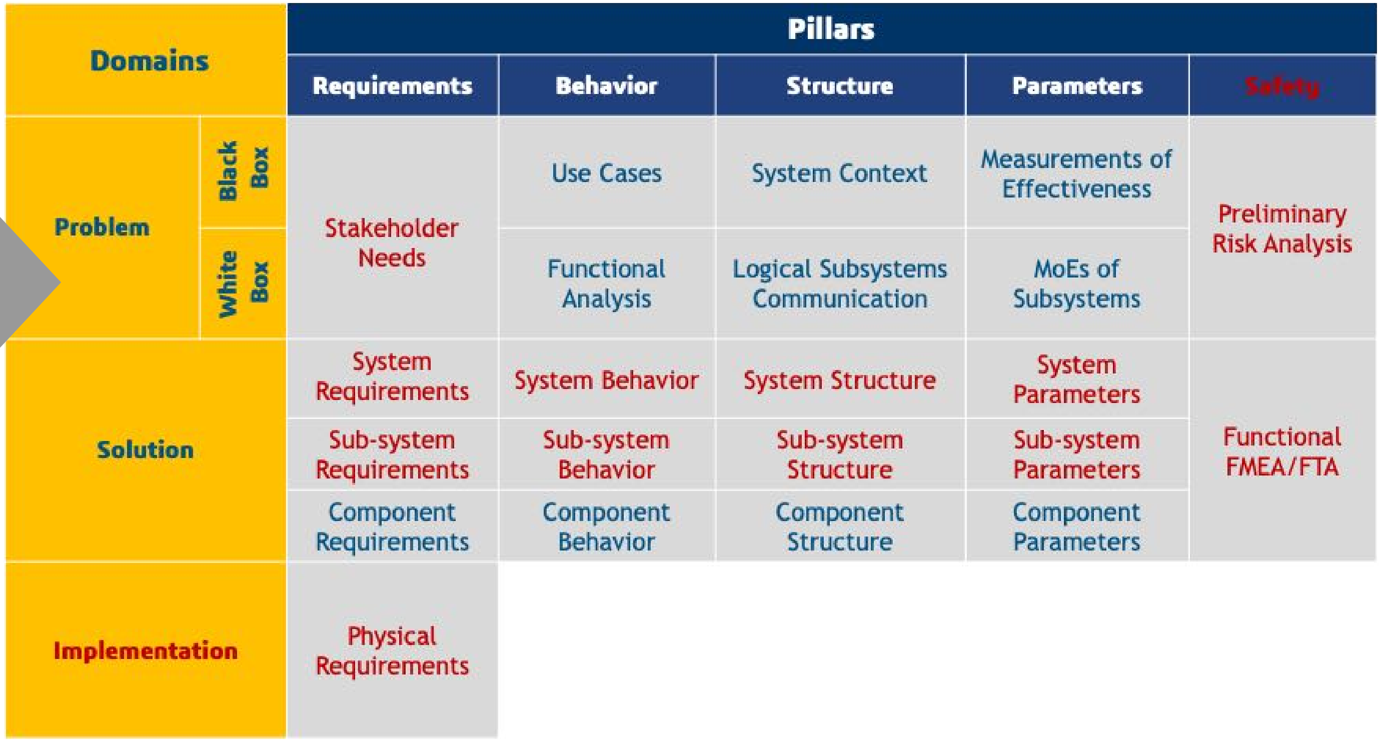
Implementation

# Proposed Modifications and Additions: Core Modifications

# MBSE Grid + Updates = MagicGrid



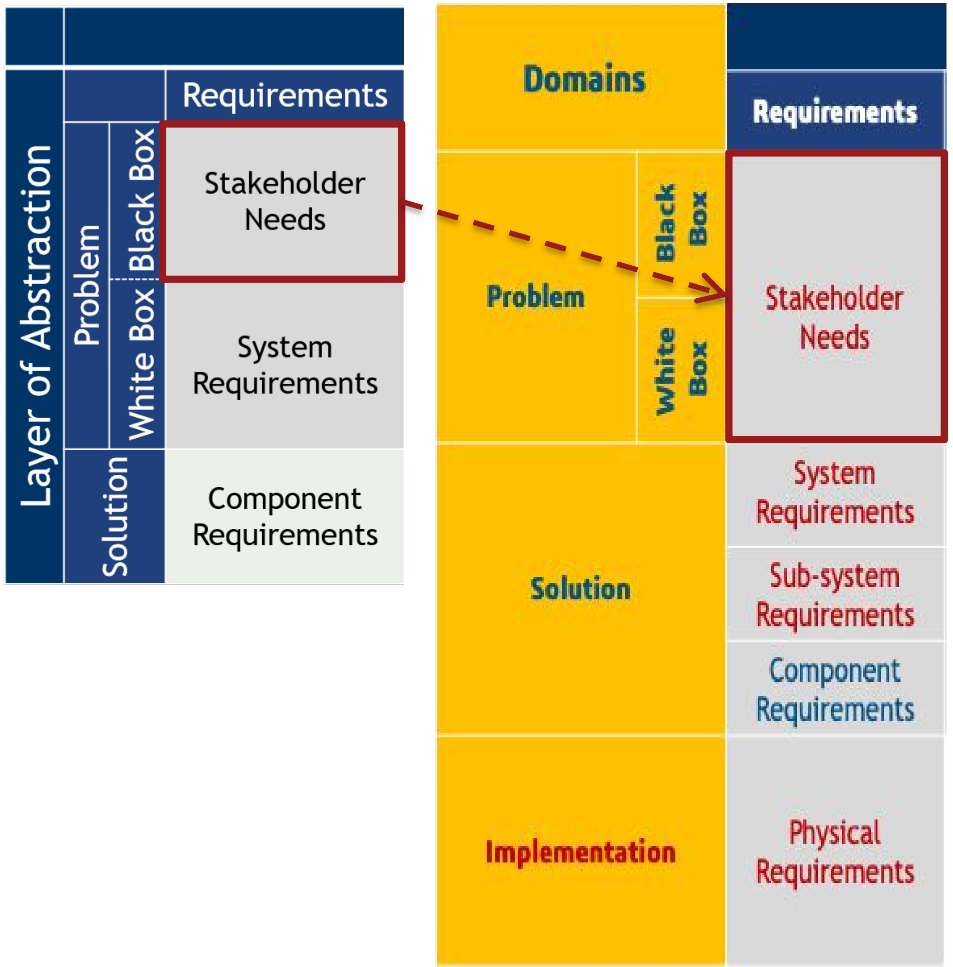
| | | | Pillar | | | |
|---|---|---|---|---|---|---|
| | | | Requirements | Behavior | Structure | Parametrics |
| Layer of Abstraction | Problem | Black Box | Stakeholder Needs | Use Cases | System Context | Measurements of Effectiveness |
| | | White Box | System Requirements | Functional Analysis | Logical Subsystems Communication | MoEs for Subsystems |
| | Solution | | Component Requirements | Component Behavior | Component Assembly | Component Parameters |



| Domains | | Pillars | | | | |
|---|---|---|---|---|---|---|
| | | Requirements | Behavior | Structure | Parameters | Safety |
| Problem | Black Box | Stakeholder Needs | Use Cases | System Context | Measurements of Effectiveness | Preliminary Risk Analysis |
| | White Box | | Functional Analysis | Logical Subsystems Communication | MoEs of Subsystems | |
| Solution | | System Requirements | System Behavior | System Structure | System Parameters | Functional FMEA/FTA |
| | | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters | |
| | | Component Requirements | Component Behavior | Component Structure | Component Parameters | |
| Implementation | | Physical Requirements | | | | |

- New domain and sub-domains including new views
- Modifications in some views
- Traceability updates

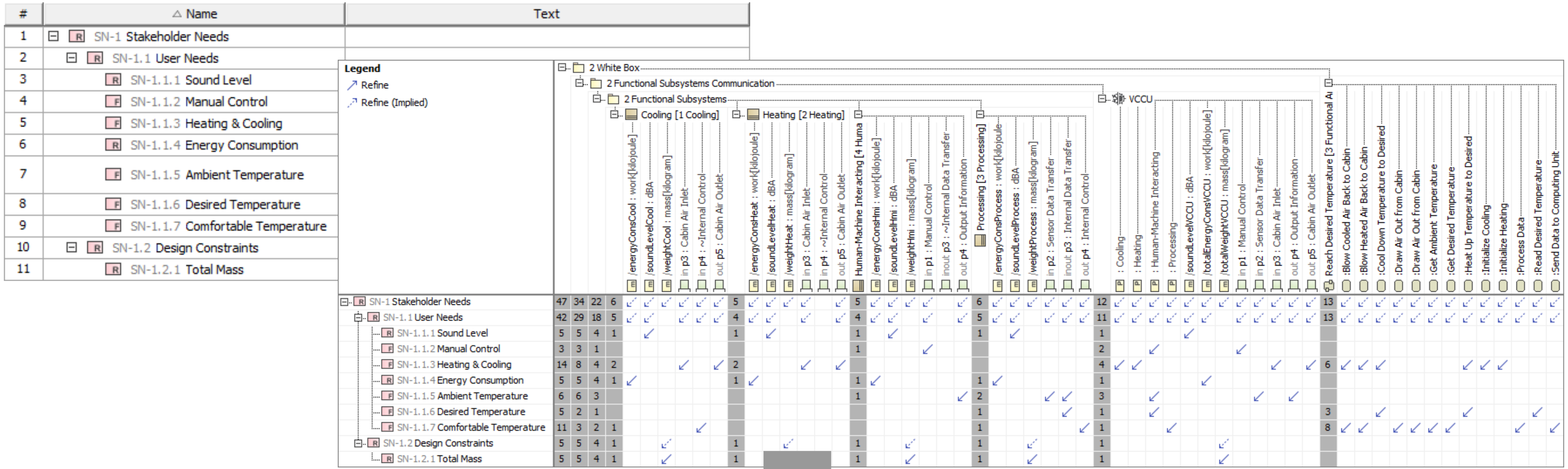
# Stakeholder Needs



- Stakeholder needs are refined during **both** phases of problem domain analysis (PDA)
- After the black-box analysis, they **continue to be the main source of information** for analyzing the expected white-box functions and structure of the System of Interest (SoI)
- Result of the PDA is SysML model that refines stakeholder needs with traceability relationships to them

# Stakeholder Needs (2)

- Result of the PDA is **input** to system requirements specification, but not system requirements specification itself



| # | Name | Text |
|---|---|---|
| 1 | R SN-1 Stakeholder Needs | |
| 2 | R SN-1.1 User Needs | |
| 3 | R SN-1.1.1 Sound Level | |
| 4 | F SN-1.1.2 Manual Control | |
| 5 | F SN-1.1.3 Heating & Cooling | |
| 6 | R SN-1.1.4 Energy Consumption | |
| 7 | F SN-1.1.5 Ambient Temperature | |
| 8 | F SN-1.1.6 Desired Temperature | |
| 9 | F SN-1.1.7 Comfortable Temperature | |
| 10 | R SN-1.2 Design Constraints | |
| 11 | R SN-1.2.1 Total Mass | |

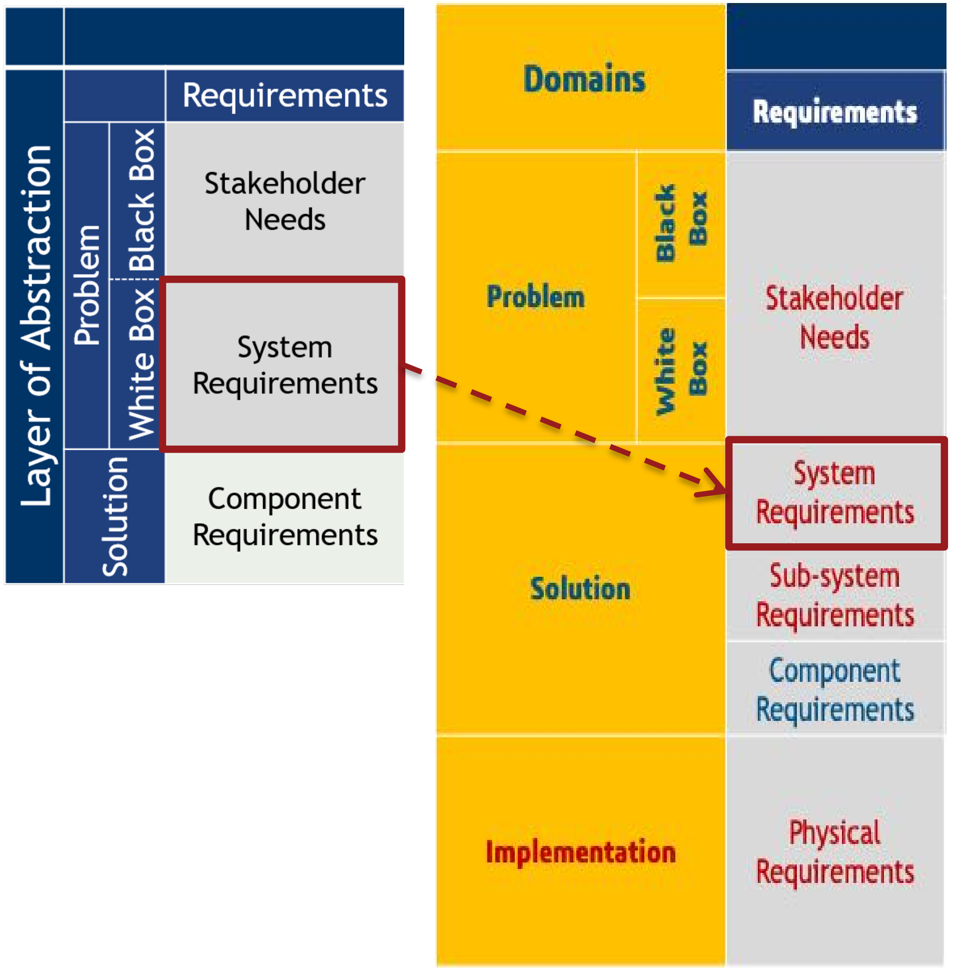
**Legend**
↗ Refine
⇢ Refine (Implied)

2 White Box
2 Functional Subsystems Communication
2 Functional Subsystems
Cooling [1 Cooling]
Heating [2 Heating]
VCCU

| | 2 White Box | 2 Functional Subsystems Communication | 2 Functional Subsystems | Cooling [1 Cooling] | /energyConsCool : work[kilojoule] | /soundLevelCool : dBA | /weightCool : mass[kilogram] | in p3 : Cabin Air Inlet | in p4 : ~Internal Control | out p5 : Cabin Air Outlet | Heating [2 Heating] | /energyConsHeat : work[kilojoule] | /soundLevelHeat : dBA | /weightHeat : mass[kilogram] | in p3 : Cabin Air Inlet | in p4 : ~Internal Control | out p5 : Cabin Air Outlet | Human-Machine Interacting [4 Huma | /energyConsHmi : work[kilojoule] | /soundLevelHmi : dBA | /weightHmi : mass[kilogram] | in p1 : Manual Control | inout p3 : ~Internal Data Transfer | out p4 : Output Information | Processing [3 Processing] | /energyConsProcess : work[kilojoule | /soundLevelProcess : dBA | /weightProcess : mass[kilogram] | in p2 : Sensor Data Transfer | inout p3 : Internal Data Transfer | out p4 : Internal Control | VCCU | : Cooling | : Heating | : Human-Machine Interacting | : Processing | /soundLevelVCCU : dBA | /totalEnergyConsVCCU : work[kilojoule] | /totalWeightVCCU : mass[kilogram] | in p1 : Manual Control | in p2 : Sensor Data Transfer | in p3 : Cabin Air Inlet | out p4 : Output Information | out p5 : Cabin Air Outlet | Reach Desired Temperature [3 Functional Ai | :Blow Cooled Air Back to Cabin | :Blow Heated Air Back to Cabin | :Cool Down Temperature to Desired | :Draw Air Out from Cabin | :Draw Air Out from Cabin | :Get Ambient Temperature | :Get Desired Temperature | :Heat Up Temperature to Desired | :Initialize Cooling | :Initialize Heating | :Process Data | :Read Desired Temperature | :Send Data to Computing Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R SN-1 Stakeholder Needs | 47 | 34 | 22 | 6 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | 5 | ⇠ | ⇠ | ⇠ | ⇠ | | ⇠ | 5 | ⇠ | ⇠ | ⇠ | ⇠ | | ⇠ | 6 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | 12 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | 13 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ |
| R SN-1.1 User Needs | 42 | 29 | 18 | 5 | ⇠ | ⇠ | | ⇠ | ⇠ | ⇠ | 4 | ⇠ | ⇠ | | ⇠ | | ⇠ | 4 | ⇠ | ⇠ | | ⇠ | | ⇠ | 5 | ⇠ | ⇠ | | ⇠ | ⇠ | ⇠ | 11 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | 13 | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ | ⇠ |
| R SN-1.1.1 Sound Level | 5 | 5 | 4 | 1 | | ↙ | | | | | 1 | | ↙ | | | | | 1 | | ↙ | | | | | 1 | | ↙ | | | | | 1 | | | | | ↙ | | | | | | | | | | | | | | | | | | | | | |
| F SN-1.1.2 Manual Control | 3 | 3 | 1 | | | | | | | | | | | | | | | 1 | | | | ↙ | | | | | | | | | | 2 | | | ↙ | | | | | ↙ | | | | | | | | | | | | | | | | | | |
| F SN-1.1.3 Heating & Cooling | 14 | 8 | 4 | 2 | | | | ↙ | | ↙ | 2 | | | | ↙ | | ↙ | | | | | | | | | | | | | | | 4 | ↙ | ↙ | | | | | | | | ↙ | | ↙ | 6 | ↙ | ↙ | ↙ | | | | | ↙ | ↙ | ↙ | | | |
| R SN-1.1.4 Energy Consumption | 5 | 5 | 4 | 1 | ↙ | | | | | | 1 | ↙ | | | | | | 1 | ↙ | | | | | | 1 | ↙ | | | | | | 1 | | | | | | ↙ | | | | | | | | | | | | | | | | | | | | |
| F SN-1.1.5 Ambient Temperature | 6 | 6 | 3 | | | | | | | | | | | | | | | 1 | | | | | | ↙ | 2 | | | | ↙ | ↙ | | 3 | | | ↙ | | | | | | ↙ | | ↙ | | | | | | | | | | | | | | | |
| F SN-1.1.6 Desired Temperature | 5 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | 1 | | | | | ↙ | | 1 | | | ↙ | | | | | | | | | | 3 | | | ↙ | | | | | ↙ | | | | ↙ | |
| F SN-1.1.7 Comfortable Temperature | 11 | 3 | 2 | 1 | | | | | ↙ | | | | | | | | | | | | | | | | 1 | | | | | | ↙ | 1 | | | | ↙ | | | | | | | | | 8 | ↙ | ↙ | | ↙ | ↙ | ↙ | ↙ | | | | ↙ | | ↙ |
| R SN-1.2 Design Constraints | 5 | 5 | 4 | 1 | | | ⇠ | | | | 1 | | | ⇠ | | | | 1 | | | ⇠ | | | | 1 | | | ⇠ | | | | 1 | | | | | | | ⇠ | | | | | | | | | | | | | | | | | | | |
| R SN-1.2.1 Total Mass | 5 | 5 | 4 | 1 | | | ↙ | | | | 1 | | | ↙ | | | | 1 | | | ↙ | | | | 1 | | | ↙ | | | | 1 | | | | | | | ↙ | | | | | | | | | | | | | | | | | | | |

System Requirements Specification

# System Requirements



Layer of Abstraction

Problem

Black Box

White Box

Solution

Requirements

Stakeholder Needs

System Requirements

Component Requirements

Domains

Requirements

Problem

Black Box

White Box

Stakeholder Needs

Solution

System Requirements

Sub-system Requirements

Component Requirements

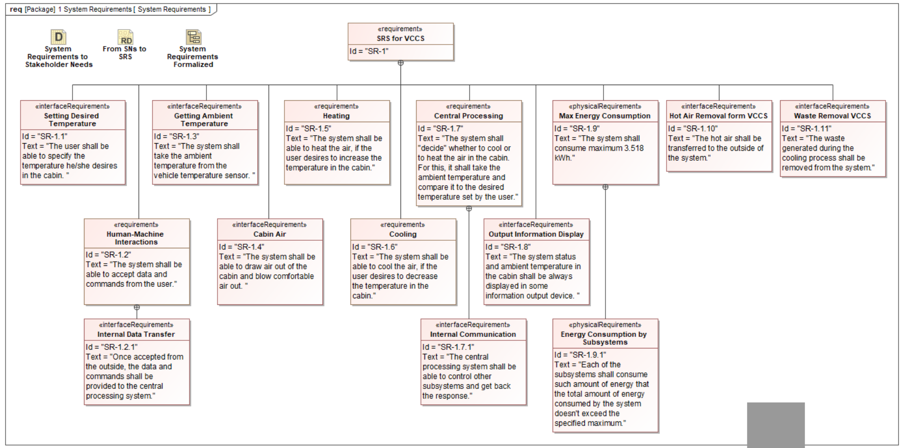
Implementation

Physical Requirements

- System requirements (SR) specification can be **produced only if** you fully understand the stakeholder needs
- SR are not only **derived** from stakeholder needs – they also **refine** the SysML model of the problem domain

# System Requirements (2)

- SR specification is **input** to the high-level solution architecture (HLSA)



req [Package] 1 System Requirements [ System Requirements ]

System Requirements to Stakeholder Needs

From SNs to SRS

System Requirements Formalized

«requirement»
**SRS for VCCS**
Id = "SR-1"

«interfaceRequirement»
**Setting Desired Temperature**
Id = "SR-1.1"
Text = "The user shall be able to specify the temperature he/she desires in the cabin. "

«interfaceRequirement»
**Getting Ambient Temperature**
Id = "SR-1.3"
Text = "The system shall take the ambient temperature from the vehicle temperature sensor. "

«requirement»
**Heating**
Id = "SR-1.5"
Text = "The system shall be able to heat the air, if the user desires to increase the temperature in the cabin."

«requirement»
**Central Processing**
Id = "SR-1.7"
Text = "The system shall "decide" whether to cool or to heat the air in the cabin. For this, it shall take the ambient temperature and compare it to the desired temperature set by the user."

«physicalRequirement»
**Max Energy Consumption**
Id = "SR-1.9"
Text = "The system shall consume maximum 3.518 kWh."

«interfaceRequirement»
**Hot Air Removal form VCCS**
Id = "SR-1.10"
Text = "The hot air shall be transferred to the outside of the system."

«interfaceRequirement»
**Waste Removal VCCS**
Id = "SR-1.11"
Text = "The waste generated during the cooling process shall be removed from the system."

«requirement»
**Human-Machine Interactions**
Id = "SR-1.2"
Text = "The system shall be able to accept data and commands from the user."

«interfaceRequirement»
**Cabin Air**
Id = "SR-1.4"
Text = "The system shall be able to draw air out of the cabin and blow comfortable air out. "

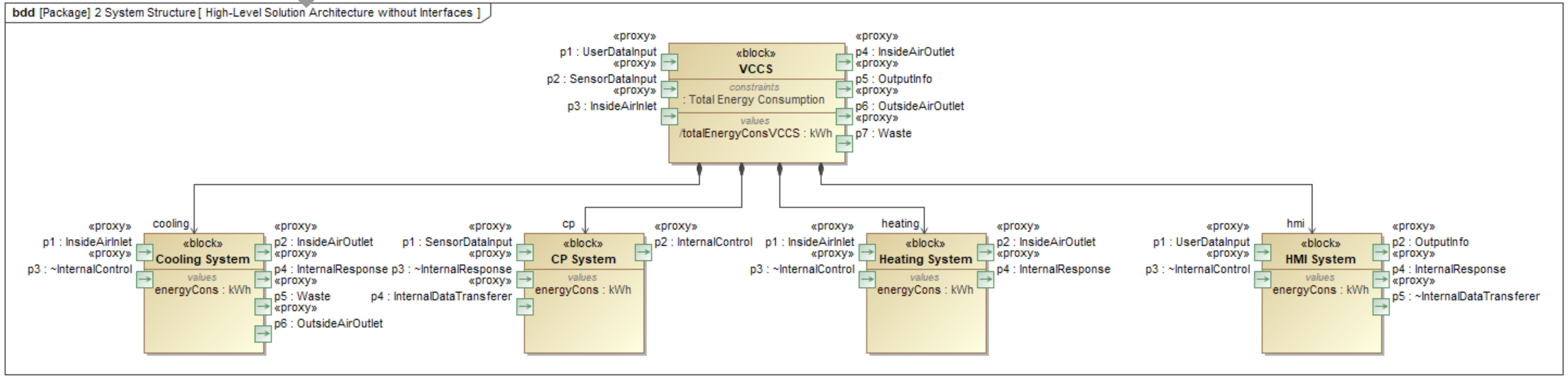
«requirement»
**Cooling**
Id = "SR-1.6"
Text = "The system shall be able to cool the air, if the user desires to decrease the temperature in the cabin."

«interfaceRequirement»
**Output Information Display**
Id = "SR-1.8"
Text = "The system status and ambient temperature in the cabin shall be always displayed in some information output device. "

«interfaceRequirement»
**Internal Data Transfer**
Id = "SR-1.2.1"
Text = "Once accepted from the outside, the data and commands shall be provided to the central processing system."

«interfaceRequirement»
**Internal Communication**
Id = "SR-1.7.1"
Text = "The central processing system shall be able to control other subsystems and get back the response."

«physicalRequirement»
**Energy Consumption by Subsystems**
Id = "SR-1.9.1"
Text = "Each of the subsystems shall consume such amount of energy that the total amount of energy consumed by the system doesn't exceed the specified maximum."



bdd [Package] 2 System Structure [ High-Level Solution Architecture without Interfaces ]

«block»
**VCCS**
constraints
: Total Energy Consumption
values
/totalEnergyConsVCCS : kWh
«proxy» p1 : UserDataInput
«proxy» p2 : SensorDataInput
«proxy» p3 : InsideAirInlet
«proxy» p4 : InsideAirOutlet
«proxy» p5 : OutputInfo
«proxy» p6 : OutsideAirOutlet
«proxy» p7 : Waste

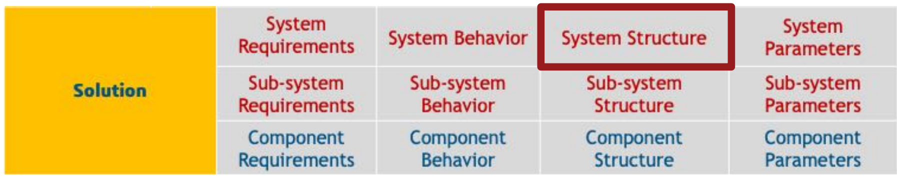
cooling
«block»
**Cooling System**
values
energyCons : kWh
«proxy» p1 : InsideAirInlet
«proxy» p3 : ~InternalControl
«proxy» p2 : InsideAirOutlet
«proxy» p4 : InternalResponse
«proxy» p5 : Waste
«proxy» p6 : OutsideAirOutlet

cp
«block»
**CP System**
values
energyCons : kWh
«proxy» p1 : SensorDataInput
«proxy» p3 : ~InternalResponse
«proxy» p4 : InternalDataTransferer
«proxy» p2 : InternalControl

heating
«block»
**Heating System**
values
energyCons : kWh
«proxy» p1 : InsideAirInlet
«proxy» p3 : ~InternalControl
«proxy» p2 : InsideAirOutlet
«proxy» p4 : InternalResponse

hmi
«block»
**HMI System**
values
energyCons : kWh
«proxy» p1 : UserDataInput
«proxy» p3 : ~InternalControl
«proxy» p2 : OutputInfo
«proxy» p4 : InternalResponse
«proxy» p5 : ~InternalDataTransferer

# System Structure

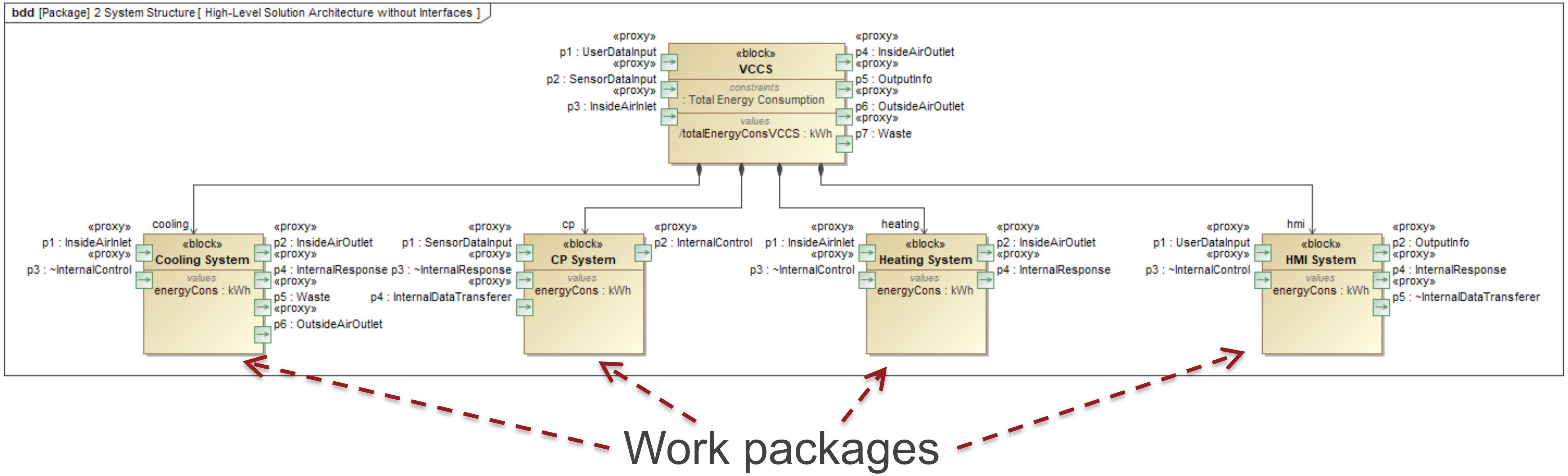


| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- A **new cell**, along with entire top-level sub-domain
- **Defines** the initial and the final tasks of **building the logical solution architecture** of the Sol
- Both tasks are **systems engineer’s** responsibility

# System Structure: HLSA

- The **first task** is to create the **HLSA model**, which **captures** the **logical subsystems** of the SoI and **identifies work packages**, as each subsystem is allocated to a separate engineering team



Work packages

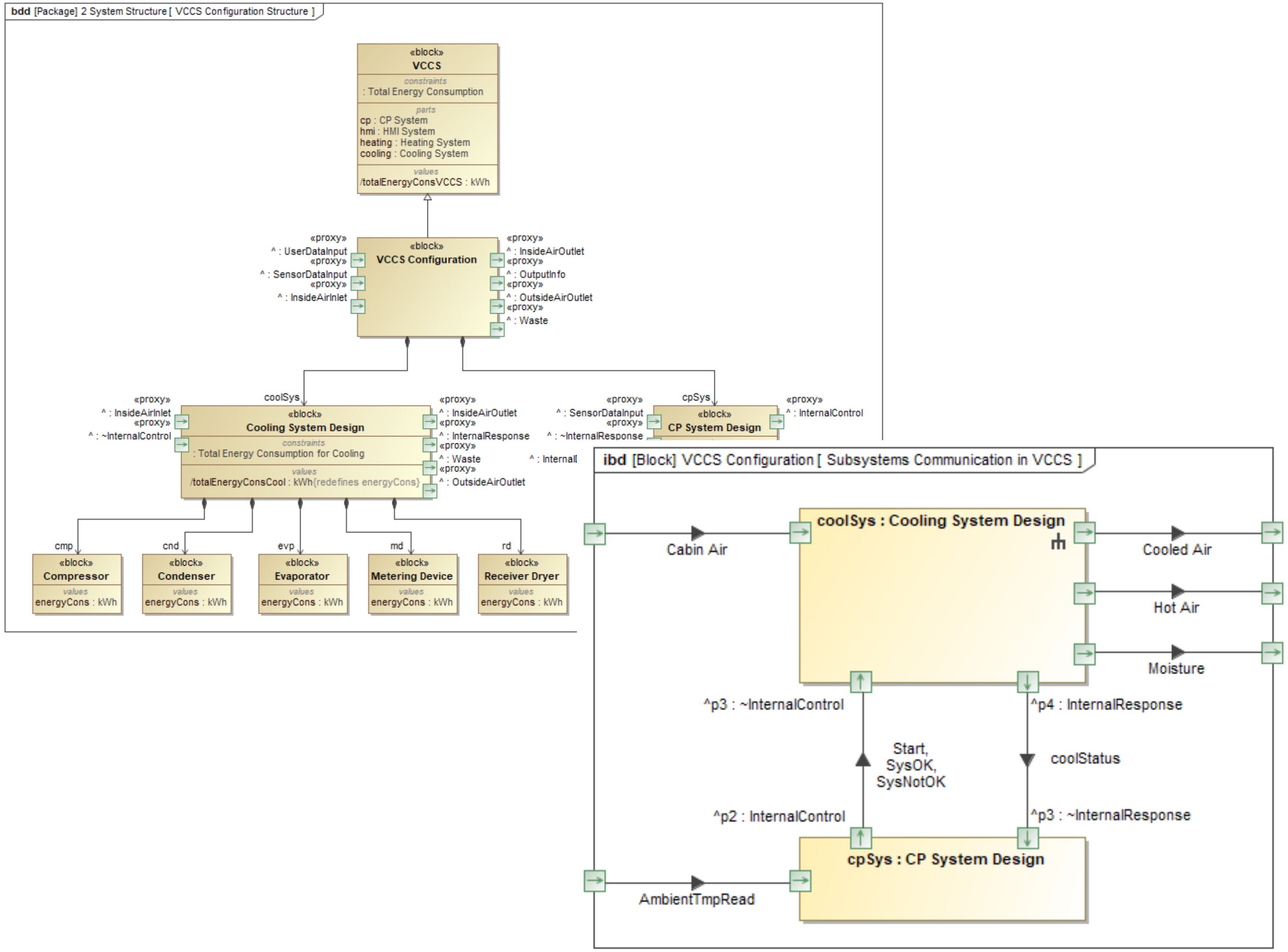
- HLSA model **satisfies** system requirements specification

# System Structure: System Configuration

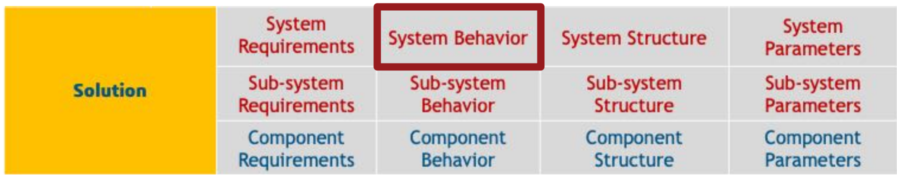
- The **final task** is build the **integrated model** of the whole SoI

- Once all engineering teams produce their **solution architectures for each subsystem**, the systems engineer is able to **integrate** them into whole

- **One or more** system configuration models can be produced



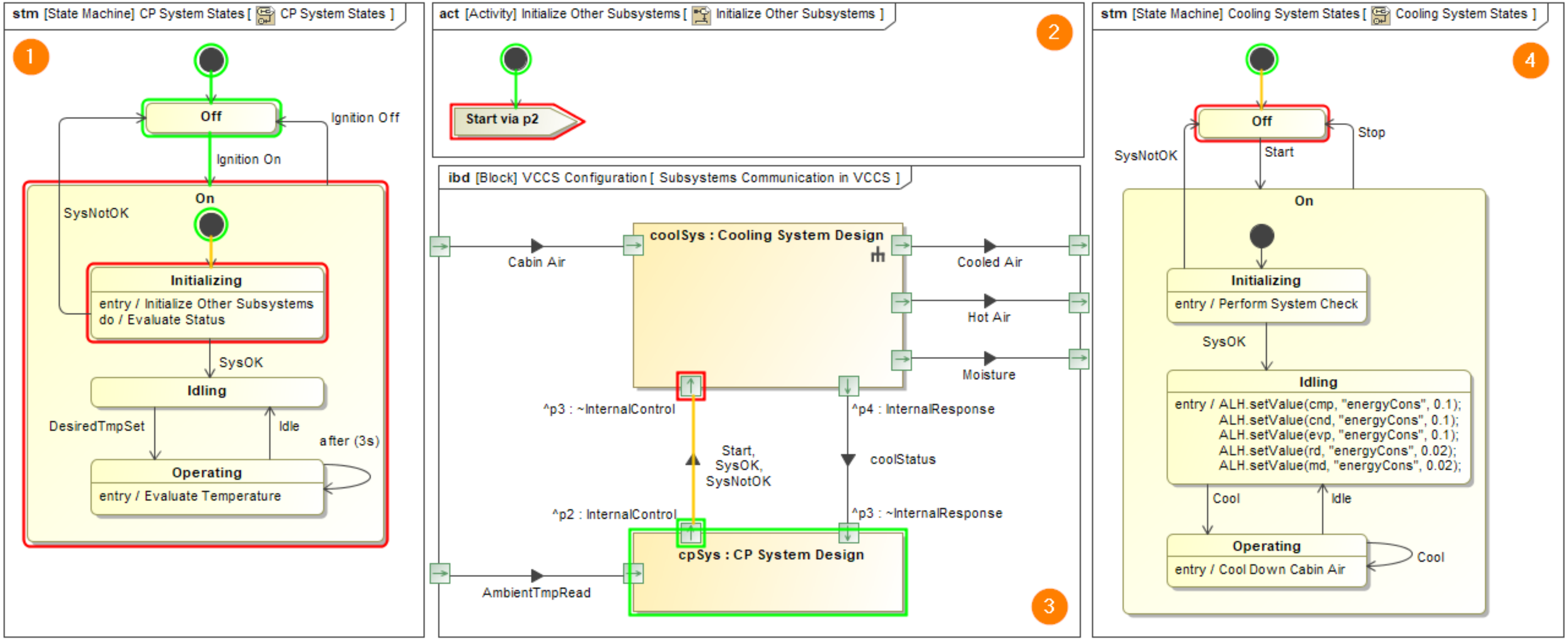
# System Behavior



| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

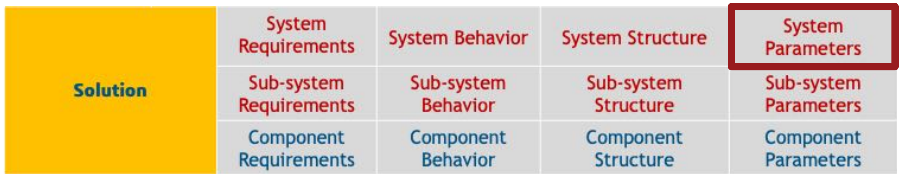
- A **new cell**, added to define how to build the **behavioral model** of the SoI
- It can be **skipped** in the HLSA model
- The **integrated model** of the selected system configuration includes the **behavioral models of all logical subsystems** of the SoI (once they are created and integrated in the system configuration model)
- The integrated system behavior model and interface compatibility **can be validated by utilizing the simulation capabilities** of the modeling tool

# System Behavior: Simulation



stm [State Machine] CP System States [ CP System States ]
1
Off
Ignition Off
Ignition On
On
SysNotOK
Initializing
entry / Initialize Other Subsystems
do / Evaluate Status
SysOK
Idling
DesiredTmpSet
Idle
after (3s)
Operating
entry / Evaluate Temperature
act [Activity] Initialize Other Subsystems [ Initialize Other Subsystems ]
2
Start via p2
ibd [Block] VCCS Configuration [ Subsystems Communication in VCCS ]
coolSys : Cooling System Design
Cabin Air
Cooled Air
Hot Air
Moisture
^p3 : ~InternalControl
^p4 : InternalResponse
Start,
SysOK,
SysNotOK
coolStatus
^p2 : InternalControl
^p3 : ~InternalResponse
cpSys : CP System Design
AmbientTmpRead
3
stm [State Machine] Cooling System States [ Cooling System States ]
4
Off
Stop
SysNotOK
Start
On
Initializing
entry / Perform System Check
SysOK
Idling
entry / ALH.setValue(cmp, "energyCons", 0.1);
ALH.setValue(cnd, "energyCons", 0.1);
ALH.setValue(evp, "energyCons", 0.1);
ALH.setValue(rd, "energyCons", 0.02);
ALH.setValue(md, "energyCons", 0.02);
Cool
Idle
Operating
entry / Cool Down Cabin Air
Cool

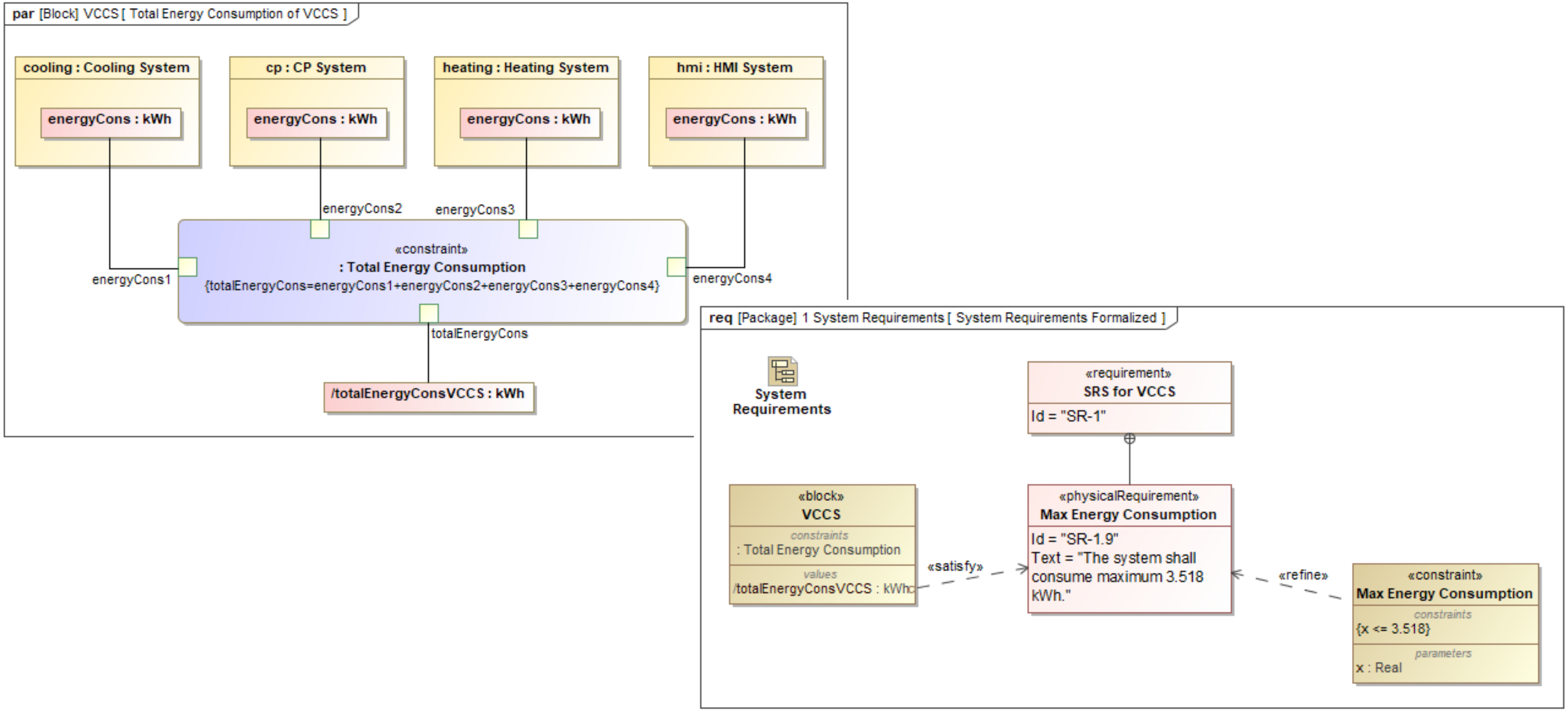
# System Parameters



| | | | | |
|---|---|---|---|---|
| **Solution** | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- A **new cell**, aded to define how to specify the **method for calculating system parameters**, which are derived from MoEs of the SoI (Problem domain)
- The simulation capabilities of the modeling tool enable users to **calculate** system parameters and **automatically** verify relevant system requirements
- System parameters can be specified as soon as the system structure is captured in the model. Even as abstract as in the HLSA model

# System Parameters: Method Definition



**par** [Block] VCCS [ Total Energy Consumption of VCCS ]

cooling : Cooling System

energyCons : kWh

cp : CP System

energyCons : kWh

heating : Heating System

energyCons : kWh

hmi : HMI System

energyCons : kWh

energyCons1

energyCons2

energyCons3

energyCons4

«constraint»
: Total Energy Consumption
{totalEnergyCons=energyCons1+energyCons2+energyCons3+energyCons4}

totalEnergyCons

/totalEnergyConsVCCS : kWh

**req** [Package] 1 System Requirements [ System Requirements Formalized ]

System Requirements

«requirement»
SRS for VCCS
Id = "SR-1"

«block»
VCCS
constraints
: Total Energy Consumption
values
/totalEnergyConsVCCS : kWh

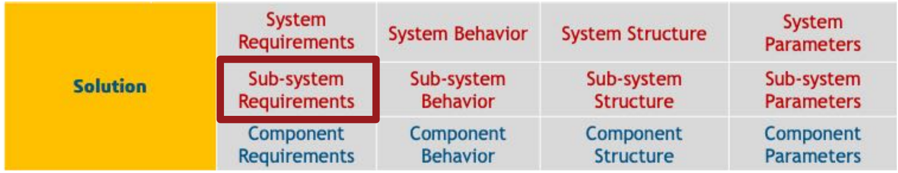
«satisfy»

«physicalRequirement»
Max Energy Consumption
Id = "SR-1.9"
Text = "The system shall consume maximum 3.518 kWh."

«refine»

«constraint»
Max Energy Consumption
constraints
{x <= 3.518}
parameters
x : Real

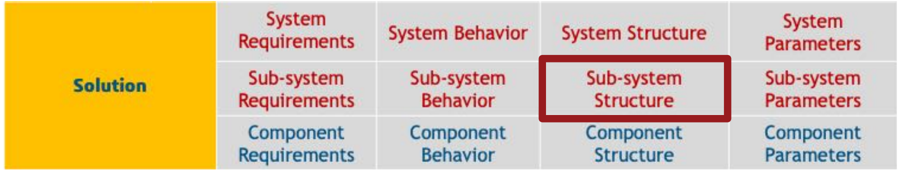
# Subsystem Requirements



| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- A **new cell**, along with entire mid-level sub-domain
- Conveys that the system requirements specification is normally produced in **several iterations** and evolves gradually: from system-level to subsystem-level requirements and then from subsystem-level to component-level requirements
- Subsystem requirements specification is **input** to subsystem-level solution architecture
- They are **satisfied** by the elements capturing the subsystem-level solution architecture

# Subsystem Structure

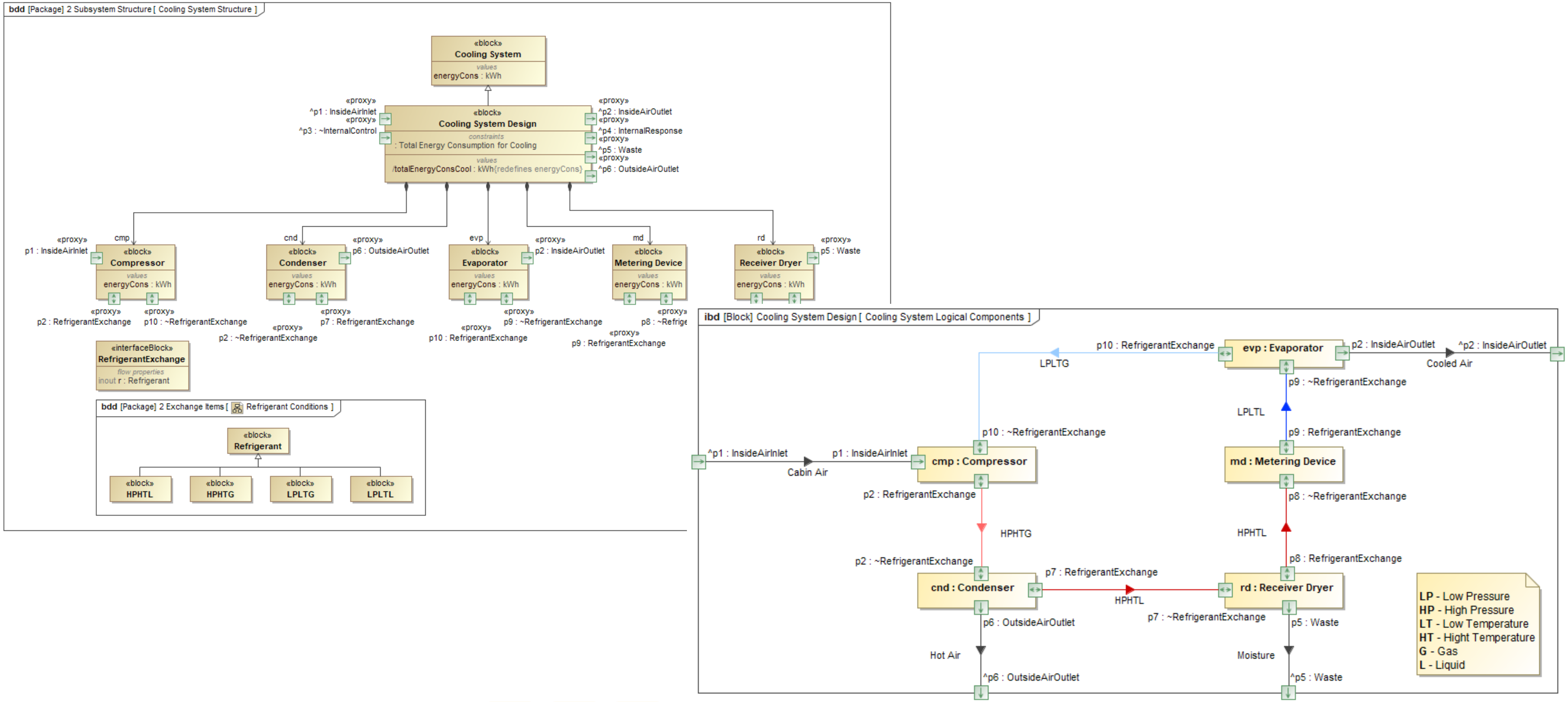


| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

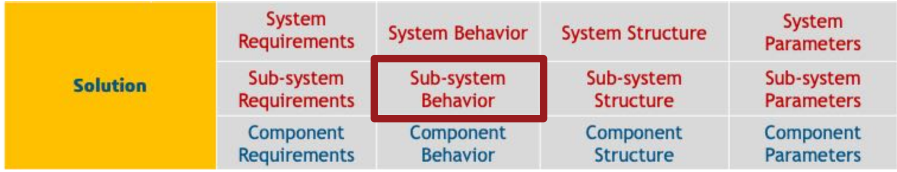
- A **new view**, added to define how to build the **solution architecture of the logical subsystem**

- To ensure the integrity of diverse solution architecture models, the appointed engineering teams **get the interfaces from the HLSA model**, and must deal with them

# Subsystem Structure: Components & Internal Communications


bdd [Package] 2 Subsystem Structure [ Cooling System Structure ]
«block» Cooling System
values
energyCons : kWh
«block» Cooling System Design
constraints
: Total Energy Consumption for Cooling
values
/totalEnergyConsCool : kWh{redefines energyCons}
«proxy» ^p1 : InsideAirInlet
«proxy» ^p3 : ~InternalControl
«proxy» ^p2 : InsideAirOutlet
«proxy» ^p4 : InternalResponse
«proxy» ^p5 : Waste
«proxy» ^p6 : OutsideAirOutlet
cmp
cnd
evp
md
rd
«block» Compressor
values
energyCons : kWh
«proxy» p1 : InsideAirInlet
«proxy» p2 : RefrigerantExchange
«proxy» p10 : ~RefrigerantExchange
«block» Condenser
values
energyCons : kWh
«proxy» p6 : OutsideAirOutlet
«proxy» p2 : ~RefrigerantExchange
«proxy» p7 : RefrigerantExchange
«block» Evaporator
values
energyCons : kWh
«proxy» p2 : InsideAirOutlet
«proxy» p10 : RefrigerantExchange
«proxy» p9 : ~RefrigerantExchange
«block» Metering Device
values
energyCons : kWh
«proxy» p9 : RefrigerantExchange
«proxy» p8 : ~Refrige
«block» Receiver Dryer
values
energyCons : kWh
«proxy» p5 : Waste
«interfaceBlock» RefrigerantExchange
flow properties
inout r : Refrigerant
bdd [Package] 2 Exchange Items [ Refrigerant Conditions ]
«block» Refrigerant
«block» HPHTL
«block» HPHTG
«block» LPLTG
«block» LPLTL
ibd [Block] Cooling System Design [ Cooling System Logical Components ]
evp : Evaporator
p10 : RefrigerantExchange
p2 : InsideAirOutlet
^p2 : InsideAirOutlet
Cooled Air
LPLTG
p9 : ~RefrigerantExchange
LPLTL
p10 : ~RefrigerantExchange
p9 : RefrigerantExchange
^p1 : InsideAirInlet
p1 : InsideAirInlet
Cabin Air
cmp : Compressor
md : Metering Device
p2 : RefrigerantExchange
p8 : ~RefrigerantExchange
HPHTG
HPHTL
p2 : ~RefrigerantExchange
p7 : RefrigerantExchange
p8 : RefrigerantExchange
cnd : Condenser
rd : Receiver Dryer
HPHTL
p7 : ~RefrigerantExchange
p6 : OutsideAirOutlet
p5 : Waste
Hot Air
Moisture
^p6 : OutsideAirOutlet
^p5 : Waste
LP - Low Pressure
HP - High Pressure
LT - Low Temperature
HT - Hight Temperature
G - Gas
L - Liquid

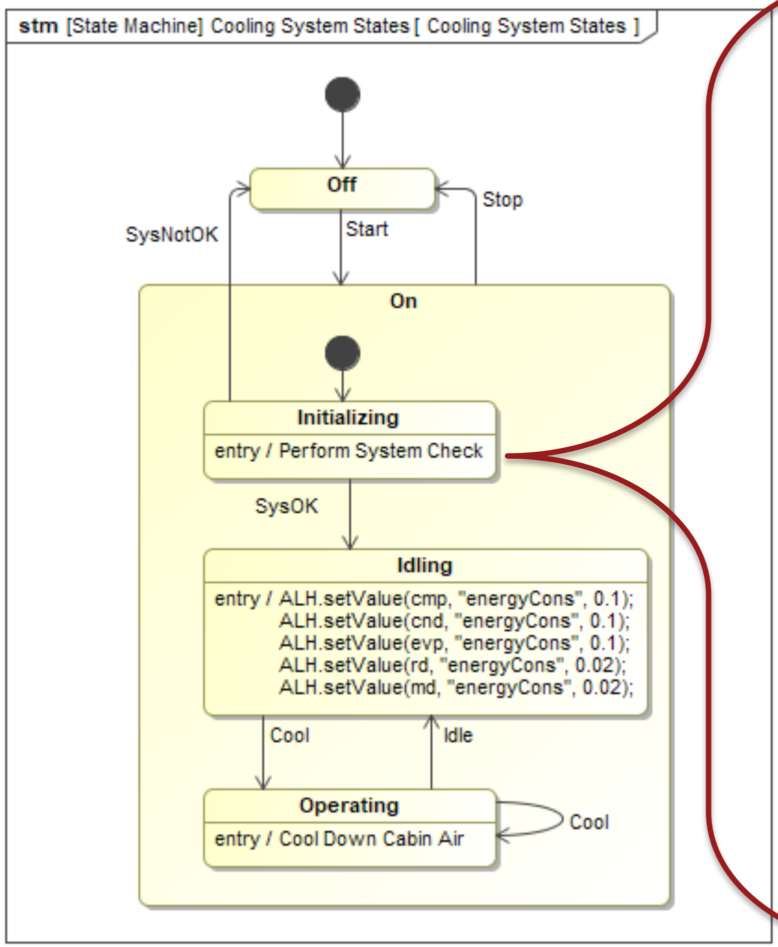
# Subsystem Behavior



| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- A **new view**, added to define how to model the **complex behavior** of the given logical subsystem
- To ensure the integrity of diverse solution architecture models, the appointed engineering team **receives the system-level signals from the HLSA** model and should take them into consideration when modeling the behavior of the particular subsystem

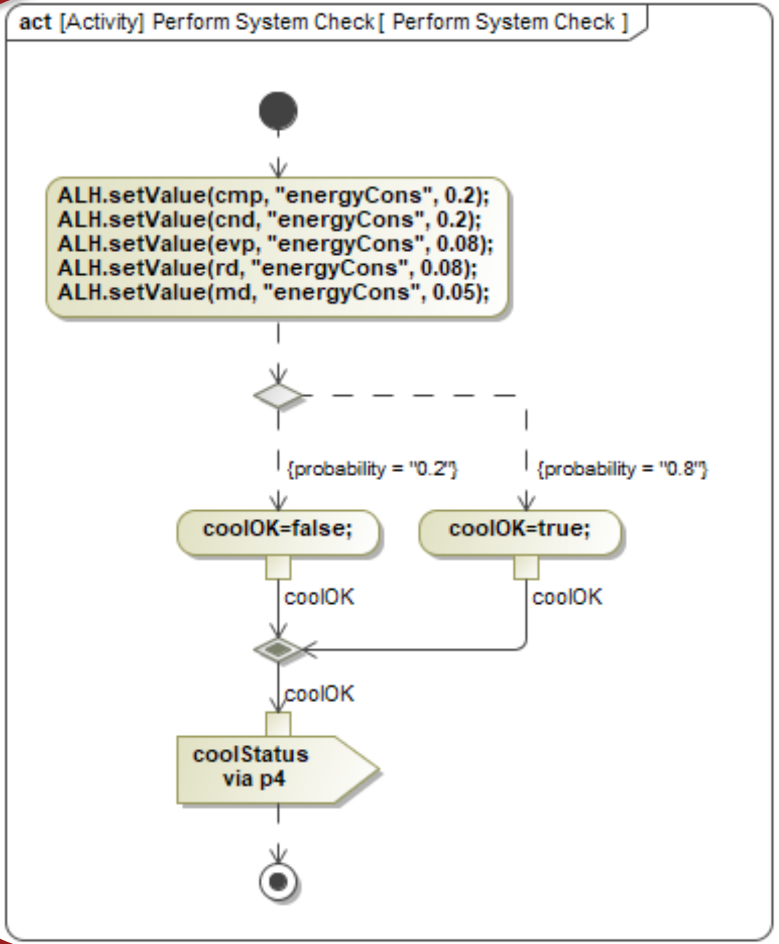
# System Behavior: States and Internal Behaviors


stm [State Machine] Cooling System States [ Cooling System States ]
Off
Stop
Start
SysNotOK
On
Initializing
SysOK
Idling
Cool
Idle
Operating
Cool

entry / Perform System Check

```
entry / ALH.setValue(cmp, "energyCons", 0.1);
        ALH.setValue(cnd, "energyCons", 0.1);
        ALH.setValue(evp, "energyCons", 0.1);
        ALH.setValue(rd, "energyCons", 0.02);
        ALH.setValue(md, "energyCons", 0.02);
```

entry / Cool Down Cabin Air

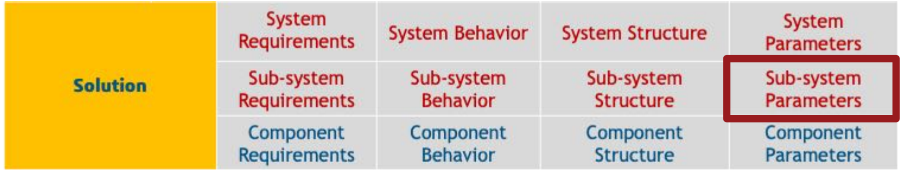


```
ALH.setValue(cmp, "energyCons", 0.2);
ALH.setValue(cnd, "energyCons", 0.2);
ALH.setValue(evp, "energyCons", 0.08);
ALH.setValue(rd, "energyCons", 0.08);
ALH.setValue(md, "energyCons", 0.05);
```

```
coolOK=false;
```

```
coolOK=true;
```

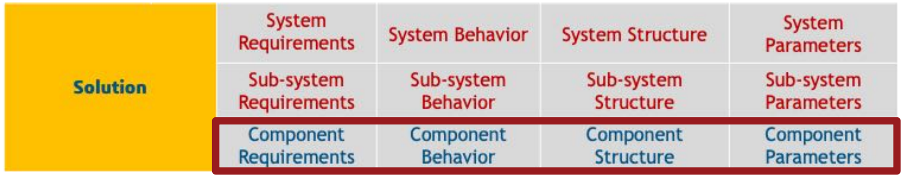
# Subsystem Parameters



| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- A **new view**, added to **correspond** with the Subsystem Structure and Subsystem Behavior views
- It describes the **method for calculating subsystem parameters** derived from MoEs or measures of performance (MoPs) of that subsystem

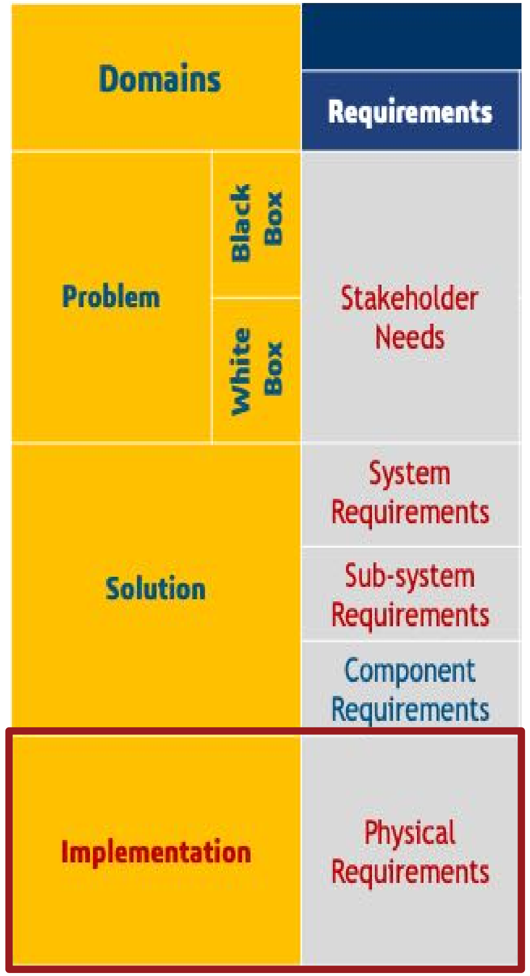
# Component Requirements, Structure, etc.



| | | | | |
|---|---|---|---|---|
| Solution | System Requirements | System Behavior | System Structure | System Parameters |
| | Sub-system Requirements | Sub-system Behavior | Sub-system Structure | Sub-system Parameters |
| | Component Requirements | Component Behavior | Component Structure | Component Parameters |

- These cells are **not new**, although their descriptions have been updated to change the keyword *“physical components”* to *“logical components”*
- Building the solution architecture of the SoI may require even greater detail than depicted in the layout if the MagicGrid framework

# Physical Requirements

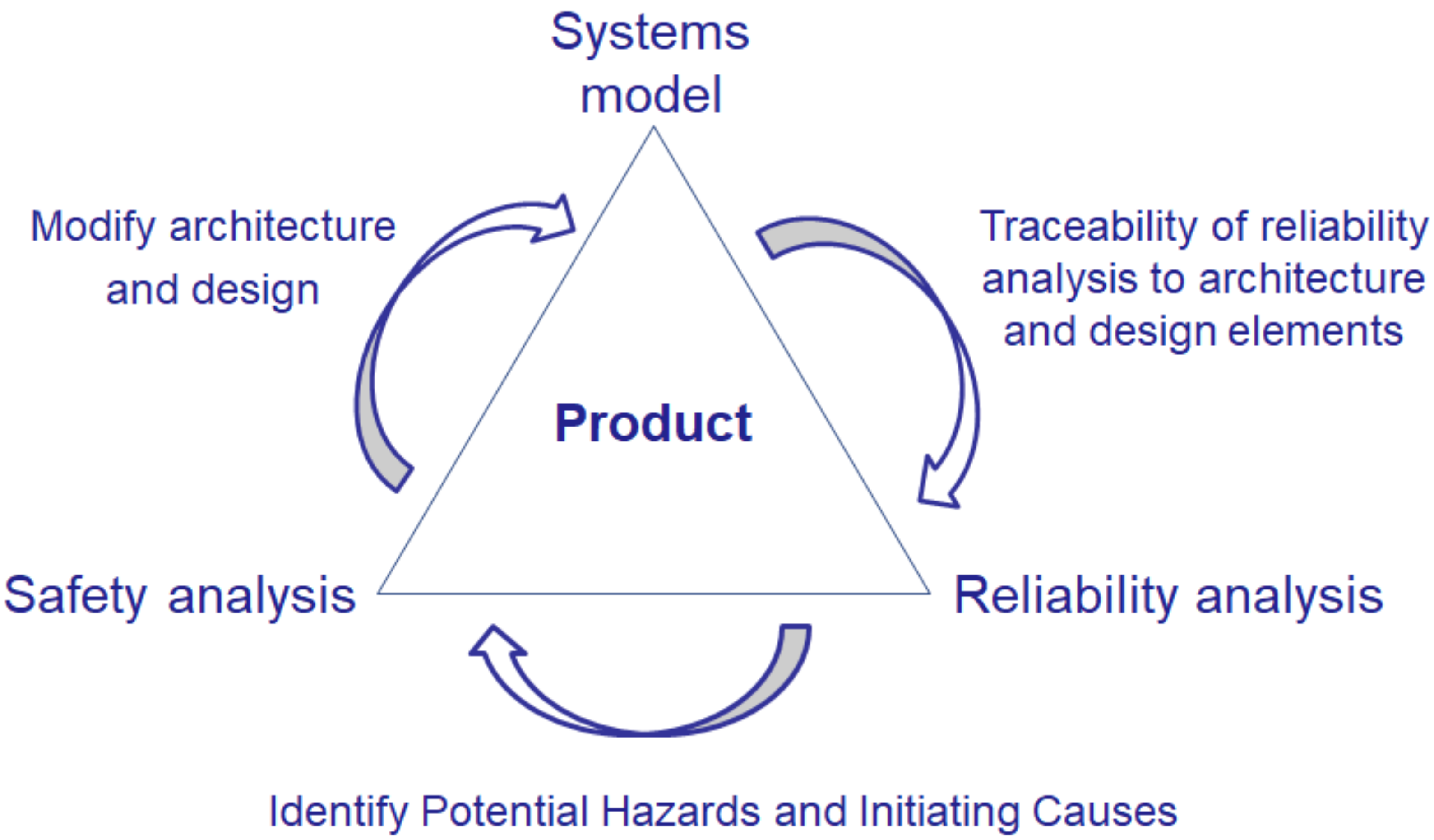


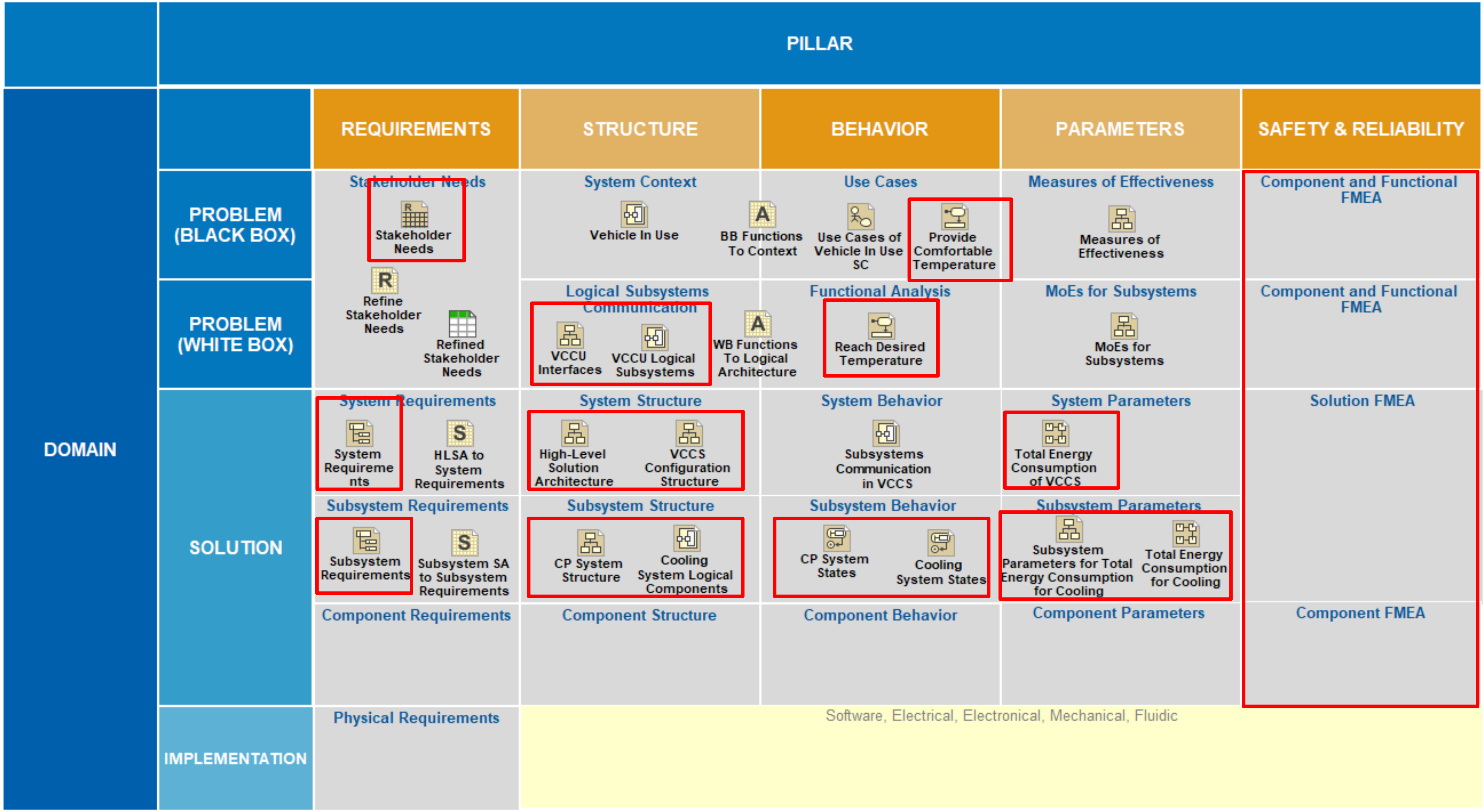
| Domains | | Requirements |
|---|---|---|
| Problem | Black Box | Stakeholder Needs |
| | White Box | |
| Solution | | System Requirements |
| | | Sub-system Requirements |
| | | Component Requirements |
| Implementation | | Physical Requirements |

- A **new domain** and a **new view**, to define how to specify and manage **detailed physical requirements** for the implementation of the selected system configuration
- Detailed physical requirements are specified **for each physical component** of the SoI (these can be Mechanical, Software, Electrical, Electronic, or Fluidic)
- Detailed physical requirements must be **derived** from the component requirements and must **refine** the solution architecture of logical components

# Safety & Reliability pillar

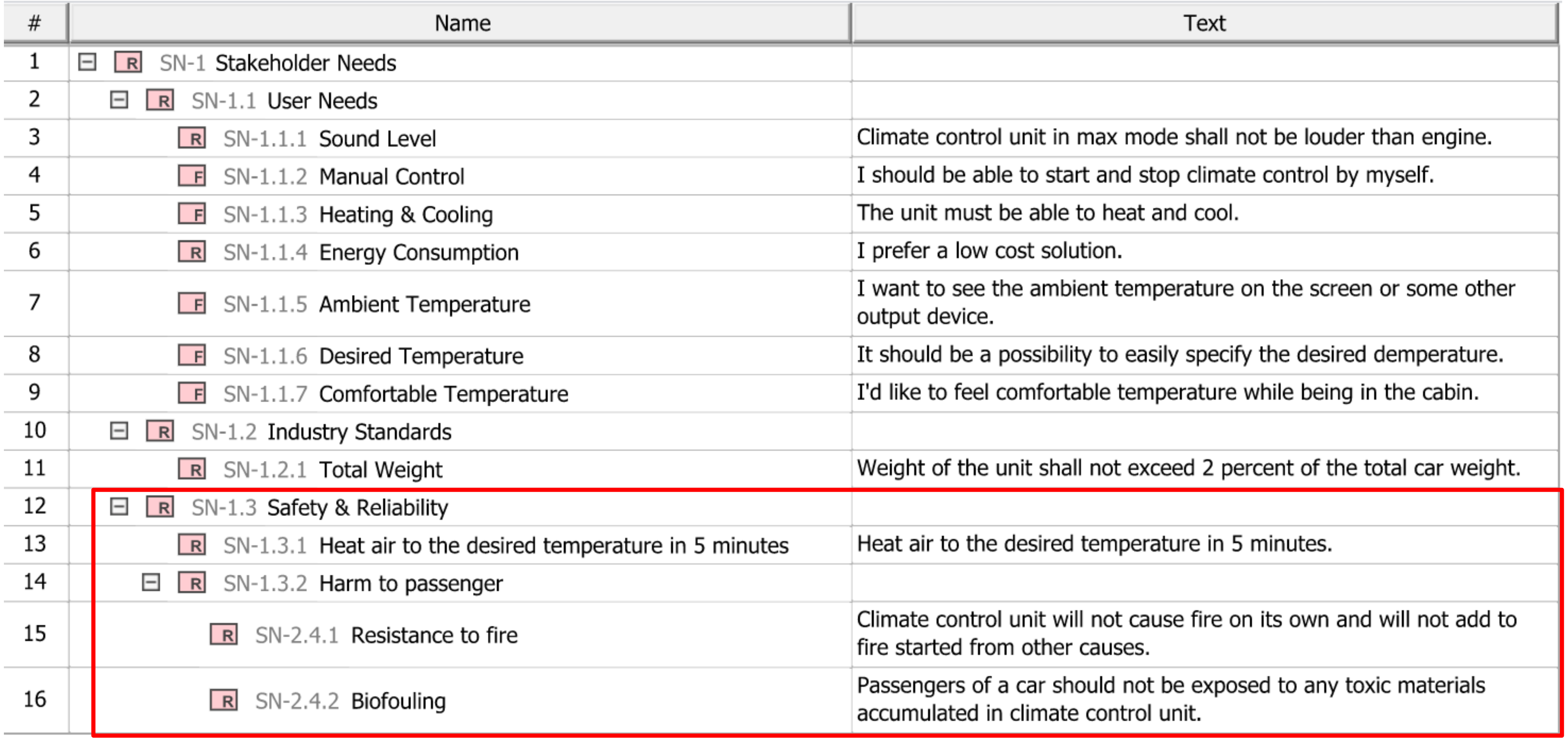
# Safety and reliability analysis relation to system model


Systems model
Modify architecture and design
Traceability of reliability analysis to architecture and design elements
Product
Safety analysis
Reliability analysis
Identify Potential Hazards and Initiating Causes



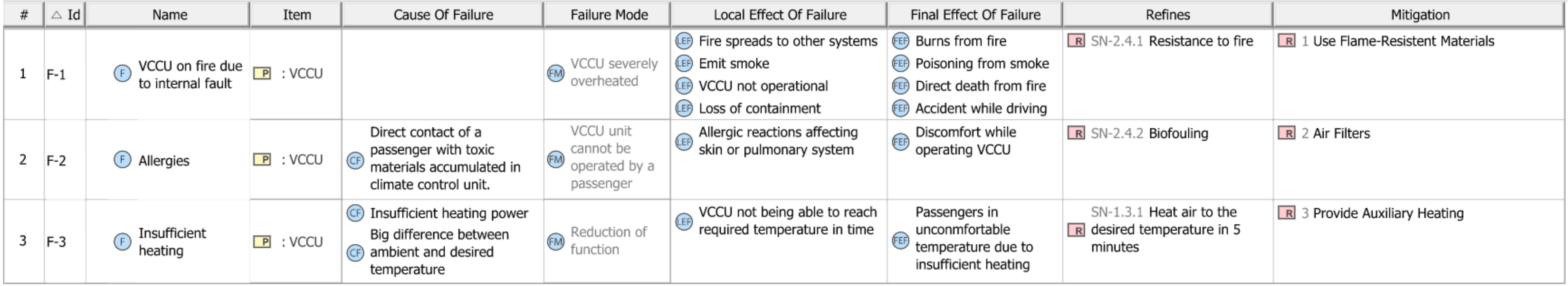
| | | PILLAR | | | | |
|---|---|---|---|---|---|---|
| | | REQUIREMENTS | STRUCTURE | BEHAVIOR | PARAMETERS | SAFETY & RELIABILITY |
| DOMAIN | PROBLEM (BLACK BOX) | Stakeholder Needs<br>Stakeholder Needs | System Context<br>Vehicle In Use<br>BB Functions To Context | Use Cases<br>Use Cases of Vehicle In Use SC<br>Provide Comfortable Temperature | Measures of Effectiveness<br>Measures of Effectiveness | Component and Functional FMEA |
| | PROBLEM (WHITE BOX) | Refine Stakeholder Needs<br>Refined Stakeholder Needs | Logical Subsystems Communication<br>VCCU Interfaces<br>VCCU Logical Subsystems<br>WB Functions To Logical Architecture | Functional Analysis<br>Reach Desired Temperature | MoEs for Subsystems<br>MoEs for Subsystems | Component and Functional FMEA |
| | SOLUTION | System Requirements<br>System Requireme nts<br>HLSA to System Requirements | System Structure<br>High-Level Solution Architecture<br>VCCS Configuration Structure | System Behavior<br>Subsystems Communication in VCCS | System Parameters<br>Total Energy Consumption of VCCS | Solution FMEA |
| | | Subsystem Requirements<br>Subsystem Requirements<br>Subsystem SA to Subsystem Requirements | Subsystem Structure<br>CP System Structure<br>Cooling System Logical Components | Subsystem Behavior<br>CP System States<br>Cooling System States | Subsystem Parameters<br>Subsystem Parameters for Total Energy Consumption for Cooling<br>Total Energy Consumption for Cooling | |
| | | Component Requirements | Component Structure | Component Behavior | Component Parameters | Component FMEA |
| | IMPLEMENTATION | Physical Requirements | Software, Electrical, Electronical, Mechanical, Fluidic | | | |

# S&R stakeholder requirements

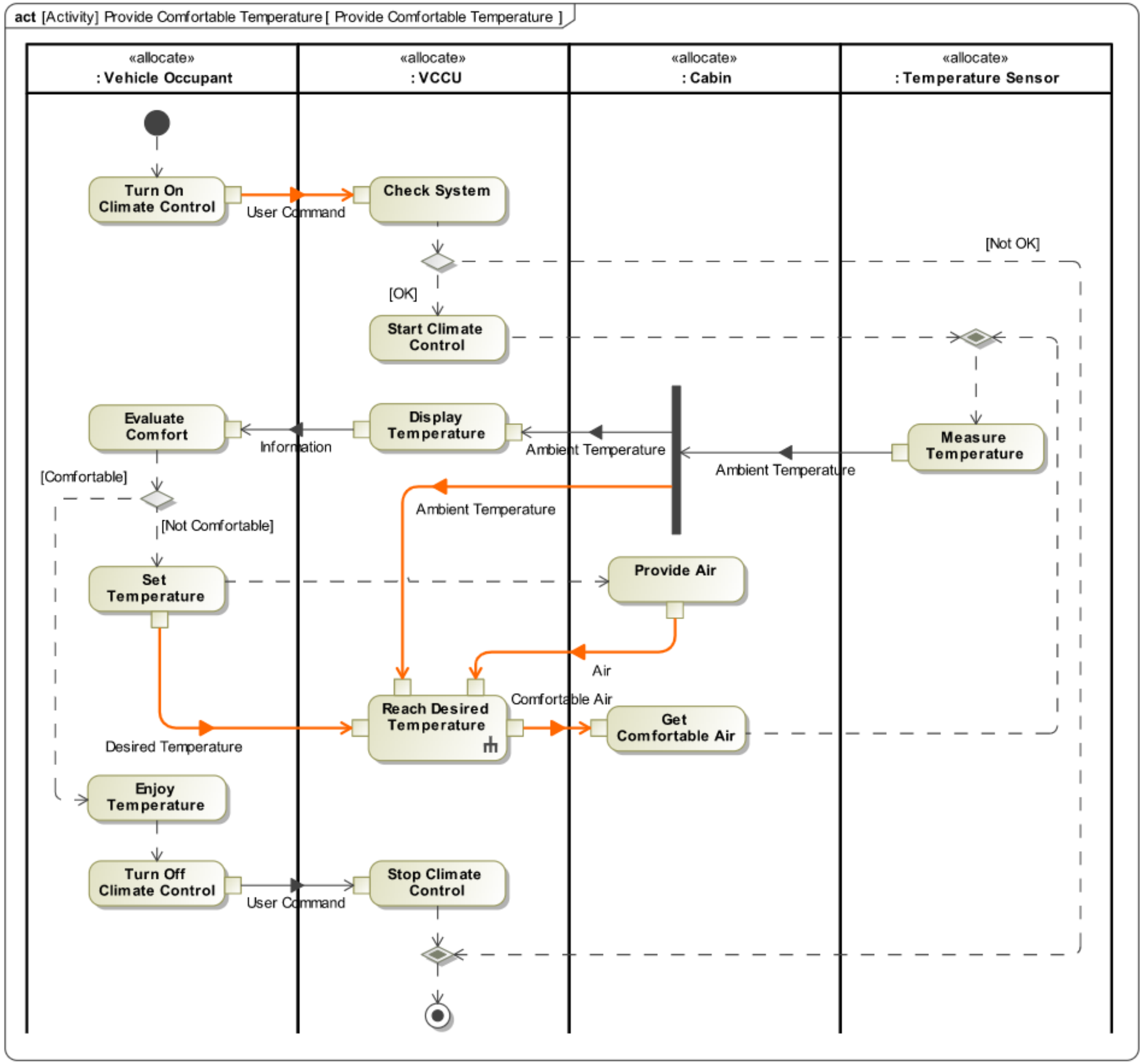


| # | Name | Text |
|---|---|---|
| 1 | ⊟ R SN-1 Stakeholder Needs | |
| 2 | ⊟ R SN-1.1 User Needs | |
| 3 | R SN-1.1.1 Sound Level | Climate control unit in max mode shall not be louder than engine. |
| 4 | F SN-1.1.2 Manual Control | I should be able to start and stop climate control by myself. |
| 5 | F SN-1.1.3 Heating & Cooling | The unit must be able to heat and cool. |
| 6 | R SN-1.1.4 Energy Consumption | I prefer a low cost solution. |
| 7 | F SN-1.1.5 Ambient Temperature | I want to see the ambient temperature on the screen or some other output device. |
| 8 | F SN-1.1.6 Desired Temperature | It should be a possibility to easily specify the desired demperature. |
| 9 | F SN-1.1.7 Comfortable Temperature | I'd like to feel comfortable temperature while being in the cabin. |
| 10 | ⊟ R SN-1.2 Industry Standards | |
| 11 | R SN-1.2.1 Total Weight | Weight of the unit shall not exceed 2 percent of the total car weight. |
| 12 | ⊟ R SN-1.3 Safety & Reliability | |
| 13 | R SN-1.3.1 Heat air to the desired temperature in 5 minutes | Heat air to the desired temperature in 5 minutes. |
| 14 | ⊟ R SN-1.3.2 Harm to passenger | |
| 15 | R SN-2.4.1 Resistance to fire | Climate control unit will not cause fire on its own and will not add to fire started from other causes. |
| 16 | R SN-2.4.2 Biofouling | Passengers of a car should not be exposed to any toxic materials accumulated in climate control unit. |

# Component FMEA in Black Box view

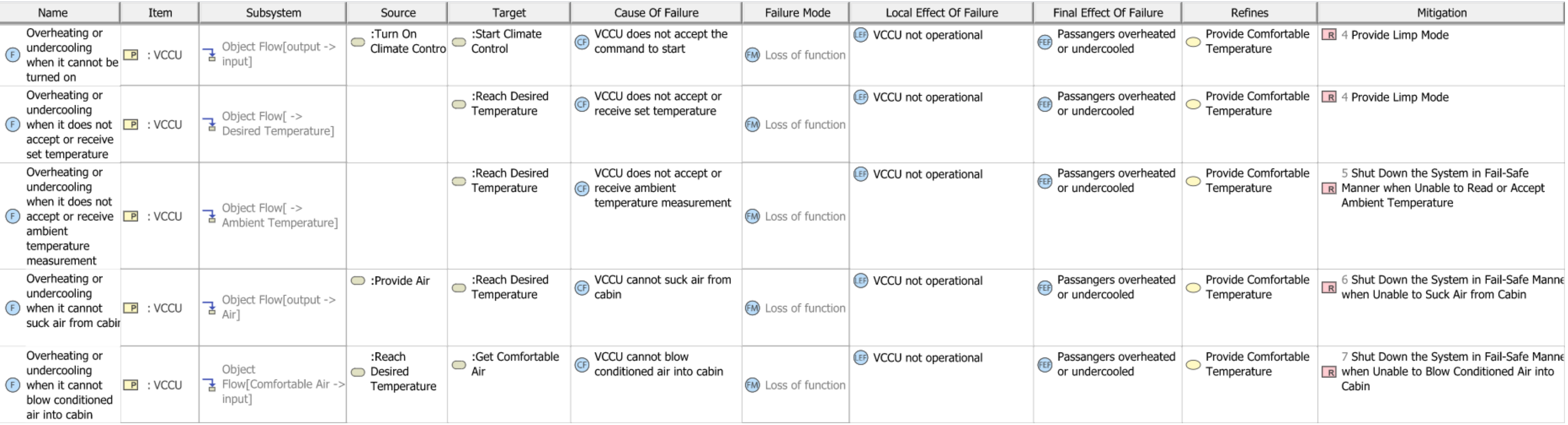


| # | Id | Name | Item | Cause Of Failure | Failure Mode | Local Effect Of Failure | Final Effect Of Failure | Refines | Mitigation |
|---|---|---|---|---|---|---|---|---|---|
| 1 | F-1 | VCCU on fire due to internal fault | : VCCU |  | VCCU severely overheated | Fire spreads to other systems<br>Emit smoke<br>VCCU not operational<br>Loss of containment | Burns from fire<br>Poisoning from smoke<br>Direct death from fire<br>Accident while driving | SN-2.4.1 Resistance to fire | 1 Use Flame-Resistent Materials |
| 2 | F-2 | Allergies | : VCCU | Direct contact of a passenger with toxic materials accumulated in climate control unit. | VCCU unit cannot be operated by a passenger | Allergic reactions affecting skin or pulmonary system | Discomfort while operating VCCU | SN-2.4.2 Biofouling | 2 Air Filters |
| 3 | F-3 | Insufficient heating | : VCCU | Insufficient heating power<br>Big difference between ambient and desired temperature | Reduction of function | VCCU not being able to reach required temperature in time | Passengers in unconmfortable temperature due to insufficient heating | SN-1.3.1 Heat air to the desired temperature in 5 minutes | 3 Provide Auxiliary Heating |

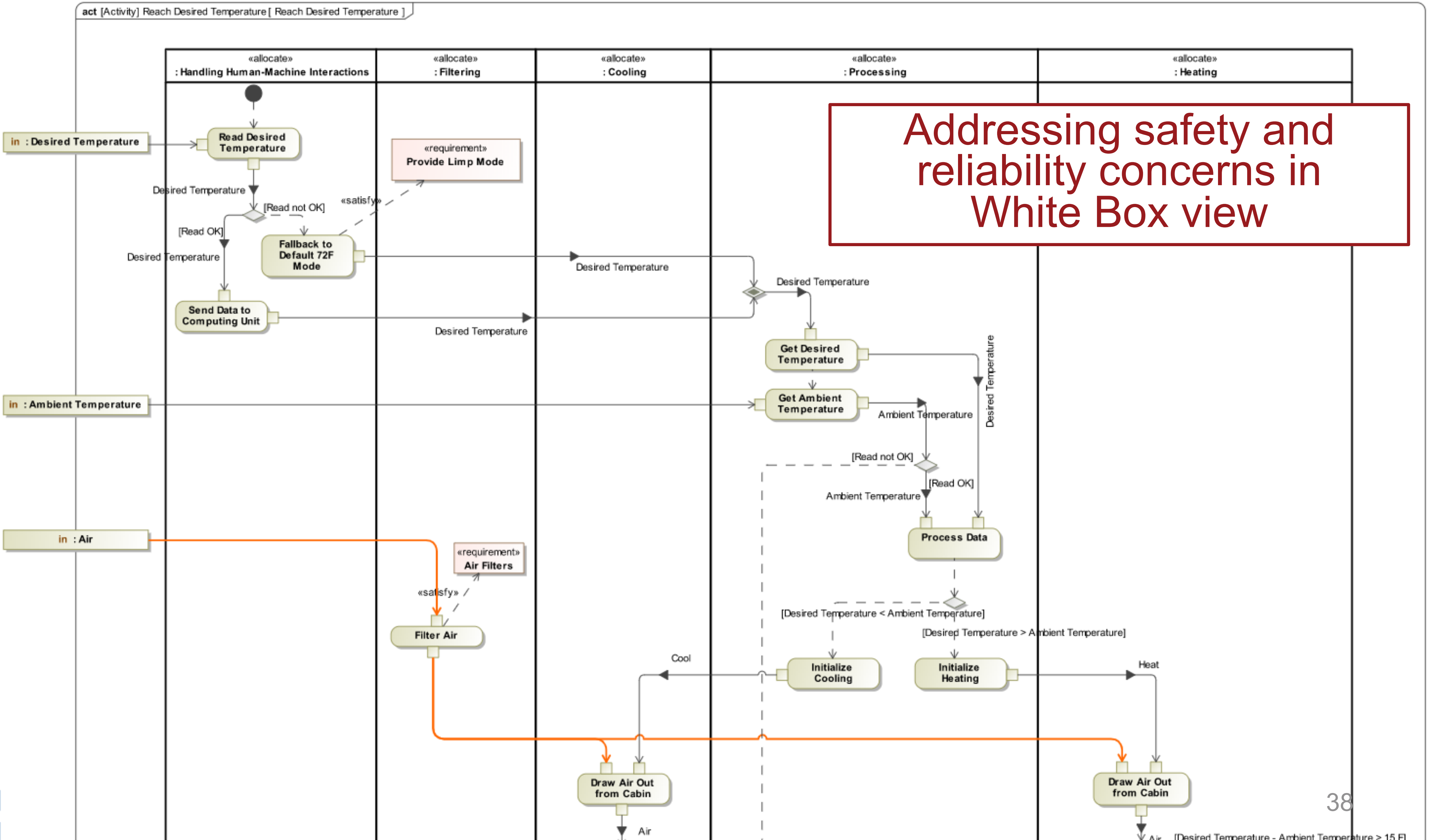


# Functional FMEA in Black Box view

# Functional FMEA in Black Box view

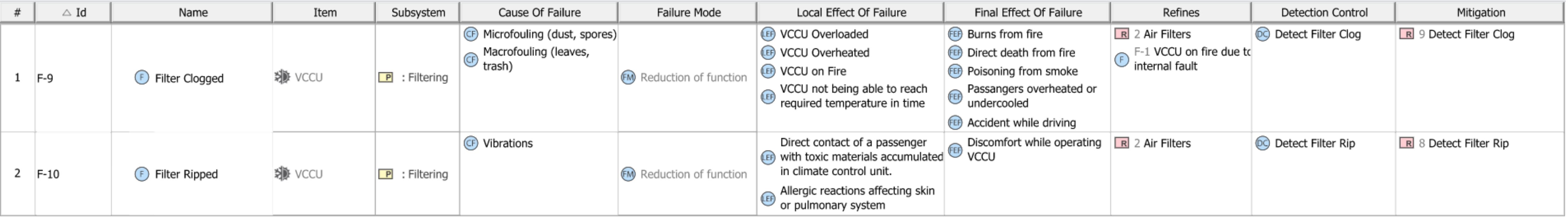


| Name | Item | Subsystem | Source | Target | Cause Of Failure | Failure Mode | Local Effect Of Failure | Final Effect Of Failure | Refines | Mitigation |
|---|---|---|---|---|---|---|---|---|---|---|
| Overheating or undercooling when it cannot be turned on | : VCCU | Object Flow[output -> input] | :Turn On Climate Contro | :Start Climate Control | VCCU does not accept the command to start | Loss of function | VCCU not operational | Passangers overheated or undercooled | Provide Comfortable Temperature | 4 Provide Limp Mode |
| Overheating or undercooling when it does not accept or receive set temperature | : VCCU | Object Flow[ -> Desired Temperature] |  | :Reach Desired Temperature | VCCU does not accept or receive set temperature | Loss of function | VCCU not operational | Passangers overheated or undercooled | Provide Comfortable Temperature | 4 Provide Limp Mode |
| Overheating or undercooling when it does not accept or receive ambient temperature measurement | : VCCU | Object Flow[ -> Ambient Temperature] |  | :Reach Desired Temperature | VCCU does not accept or receive ambient temperature measurement | Loss of function | VCCU not operational | Passangers overheated or undercooled | Provide Comfortable Temperature | 5 Shut Down the System in Fail-Safe Manner when Unable to Read or Accept Ambient Temperature |
| Overheating or undercooling when it cannot suck air from cabir | : VCCU | Object Flow[output -> Air] | :Provide Air | :Reach Desired Temperature | VCCU cannot suck air from cabin | Loss of function | VCCU not operational | Passangers overheated or undercooled | Provide Comfortable Temperature | 6 Shut Down the System in Fail-Safe Manne when Unable to Suck Air from Cabin |
| Overheating or undercooling when it cannot blow conditioned air into cabin | : VCCU | Object Flow[Comfortable Air -> input] | :Reach Desired Temperature | :Get Comfortable Air | VCCU cannot blow conditioned air into cabin | Loss of function | VCCU not operational | Passangers overheated or undercooled | Provide Comfortable Temperature | 7 Shut Down the System in Fail-Safe Manne when Unable to Blow Conditioned Air into Cabin |

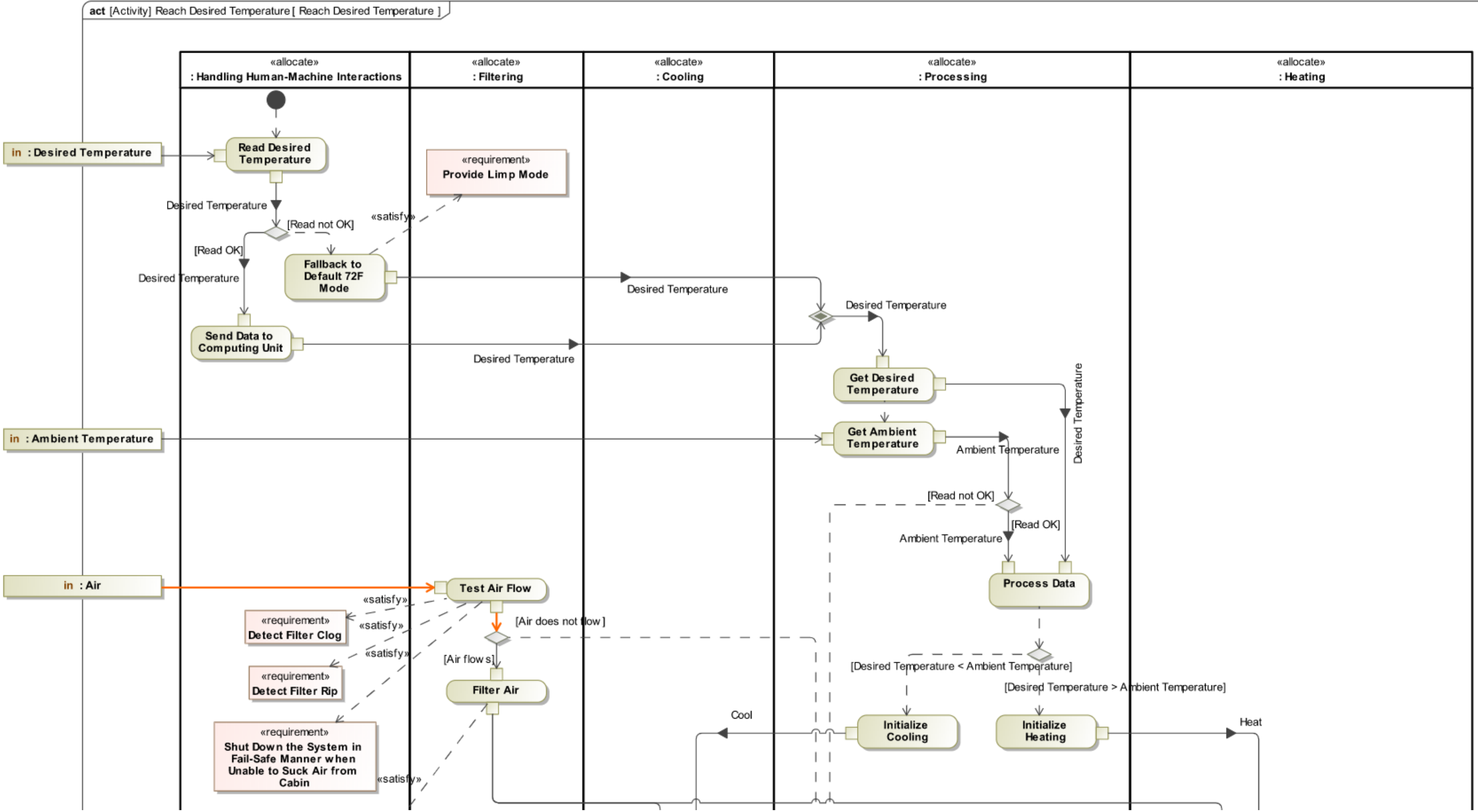


# Addressing safety and reliability concerns in White Box view

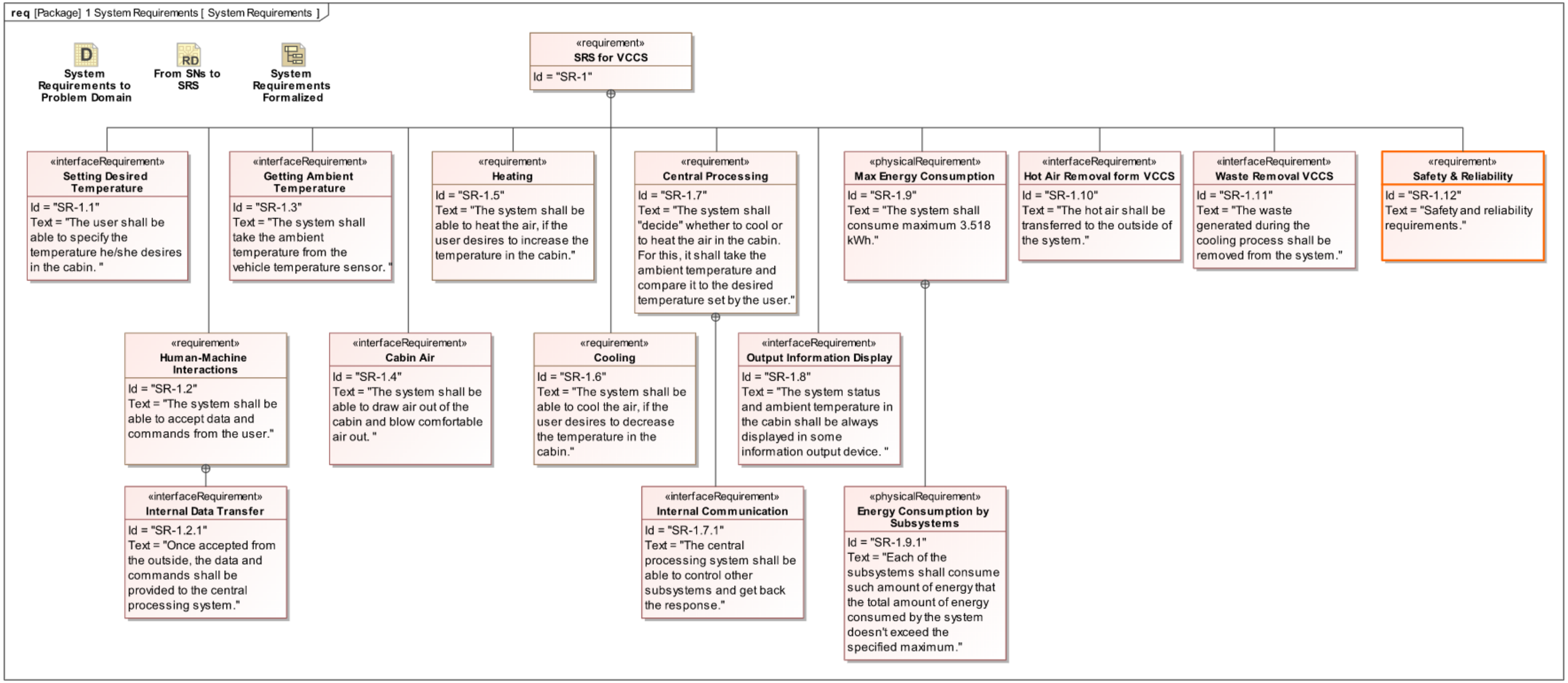
# Component FMEA at the White Box view



| # | Id | Name | Item | Subsystem | Cause Of Failure | Failure Mode | Local Effect Of Failure | Final Effect Of Failure | Refines | Detection Control | Mitigation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F-9 | Filter Clogged | VCCU | : Filtering | Microfouling (dust, spores)<br>Macrofouling (leaves, trash) | Reduction of function | VCCU Overloaded<br>VCCU Overheated<br>VCCU on Fire<br>VCCU not being able to reach required temperature in time | Burns from fire<br>Direct death from fire<br>Poisoning from smoke<br>Passangers overheated or undercooled<br>Accident while driving | 2 Air Filters<br>F-1 VCCU on fire due tc internal fault | Detect Filter Clog | 9 Detect Filter Clog |
| 2 | F-10 | Filter Ripped | VCCU | : Filtering | Vibrations | Reduction of function | Direct contact of a passenger with toxic materials accumulated in climate control unit.<br>Allergic reactions affecting skin or pulmonary system | Discomfort while operating VCCU | 2 Air Filters | Detect Filter Rip | 8 Detect Filter Rip |


act [Activity] Reach Desired Temperature [ Reach Desired Temperature ]
«allocate»
: Handling Human-Machine Interactions
«allocate»
: Filtering
«allocate»
: Cooling
«allocate»
: Processing
«allocate»
: Heating
in : Desired Temperature
Read Desired Temperature
Desired Temperature
[Read not OK]
«satisfy»
«requirement»
Provide Limp Mode
[Read OK]
Desired Temperature
Fallback to Default 72F Mode
Desired Temperature
Desired Temperature
Send Data to Computing Unit
Desired Temperature
Get Desired Temperature
Desired Temperature
in : Ambient Temperature
Get Ambient Temperature
Ambient Temperature
[Read not OK]
[Read OK]
Ambient Temperature
Process Data
in : Air
Test Air Flow
«satisfy»
«requirement»
Detect Filter Clog
«satisfy»
[Air does not flow ]
«satisfy»
[Air flow s]
«requirement»
Detect Filter Rip
Filter Air
«requirement»
Shut Down the System in Fail-Safe Manner when Unable to Suck Air from Cabin
«satisfy»
[Desired Temperature < Ambient Temperature]
[Desired Temperature > Ambient Temperature]
Cool
Initialize Cooling
Initialize Heating
Heat

# Addressing safety and reliability concerns in HLSA



req [Package] 1 System Requirements [ System Requirements ]

D
System Requirements to Problem Domain
RD
From SNs to SRS
System Requirements Formalized

«requirement»
**SRS for VCCS**
Id = "SR-1"

«interfaceRequirement»
**Setting Desired Temperature**
Id = "SR-1.1"
Text = "The user shall be able to specify the temperature he/she desires in the cabin. "

«interfaceRequirement»
**Getting Ambient Temperature**
Id = "SR-1.3"
Text = "The system shall take the ambient temperature from the vehicle temperature sensor. "

«requirement»
**Heating**
Id = "SR-1.5"
Text = "The system shall be able to heat the air, if the user desires to increase the temperature in the cabin."

«requirement»
**Central Processing**
Id = "SR-1.7"
Text = "The system shall "decide" whether to cool or to heat the air in the cabin. For this, it shall take the ambient temperature and compare it to the desired temperature set by the user."

«physicalRequirement»
**Max Energy Consumption**
Id = "SR-1.9"
Text = "The system shall consume maximum 3.518 kWh."

«interfaceRequirement»
**Hot Air Removal form VCCS**
Id = "SR-1.10"
Text = "The hot air shall be transferred to the outside of the system."

«interfaceRequirement»
**Waste Removal VCCS**
Id = "SR-1.11"
Text = "The waste generated during the cooling process shall be removed from the system."

«requirement»
**Safety & Reliability**
Id = "SR-1.12"
Text = "Safety and reliability requirements."

«requirement»
**Human-Machine Interactions**
Id = "SR-1.2"
Text = "The system shall be able to accept data and commands from the user."

«interfaceRequirement»
**Cabin Air**
Id = "SR-1.4"
Text = "The system shall be able to draw air out of the cabin and blow comfortable air out. "

«requirement»
**Cooling**
Id = "SR-1.6"
Text = "The system shall be able to cool the air, if the user desires to decrease the temperature in the cabin."

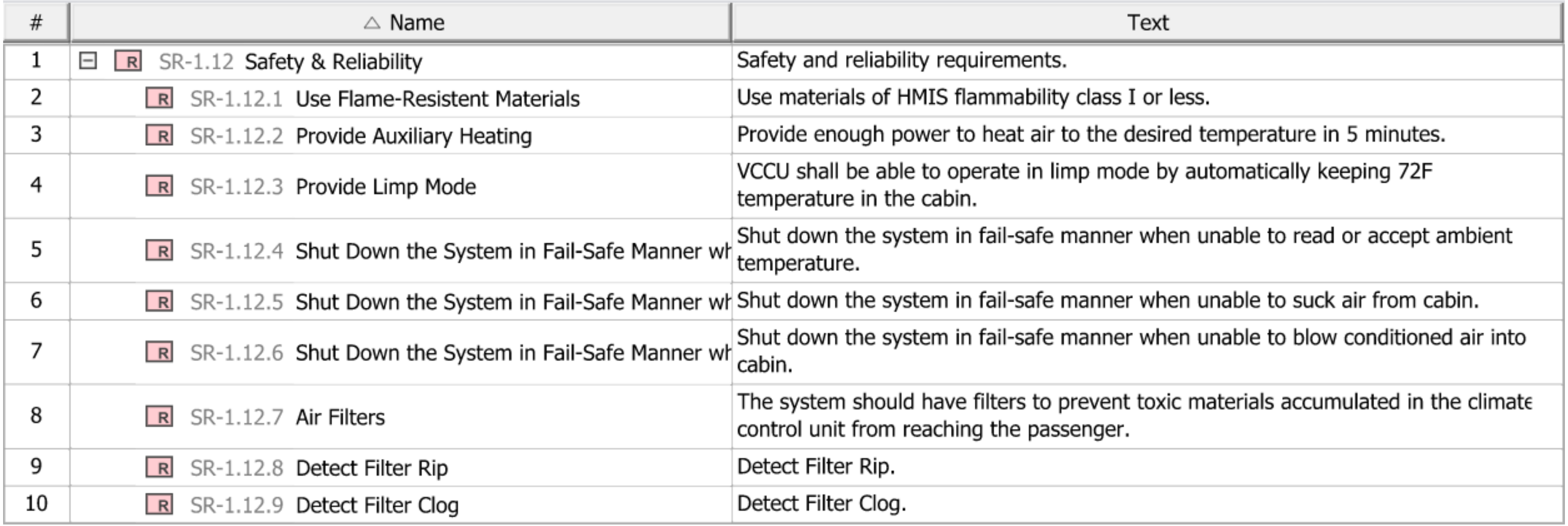
«interfaceRequirement»
**Output Information Display**
Id = "SR-1.8"
Text = "The system status and ambient temperature in the cabin shall be always displayed in some information output device. "

«interfaceRequirement»
**Internal Data Transfer**
Id = "SR-1.2.1"
Text = "Once accepted from the outside, the data and commands shall be provided to the central processing system."

«interfaceRequirement»
**Internal Communication**
Id = "SR-1.7.1"
Text = "The central processing system shall be able to control other subsystems and get back the response."

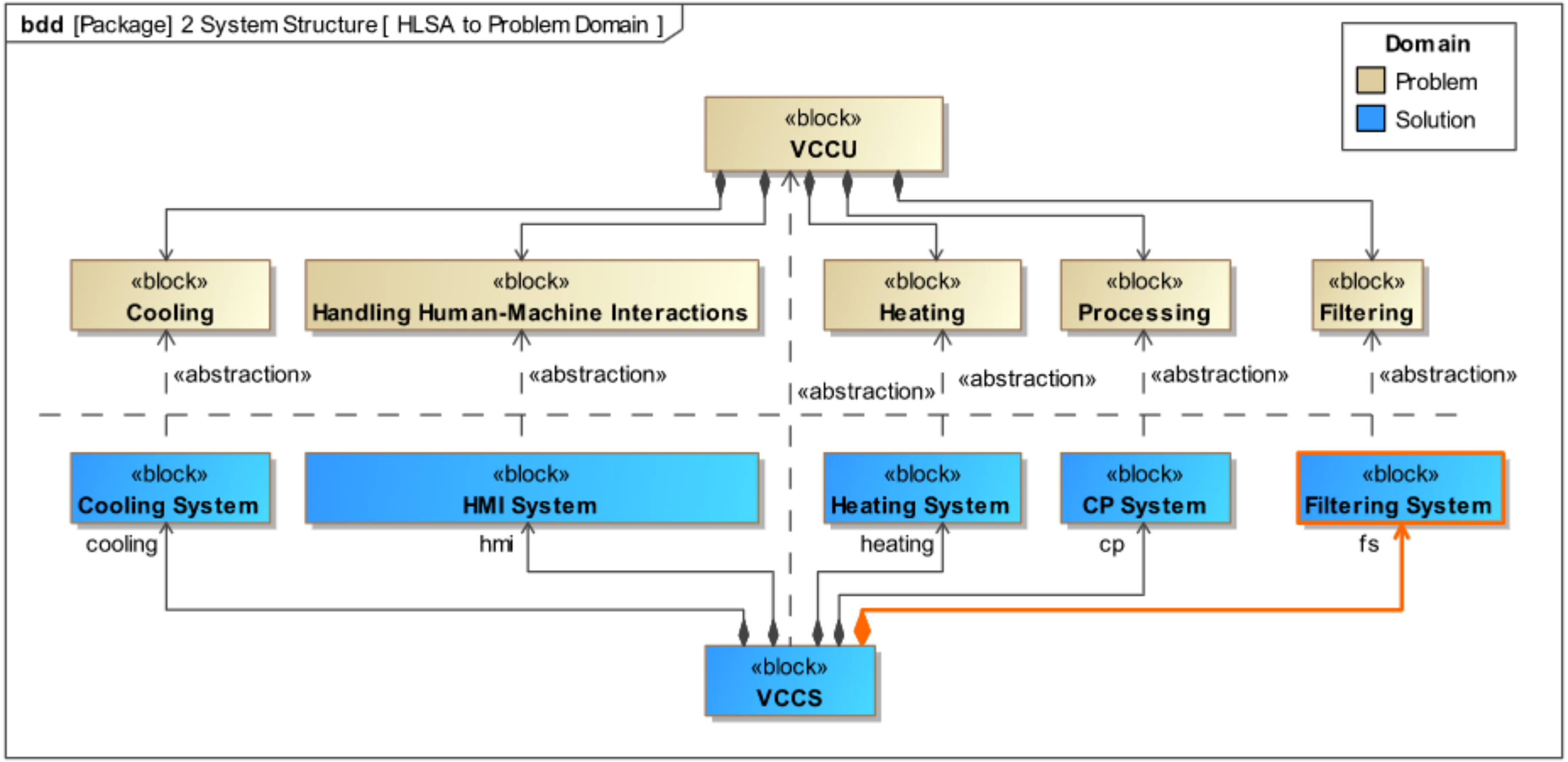
«physicalRequirement»
**Energy Consumption by Subsystems**
Id = "SR-1.9.1"
Text = "Each of the subsystems shall consume such amount of energy that the total amount of energy consumed by the system doesn't exceed the specified maximum."

# Addressing safety and reliability concerns in HLSA

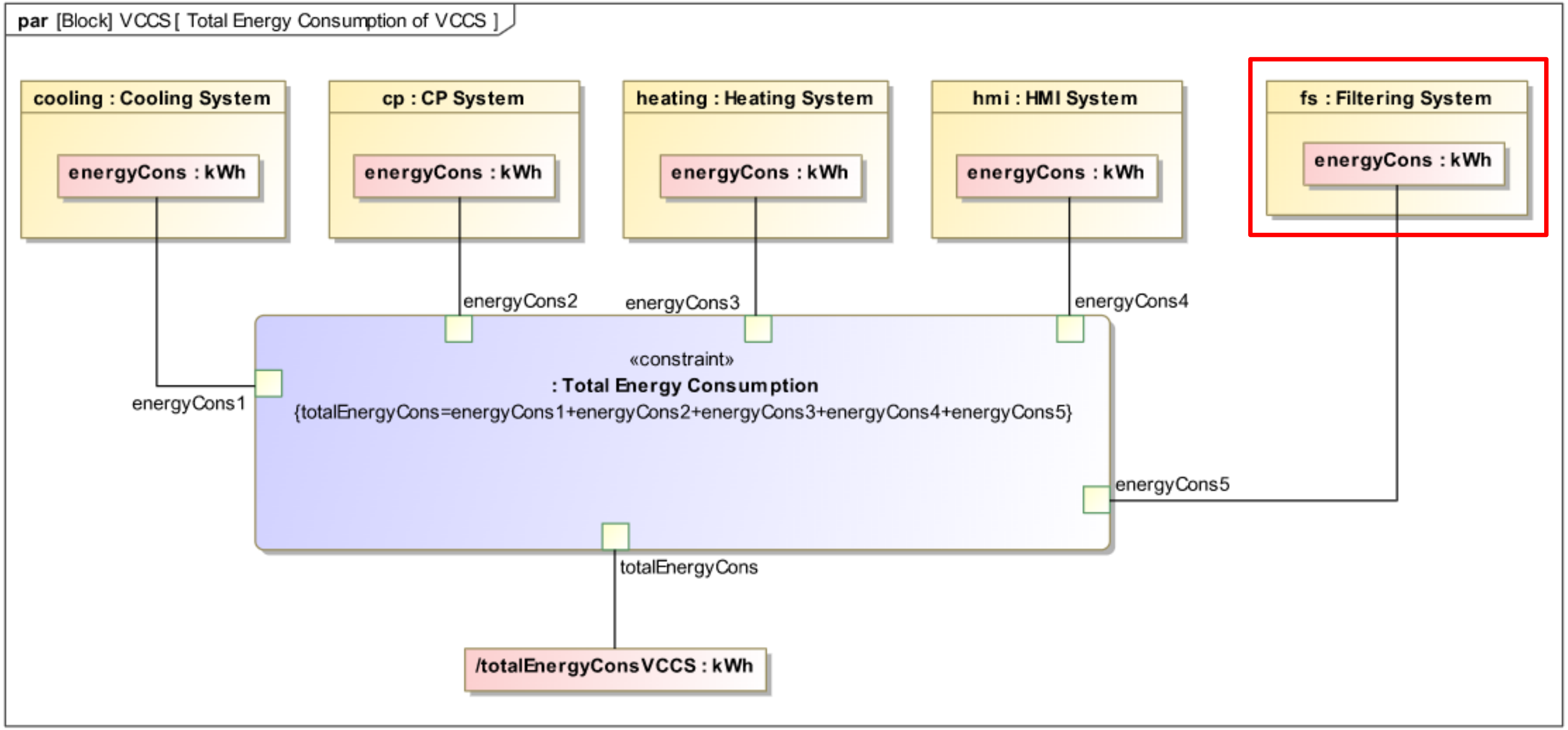


| # | △ Name | Text |
|---|---|---|
| 1 | ⊟ R SR-1.12 Safety & Reliability | Safety and reliability requirements. |
| 2 | R SR-1.12.1 Use Flame-Resistent Materials | Use materials of HMIS flammability class I or less. |
| 3 | R SR-1.12.2 Provide Auxiliary Heating | Provide enough power to heat air to the desired temperature in 5 minutes. |
| 4 | R SR-1.12.3 Provide Limp Mode | VCCU shall be able to operate in limp mode by automatically keeping 72F temperature in the cabin. |
| 5 | R SR-1.12.4 Shut Down the System in Fail-Safe Manner wh | Shut down the system in fail-safe manner when unable to read or accept ambient temperature. |
| 6 | R SR-1.12.5 Shut Down the System in Fail-Safe Manner wh | Shut down the system in fail-safe manner when unable to suck air from cabin. |
| 7 | R SR-1.12.6 Shut Down the System in Fail-Safe Manner wh | Shut down the system in fail-safe manner when unable to blow conditioned air into cabin. |
| 8 | R SR-1.12.7 Air Filters | The system should have filters to prevent toxic materials accumulated in the climate control unit from reaching the passenger. |
| 9 | R SR-1.12.8 Detect Filter Rip | Detect Filter Rip. |
| 10 | R SR-1.12.9 Detect Filter Clog | Detect Filter Clog. |

# Addressing safety and reliability concerns in HLSA



# Addressing safety and reliability concerns in HLSA



# Bridging the gap between MBSE and MBD

# Bridging the gap btw. MBSE and MBD

| Domains | | Pillars | | | | |
|---|---|---|---|---|---|---|
| | | Requirements | Behavior | Structure | Parameters | Safety |
| Problem | Black Box | Stakeholder Needs | Use Cases | System Context | Measures of Effectiveness | Preliminary Risk Analysis |
| | White Box | | Functional Analysis | Logical Subsystems Communication | MoEs of Subsystems | |
| Solution | | System Requirements | System Behavior | System Structure<br>Geographic Zones | System Parameters | Functional FMEA/FTA |
| | | Subsystem… | … | … | … | |
| | | Component… | Logical Component Design | | | |
| Implementation | | Physical Requirements | Physical Component Design<br>Mechanical, Electrical, Fluid, Electronics, Software… | | | Design FMEA/FTA |

# Bridging the gap btw. MBSE and MBD (2)

- **System Zones.** One of the physical aspects captured in the solution domain is the organization of a system into physical zones.
- **Discipline-Specific Design Including Safety.** The detail design of the (selected) solution is carried out outside of SysML. It is, however, necessary to capture traceability between system architecture and geometrical architecture, fluid, electrical electronic, and Software architectures.
- **Implementation Domain.** Traceability between the Solution and Implementation Domains is necessary to be discussed and is a core component of the digital continuity

# Conclusions

- The study of existing MBSE methods and feedback collected from industry proved once more that the basis we developed previously is still the most up-to-date methodology, fully aligned with SysML.
- In accordance with this conclusion we identified areas to update to better support an evolving MBSE market and bridge the gap between MBSE and Model-based Design (MBD).
  - Major expansion areas, such as the Safety pillar and Implementation domain, were defined.
  - Some slight updates for stakeholder requirements, system structure, and system behavior views have been developed.
- The ongoing work of improving and extending MagicGrid is far from being complete. There are many different areas to be addressed in MagicGrid to continue its evolution, including trade-off analysis, security, behavioral simulation, Product Line Engineering (PLE), system model to physical models integration, etc.

30th Annual **INCOSE**
international symposium

Virtual Event
July 20 - 22, 2020

www.incose.org/symp2020