MATLAB EXPO 2021

Allocation Workflows for Architectures and Requirements

Becky Petteys

MathWorks®

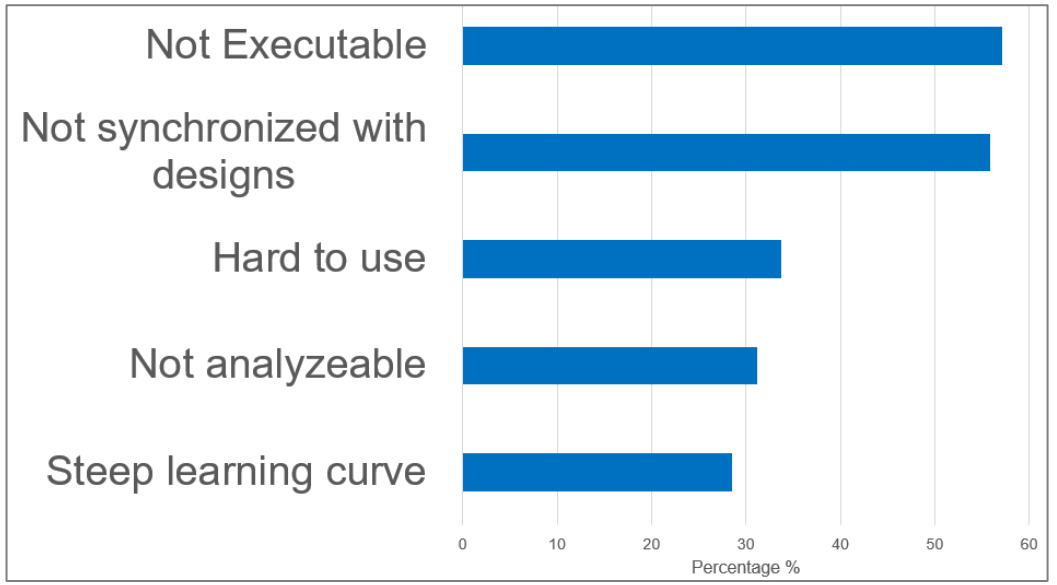
Poll Time!

- What challenges do you face today in your system engineering process?
  a) Multiple tools are needed
  b) Tools are hard to learn
  c) Responding to changes
  d) Tracing between artifacts
  e) Lack of execution and analysis
  f) Synchronization with design

What we've heard from YOU

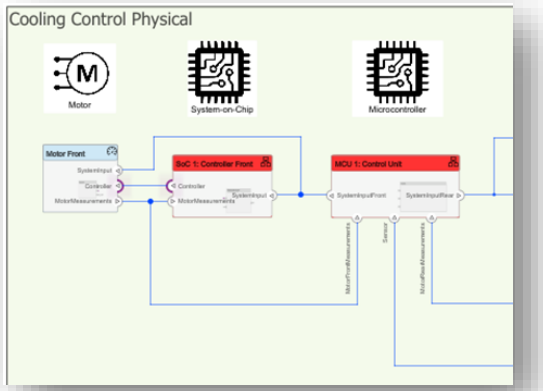
- Model Based Systems Engineering is a huge improvement over document-based methods

- Existing tools are often missing key capabilities

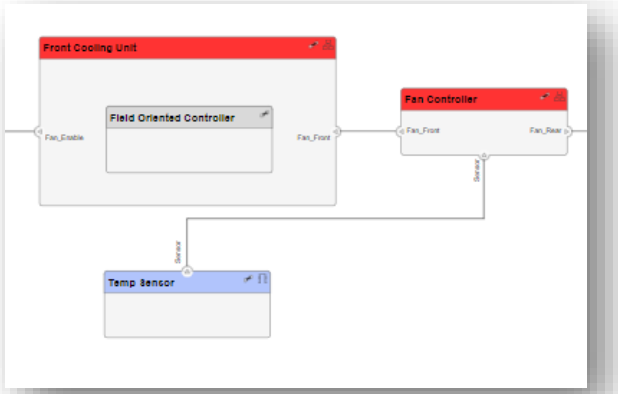


Why MBSE with MathWorks Tools?

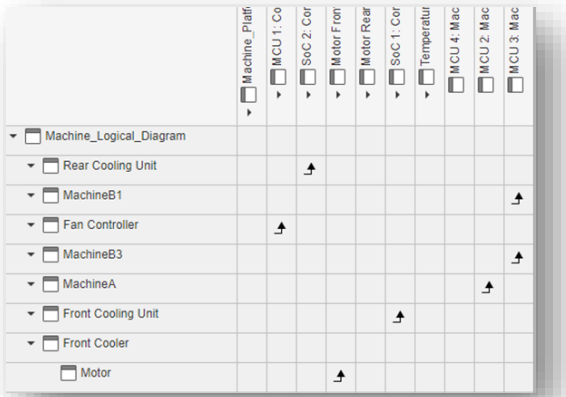
Be Intuitive



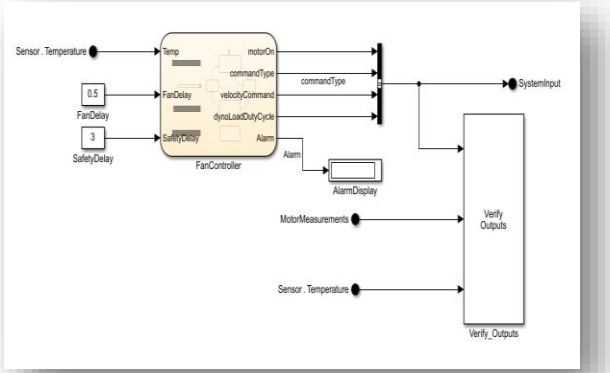
Tackle Complexity



Facilitate Traceability

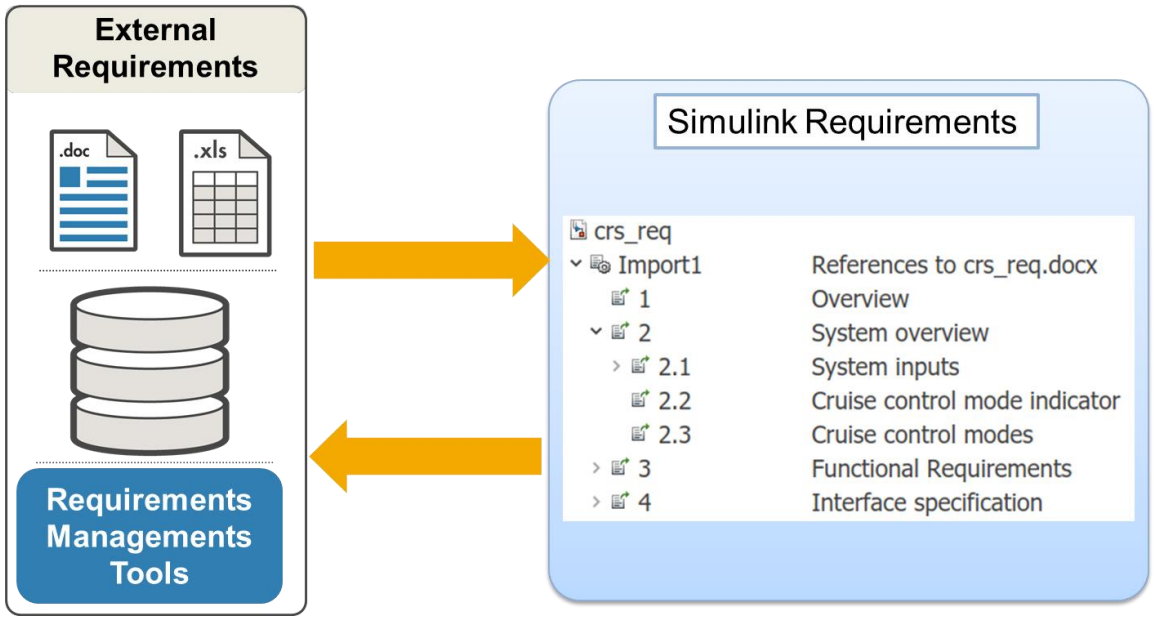


Enable Implementation



Key Takeaways

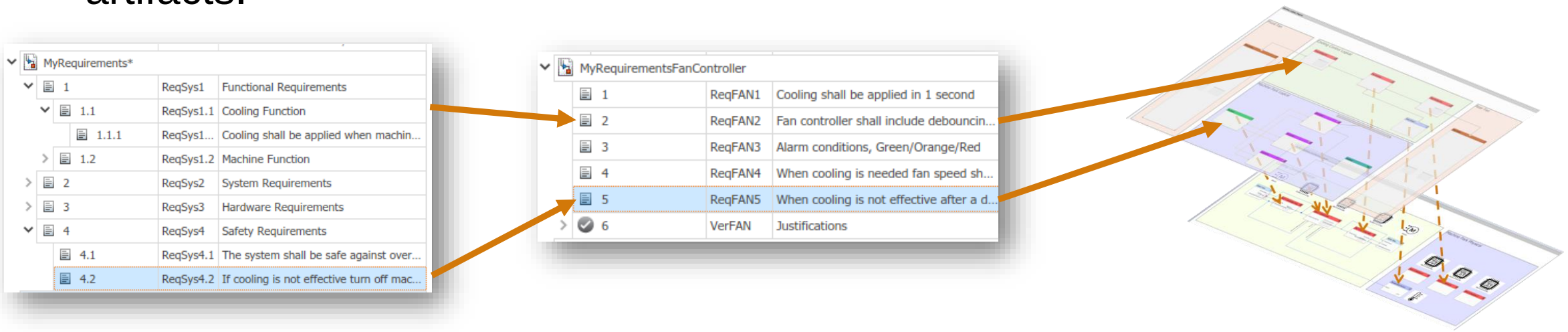
- You can import, write, and store textual requirements right **in the same environment** as your architecture and design models.



Key Takeaways

- You can import, write, and store textual requirements right **in the same environment** as your architecture and design models.

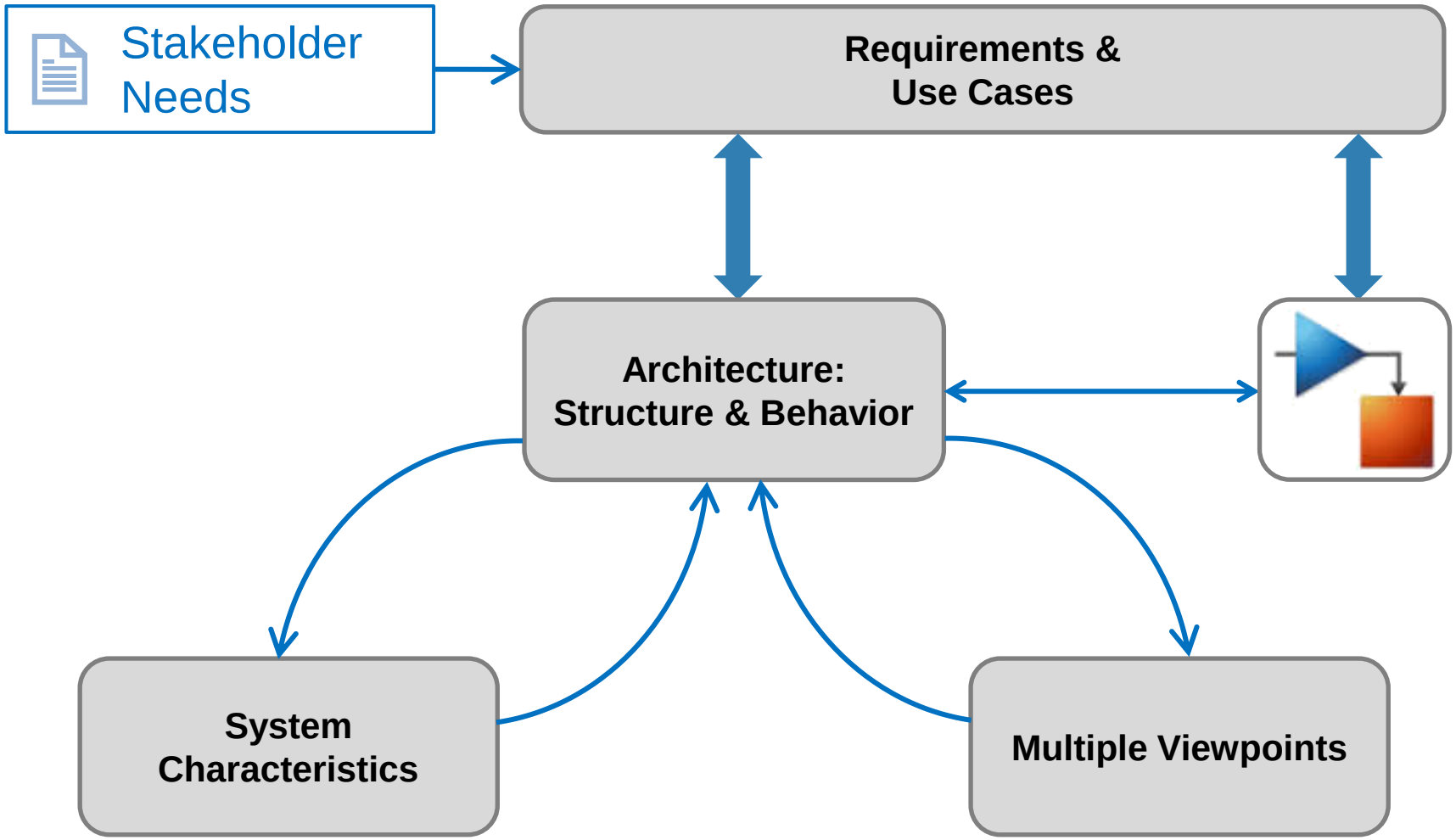
- You can **understand the impact of changes in your system** by establishing relationships among multiple requirements and architecture artifacts.

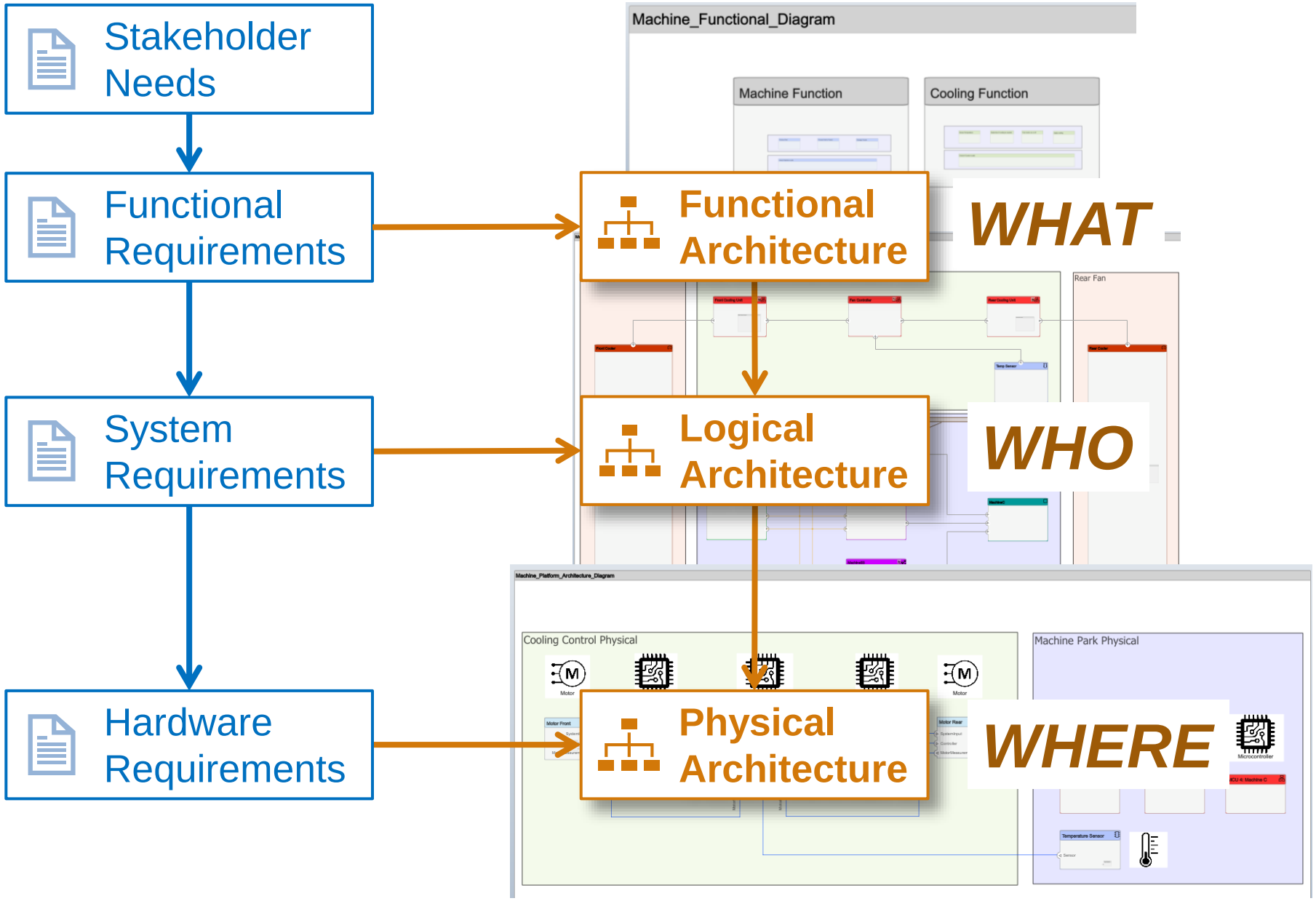


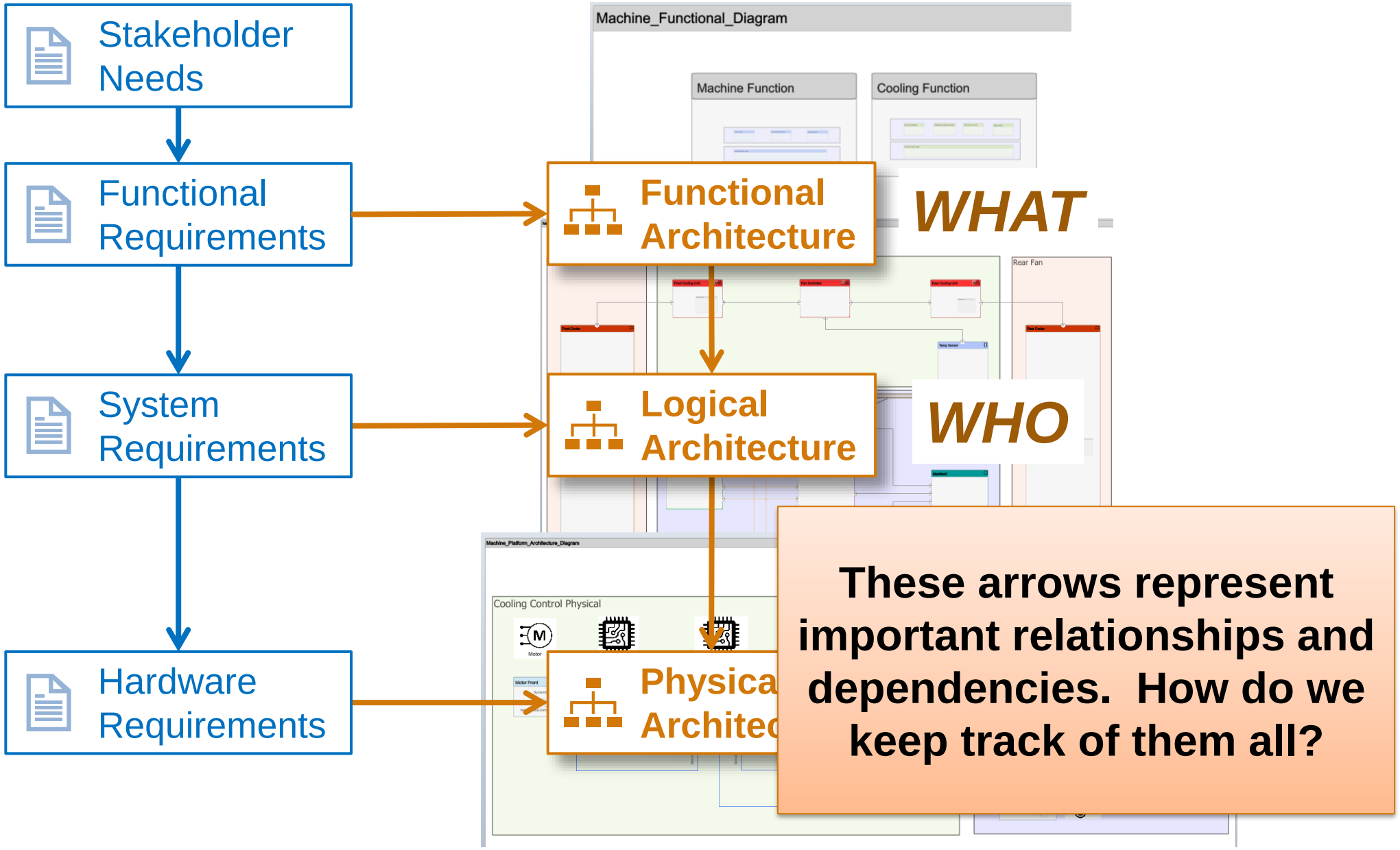
Key Takeaways

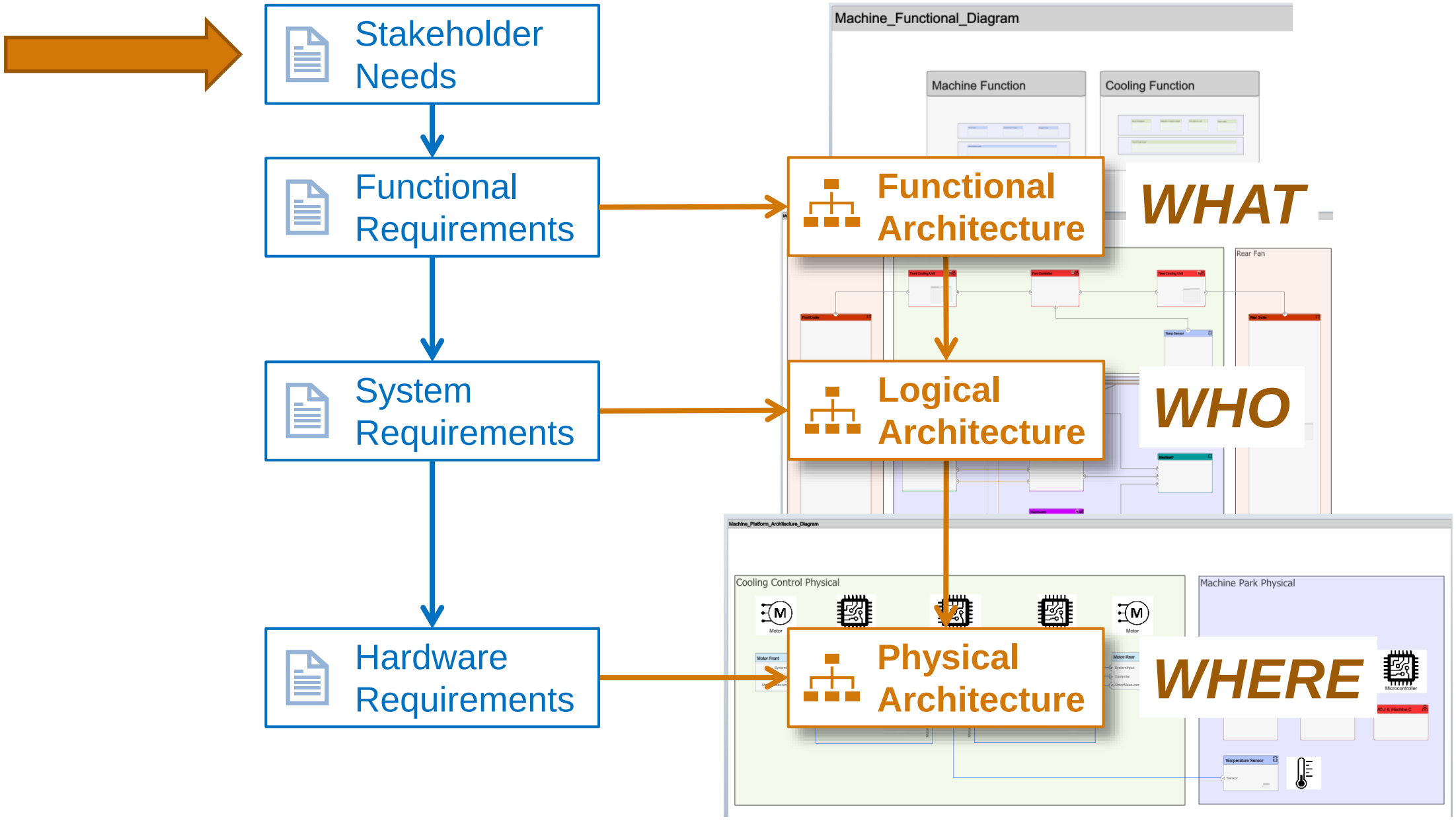
- You can import, write, and store textual requirements right **in the same environment** as your architecture and design models.
- You can **understand the impact of changes in your system** by establishing relationships among multiple requirements and architecture artifacts.
- You can **assess the completeness of your system** by visualizing those relationships.

Typical System Engineering Tasks

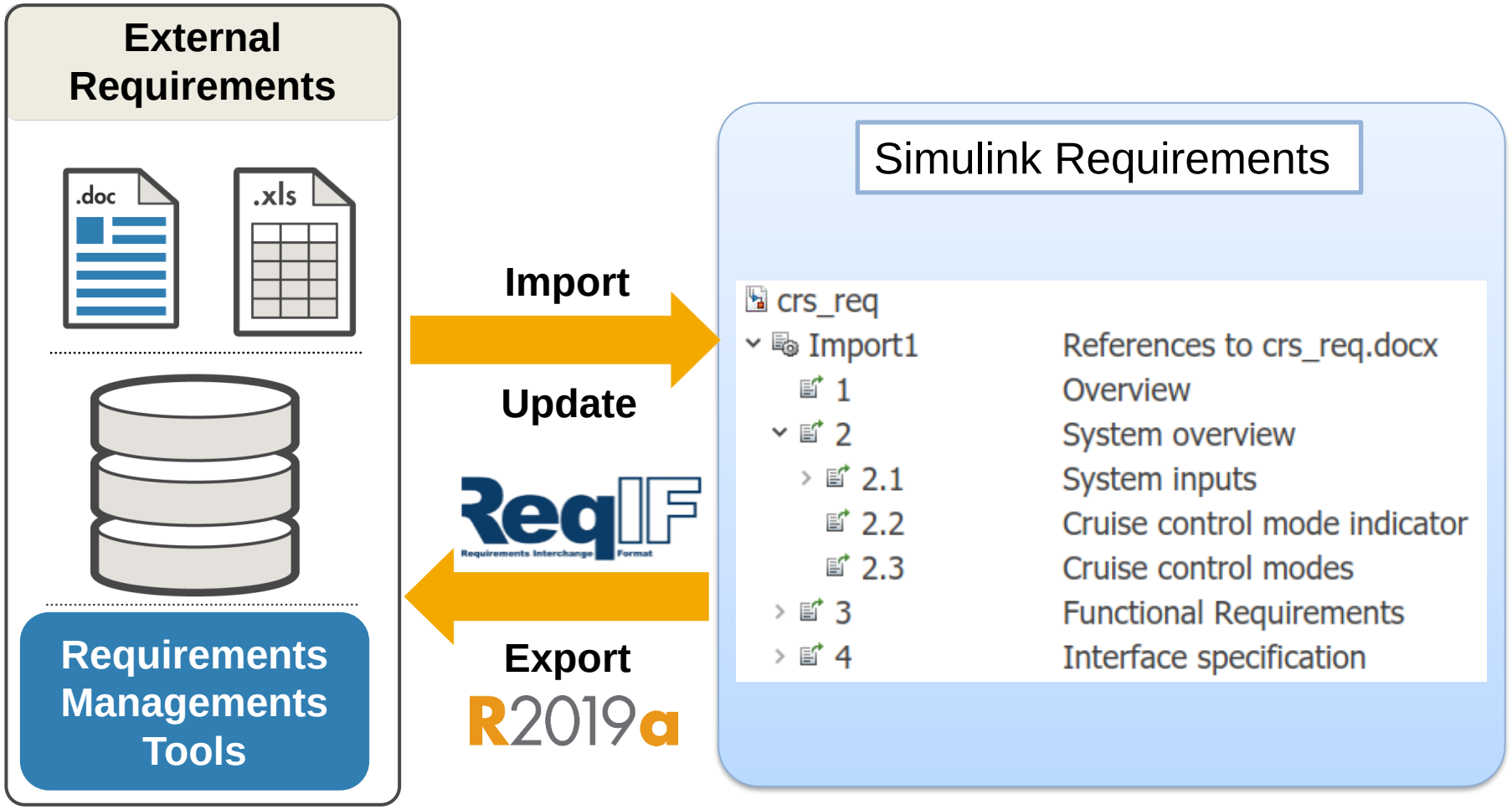



Stakeholder Needs
Functional Requirements
System Requirements
Hardware Requirements
Machine_Functional_Diagram
Machine Function
Cooling Function
Rear Fan
Functional Architecture
WHAT
Logical Architecture
WHO
Machine_Platform_Architecture_Diagram
Cooling Control Physical
Machine Park Physical
Physical Architecture
WHERE


Stakeholder Needs
Functional Requirements
System Requirements
Hardware Requirements
Machine_Functional_Diagram
Machine Function
Cooling Function
Functional Architecture
WHAT
Logical Architecture
WHO
Rear Fan
Cooling Control Physical
Physica
Archite
These arrows represent important relationships and dependencies. How do we keep track of them all?


Stakeholder Needs
Functional Requirements
System Requirements
Hardware Requirements
Functional Architecture
Logical Architecture
Physical Architecture
WHAT
WHO
WHERE
Machine_Functional_Diagram
Machine Function
Cooling Function
Rear Fan
Machine_Platform_Architecture_Diagram
Cooling Control Physical
Machine Park Physical

Exchange data with third party requirements tools



- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - DOORS Next
  - ReqIF™ standard
- Synchronize changes from source
- Edit and add further details to import R2019a
- Export ReqIF
  - Enables roundtrip with external tools R2019a

Stakeholder Needs

Functional Requirements

System Requirements

Hardware Requirements

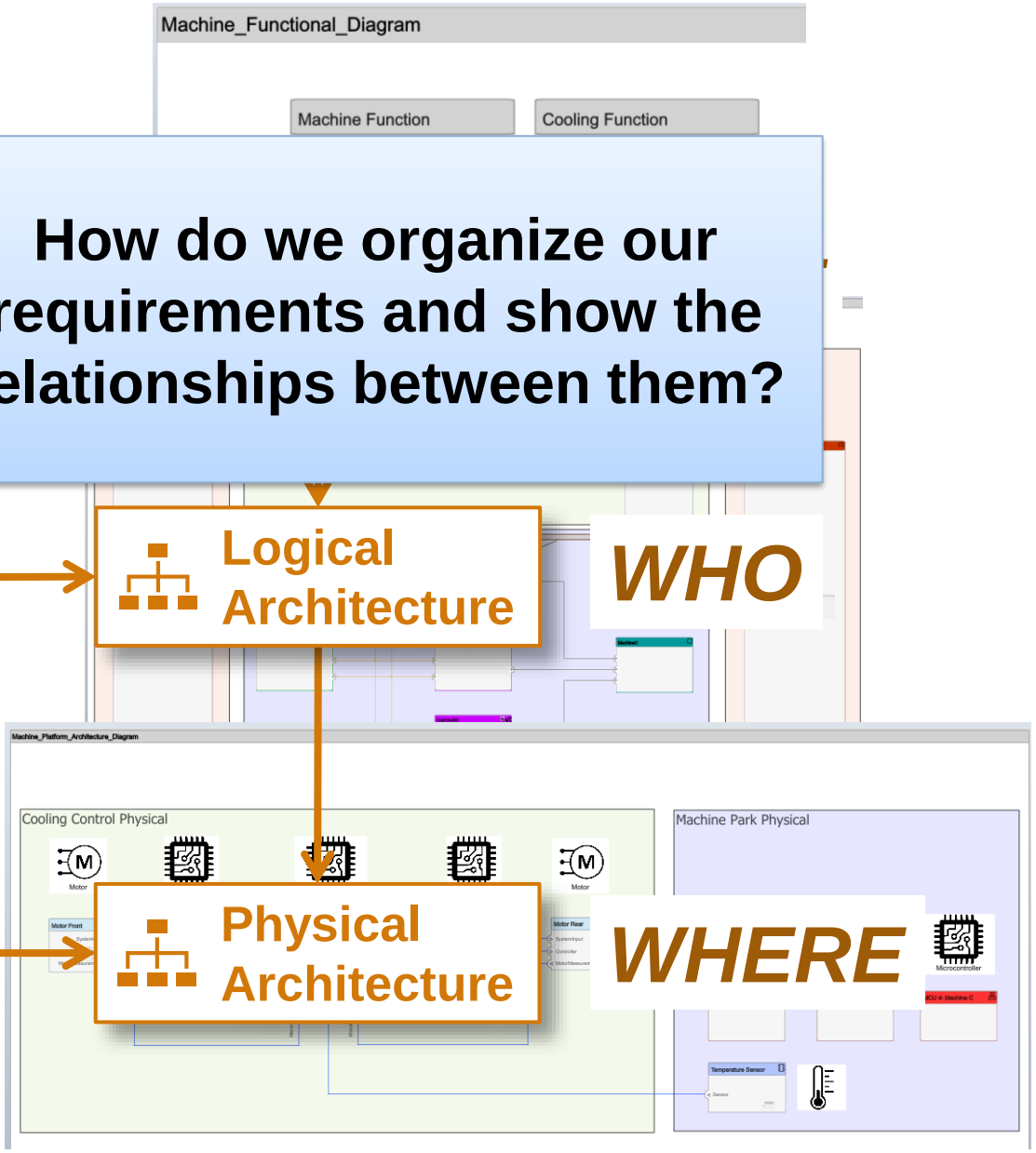
How do we organize our requirements and show the relationships between them?

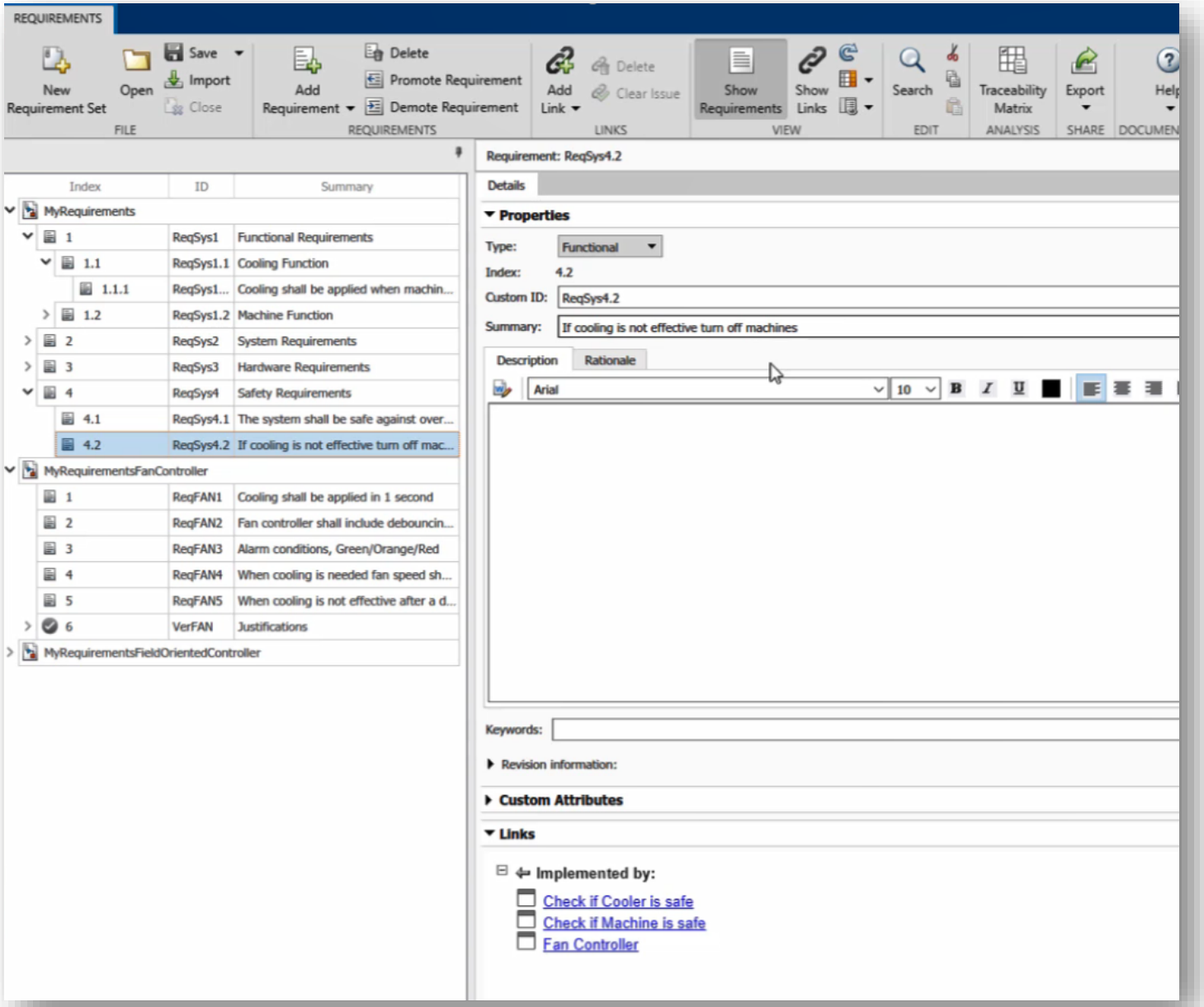
Logical Architecture

WHO

Physical Architecture

WHERE



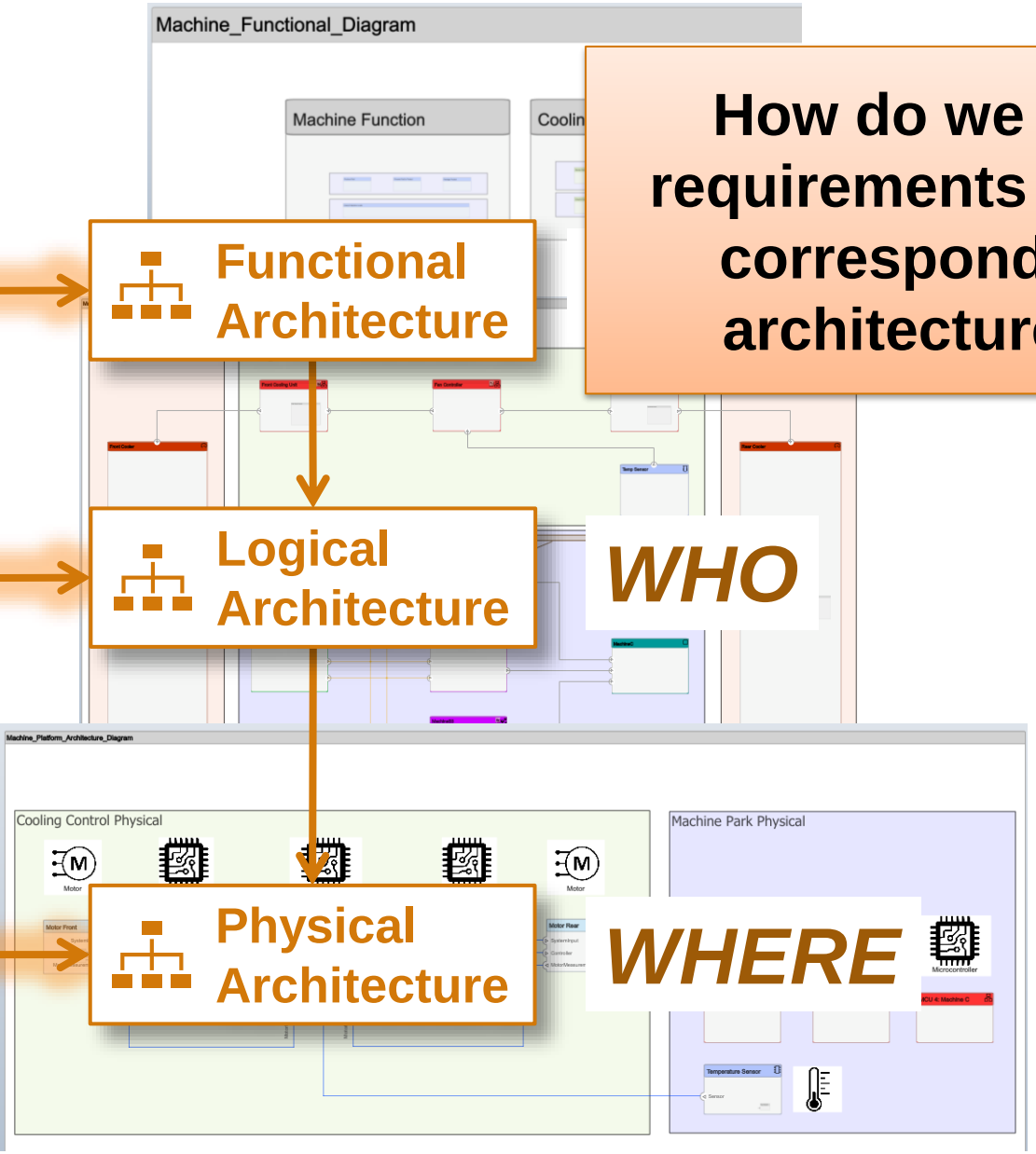

REQUIREMENTS
New Requirement Set
Open
Save
Import
Close
FILE
Add Requirement
Delete
Promote Requirement
Demote Requirement
REQUIREMENTS
Add Link
Delete
Clear Issue
LINKS
Show Requirements
Show Links
VIEW
Search
EDIT
Traceability Matrix
ANALYSIS
Export
SHARE
Index
ID
Summary
MyRequirements
1 ReqSys1 Functional Requirements
1.1 ReqSys1.1 Cooling Function
1.1.1 ReqSys1... Cooling shall be applied when machin...
1.2 ReqSys1.2 Machine Function
2 ReqSys2 System Requirements
3 ReqSys3 Hardware Requirements
4 ReqSys4 Safety Requirements
4.1 ReqSys4.1 The system shall be safe against over...
4.2 ReqSys4.2 If cooling is not effective turn off mac...
MyRequirementsFanController
1 ReqFAN1 Cooling shall be applied in 1 second
2 ReqFAN2 Fan controller shall include debouncin...
3 ReqFAN3 Alarm conditions, Green/Orange/Red
4 ReqFAN4 When cooling is needed fan speed sh...
5 ReqFAN5 When cooling is not effective after a d...
6 VerFAN Justifications
MyRequirementsFieldOrientedController
Requirement: ReqSys4.2
Details
Properties
Type: Functional
Index: 4.2
Custom ID: ReqSys4.2
Summary: If cooling is not effective turn off machines
Description
Rationale
Arial
10
Keywords:
Revision information:
Custom Attributes
Links
Implemented by:
Check if Cooler is safe
Check if Machine is safe
Fan Controller

Stakeholder Needs

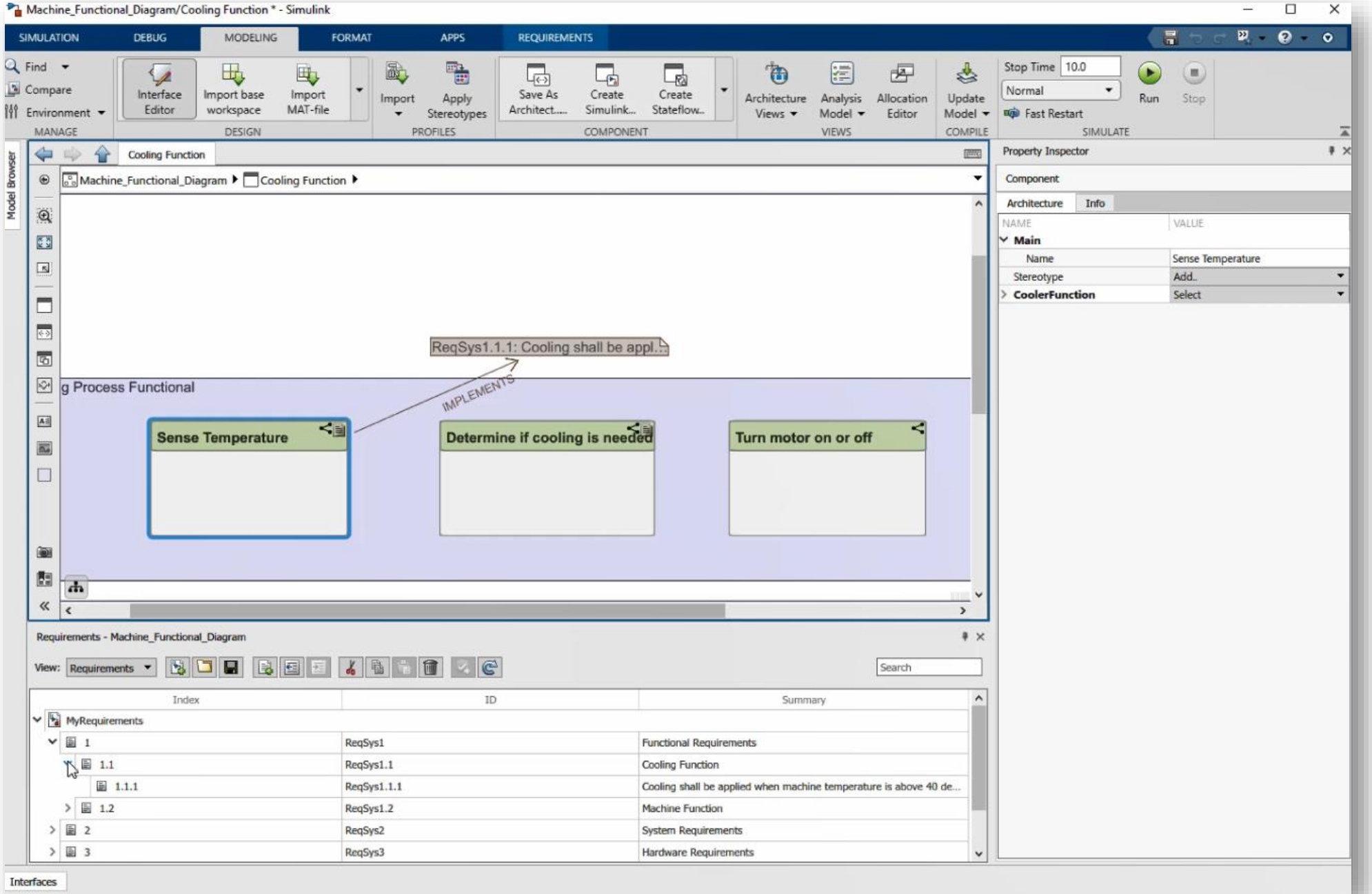
Functional Requirements

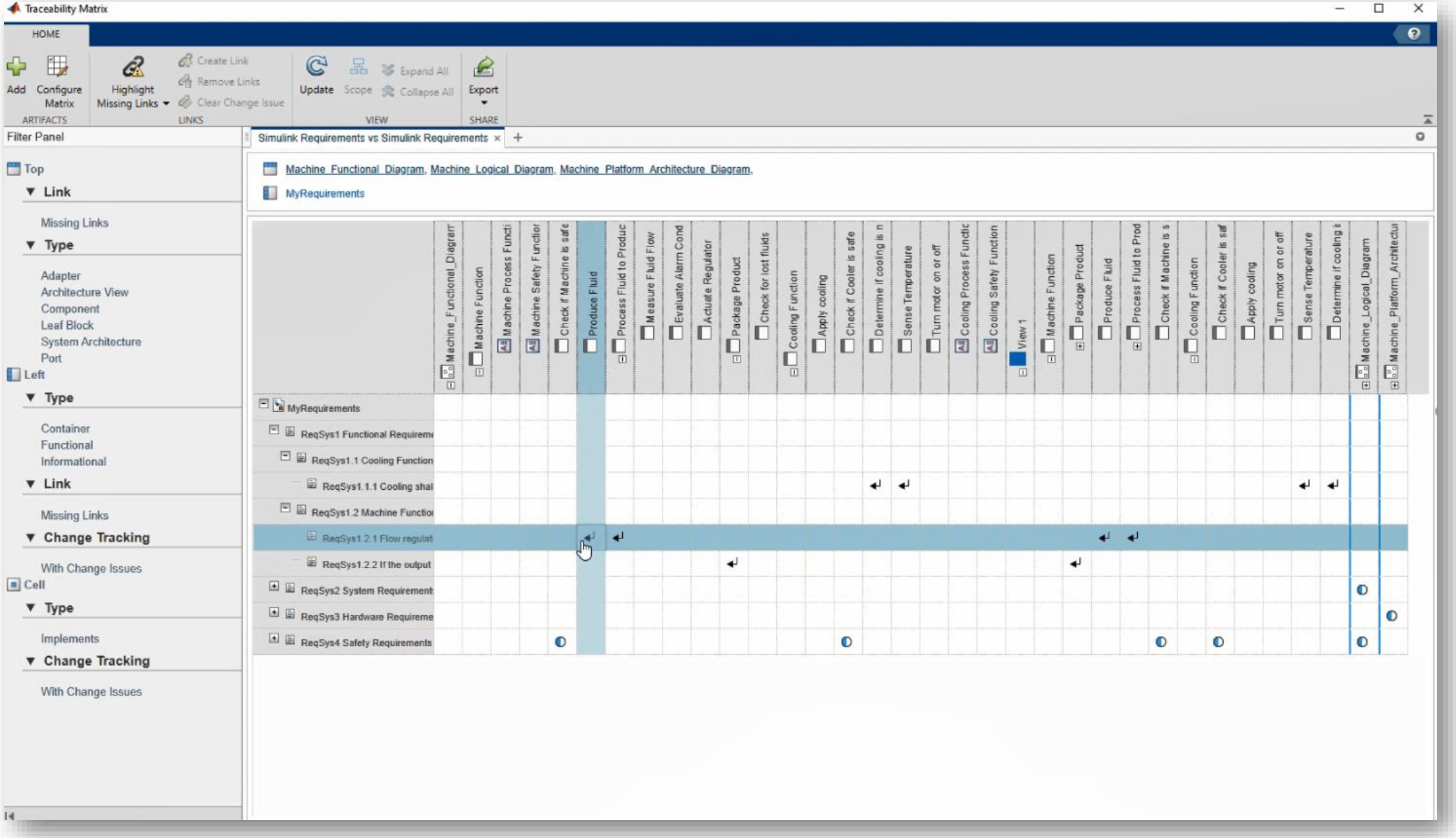
System Requirements

Hardware Requirements

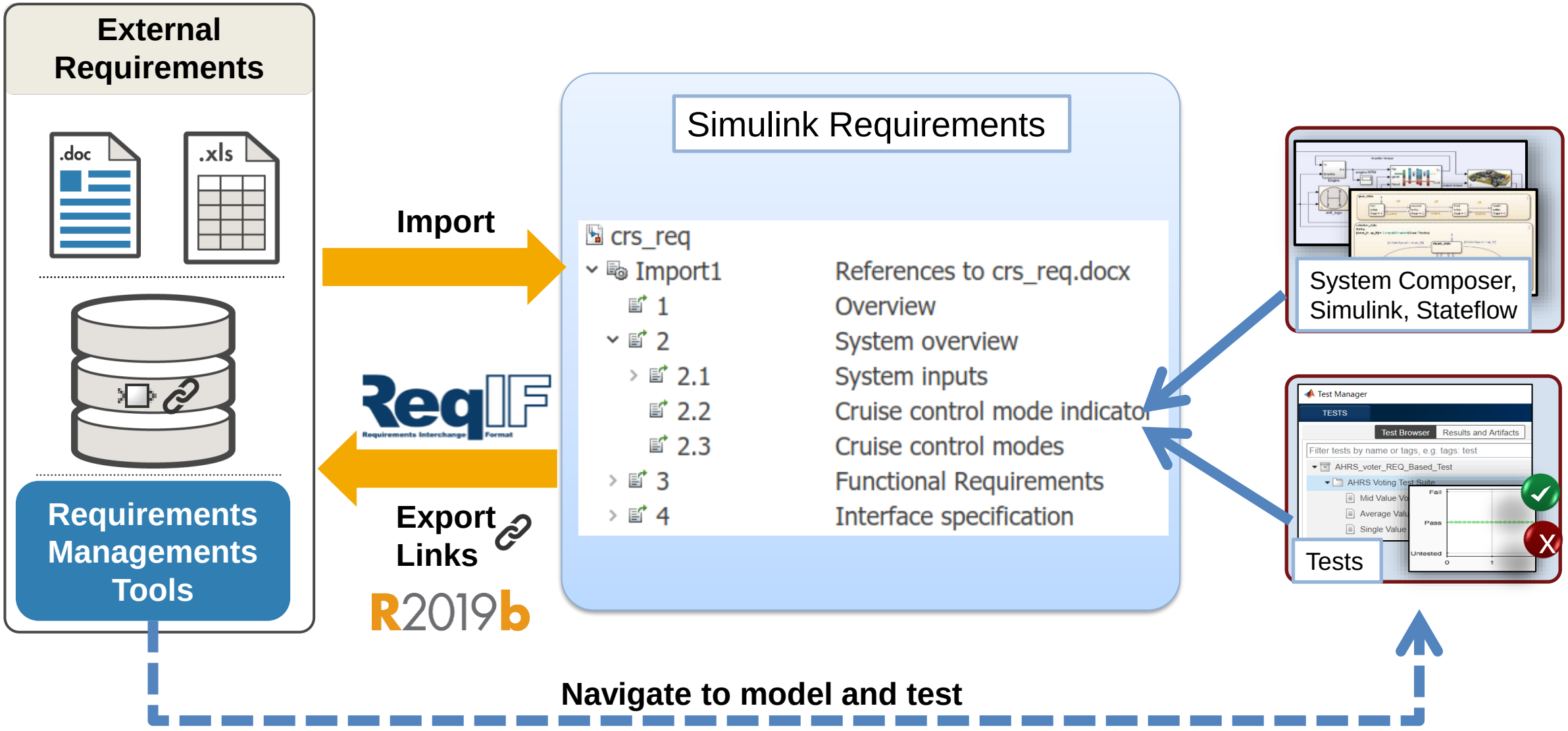


How do we link requirements to the corresponding architectures?


Machine_Functional_Diagram/Cooling Function * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
REQUIREMENTS
Find
Compare
Environment
MANAGE
Interface Editor
Import base workspace
Import MAT-file
DESIGN
Import
Apply Stereotypes
PROFILES
Save As Architect...
Create Simulink...
Create Stateflow...
COMPONENT
Architecture Views
Analysis Model
Allocation Editor
VIEWS
Update Model
COMPILE
Stop Time
10.0
Normal
Fast Restart
Run
Stop
SIMULATE
Model Browser
Cooling Function
Machine_Functional_Diagram
Cooling Function
ReqSys1.1.1: Cooling shall be appl...
IMPLEMENTS
g Process Functional
Sense Temperature
Determine if cooling is needed
Turn motor on or off
Property Inspector
Component
Architecture
Info
NAME
VALUE
Main
Name
Sense Temperature
Stereotype
Add..
CoolerFunction
Select
Requirements - Machine_Functional_Diagram
View:
Requirements
Search
Index
ID
Summary
MyRequirements
1
ReqSys1
Functional Requirements
1.1
ReqSys1.1
Cooling Function
1.1.1
ReqSys1.1.1
Cooling shall be applied when machine temperature is above 40 de...
1.2
ReqSys1.2
Machine Function
2
ReqSys2
System Requirements
3
ReqSys3
Hardware Requirements
Interfaces


Traceability Matrix
HOME
Add
Configure Matrix
ARTIFACTS
Highlight Missing Links
Create Link
Remove Links
Clear Change Issue
LINKS
Update
Scope
Expand All
Collapse All
VIEW
Export
SHARE
Filter Panel
Top
Link
Missing Links
Type
Adapter
Architecture View
Component
Leaf Block
System Architecture
Port
Left
Type
Container
Functional
Informational
Link
Missing Links
Change Tracking
With Change Issues
Cell
Type
Implements
Change Tracking
With Change Issues
Simulink Requirements vs Simulink Requirements
Machine_Functional_Diagram, Machine_Logical_Diagram, Machine_Platform_Architecture_Diagram,
MyRequirements
Machine_Functional_Diagram
Machine Function
Machine Process Functi
Machine Safety Function
Check if Machine is safe
Produce Fluid
Process Fluid to Produc
Measure Fluid Flow
Evaluate Alarm Cond
Actuate Regulator
Package Product
Check for lost fluids
Cooling Function
Apply cooling
Check if Cooler is safe
Determine if cooling is n
Sense Temperature
Turn motor on or off
Cooling Process Functic
Cooling Safety Function
View 1
Machine Function
Package Product
Produce Fluid
Process Fluid to Prod
Check if Machine is s
Cooling Function
Check if Cooler is saf
Apply cooling
Turn motor on or off
Sense Temperature
Determine if cooling is
Machine_Logical_Diagram
Machine_Platform_Architectu
MyRequirements
ReqSys1 Functional Requirem
ReqSys1.1 Cooling Function
ReqSys1.1.1 Cooling shal
ReqSys1.2 Machine Functio
ReqSys1.2.1 Flow regulat
ReqSys1.2.2 If the output
ReqSys2 System Requirement
ReqSys3 Hardware Requireme
ReqSys4 Safety Requirements

Export links for traceability to model and test



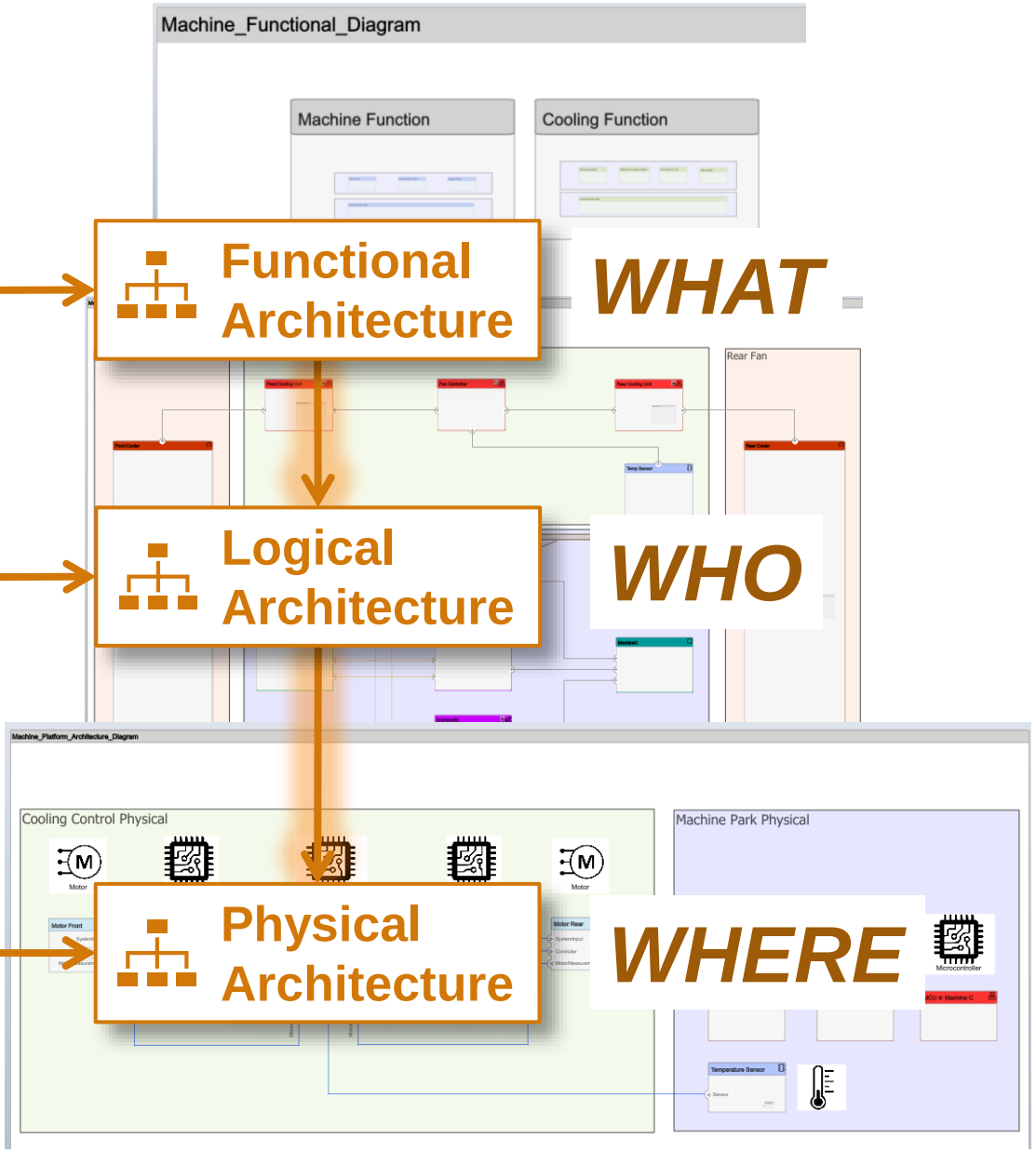
Stakeholder Needs

Functional Requirements

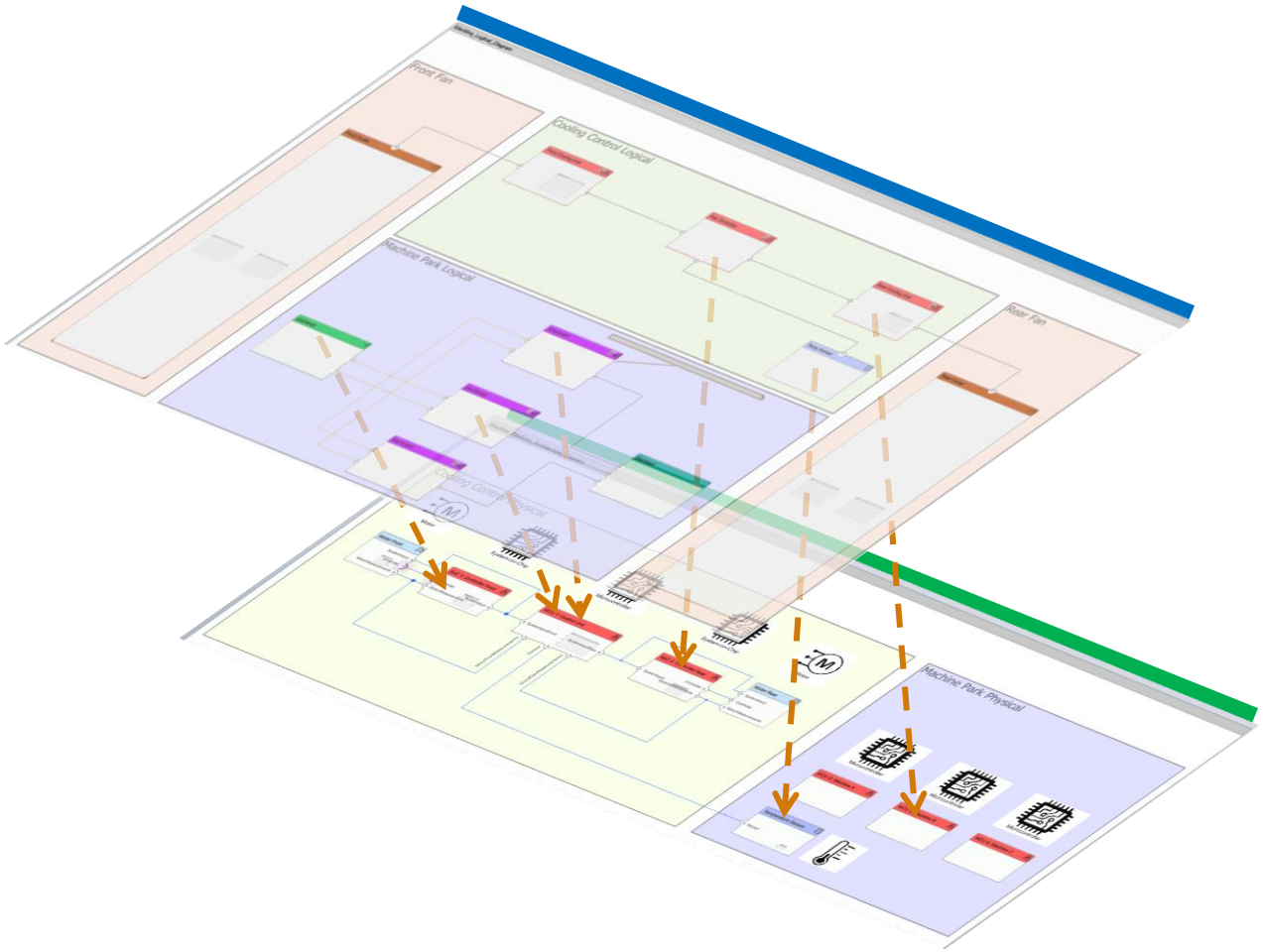
How do we show the relationships between architectures?

ts

Hardware Requirements



Allocating between architectures



Allocation Editor

ALLOCATIONS

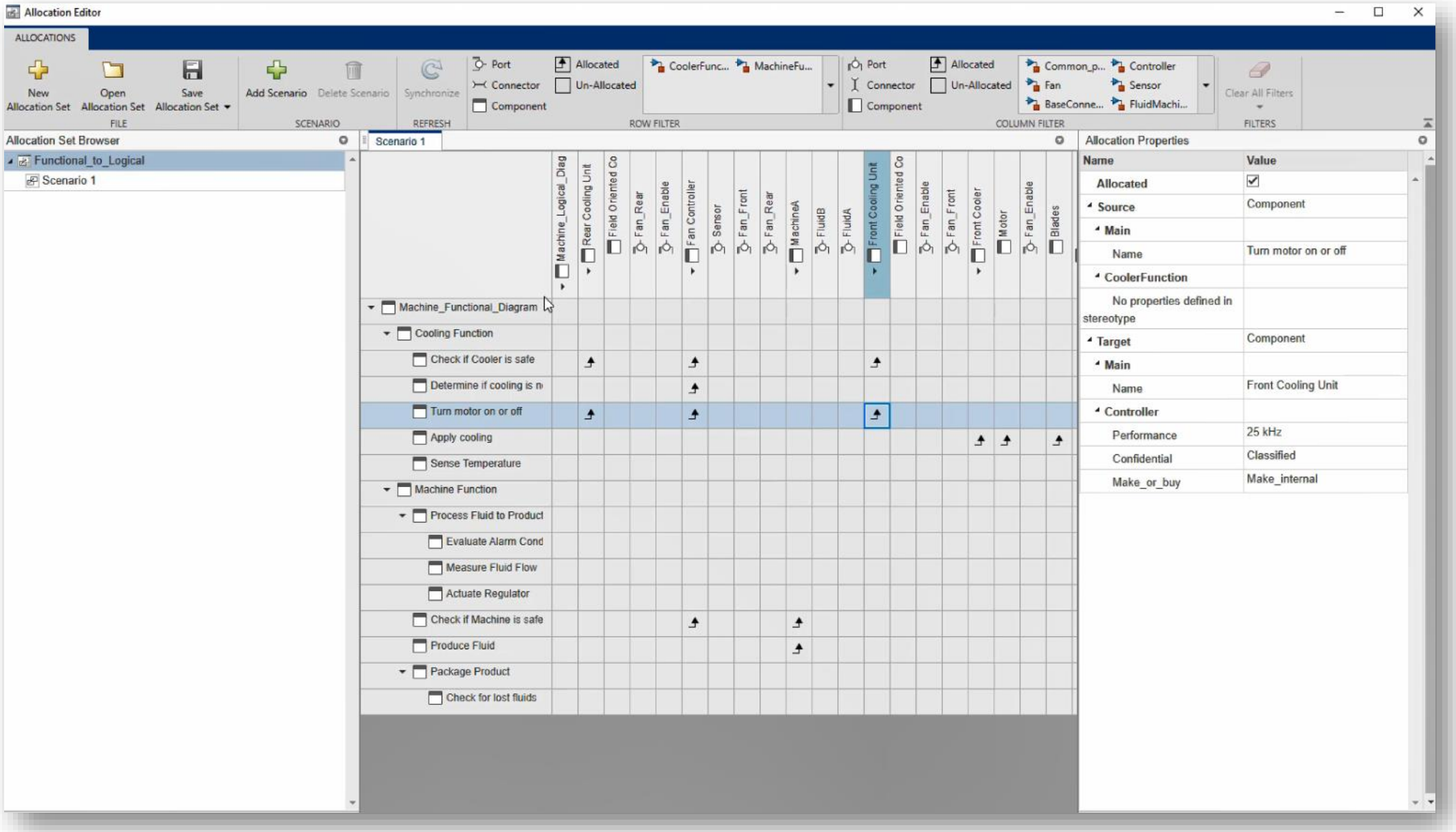
New Allocation Set | Open Allocation Set | Save Allocation Set — FILE
Add Scenario | Delete Scenario — SCENARIO
Synchronize — REFRESH
Port | Connector | Component | Allocated | Un-Allocated — ROW FILTER

ALLOCATION SET BROWSER

- Functional_to_Logical
  - Scenario 1
- Logical_to_Physical
  - Scenario 1

Scenario 1

	Machine_Platform_Arc	MCU 1: Control Un	SoC 2: Controller F	Motor Front	Motor Rear	SoC 1: Controller F	Temperature Sensc	MCU 4: Machine C	MCU 2: Machine A	MCU 3: Machine B
Machine_Logical_Diagram										
Rear Cooling Unit			↑							
MachineB1										↑
Fan Controller		↑								
MachineB3										↑
MachineA									↑	
Front Cooling Unit						↑				
Front Cooler										
Motor				↑						
MachineB2										↑
MachineC								↑		
Rear Cooler										
Motor					↑					
Temp Sensor							↑			


Allocation Editor
ALLOCATIONS
New Allocation Set
Open Allocation Set
Save Allocation Set
FILE
Add Scenario
Delete Scenario
SCENARIO
Synchronize
REFRESH
Port
Connector
Component
Allocated
Un-Allocated
CoolerFunc...
MachineFu...
ROW FILTER
Common_p...
Controller
Fan
Sensor
BaseConne...
FluidMachi...
COLUMN FILTER
Clear All Filters
FILTERS
Allocation Set Browser
Functional_to_Logical
Scenario 1
Machine_Logical_Diag
Rear Cooling Unit
Field Oriented Co
Fan_Rear
Fan_Enable
Fan Controller
Sensor
Fan_Front
Fan_Rear
MachineA
FluidB
FluidA
Front Cooling Unit
Field Oriented Co
Fan_Enable
Fan_Front
Front Cooler
Motor
Fan_Enable
Blades
Machine_Functional_Diagram
Cooling Function
Check if Cooler is safe
Determine if cooling is n
Turn motor on or off
Apply cooling
Sense Temperature
Machine Function
Process Fluid to Product
Evaluate Alarm Cond
Measure Fluid Flow
Actuate Regulator
Check if Machine is safe
Produce Fluid
Package Product
Check for lost fluids
Allocation Properties
Name
Value
Allocated
Source
Component
Main
Name
Turn motor on or off
CoolerFunction
No properties defined in stereotype
Target
Component
Main
Name
Front Cooling Unit
Controller
Performance
25 kHz
Confidential
Classified
Make_or_buy
Make_internal

Assess different allocation scenarios quantitatively

Allocate Architectures in a Tire Pressure Monitoring System

R2021a

This example shows how to use allocations to analyze a tire pressure monitoring system.

View MATLAB Command

Overview

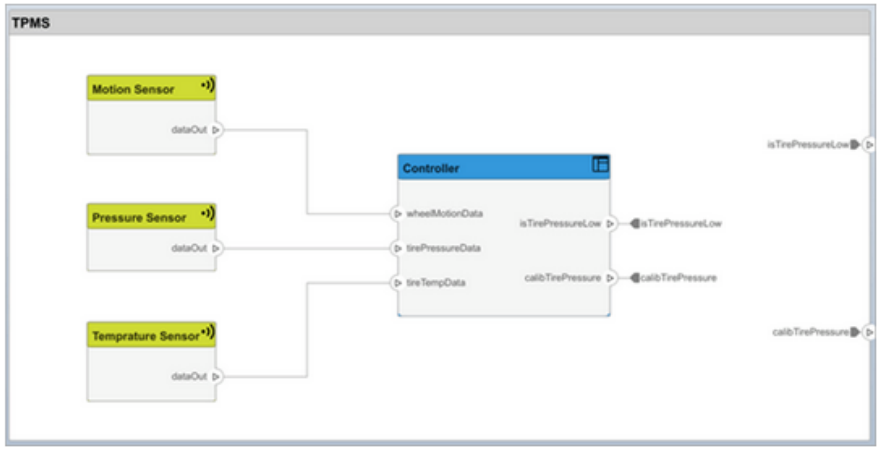
In systems engineering, it is common to describe a system at different levels of abstraction. For example, you can describe a systen not have any behavior associated with them but most likely trace back to some operating requirements the system must fulfill. We re *architecture*. In this example, an automobile tire pressure monitoring system is described in three different architectures:

1. Functional Architecture — Describes the system in terms of its high-level functions. The connections show dependencies betwe
2. Logical Architecture — Describes the system in terms of its logical components and how data is exchanged between them. Add simulation.
3. Platform Architecture — Describes the physical hardware needed for the system at a high level.

The allocation process is defined as linking these three architectures that fully describe the system. The linking captures the informa accessible to the others.

Use this command to open the project.

```
scExampleTirePressureMonitorSystem
```

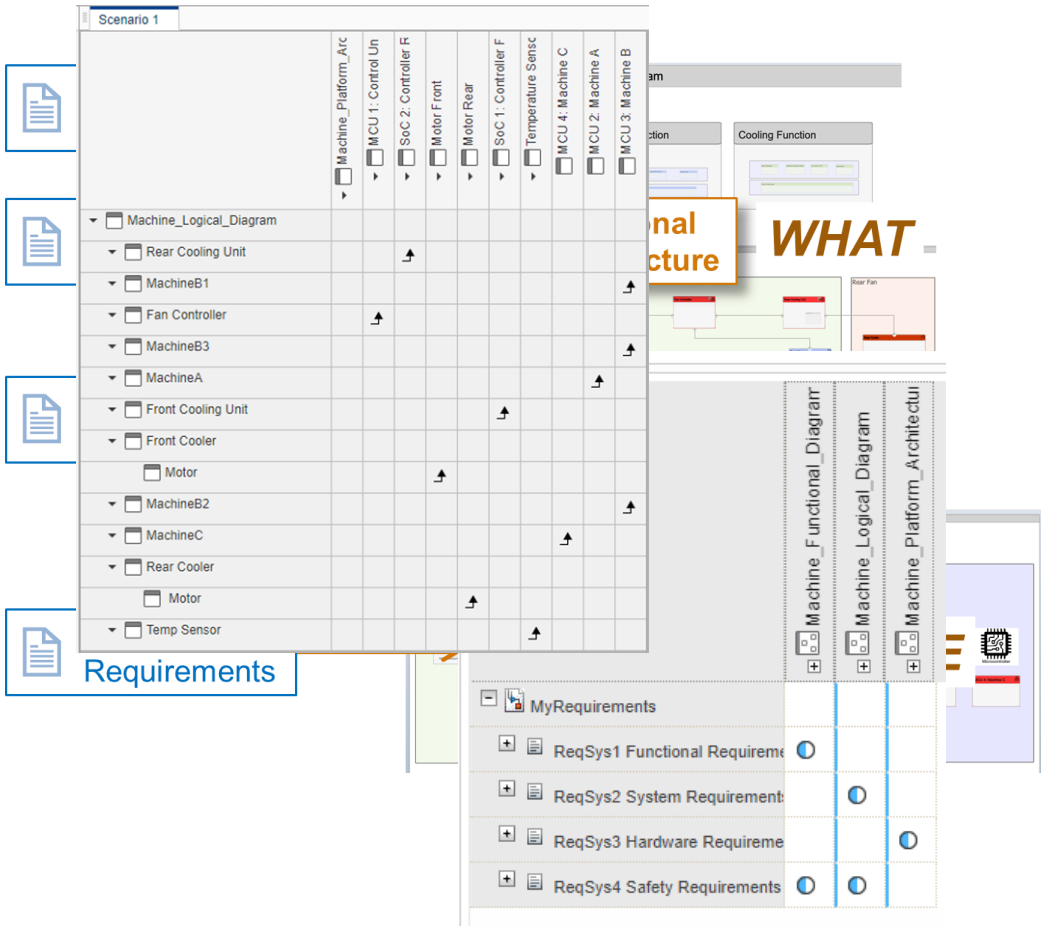


	Supplier A	Supplier B	Supplier C	Supplier D
Report Low Tire Pressure	1	0	0	0
Measure pressure on tire	0	0	1	0
Calculate Tire Pressure	0	1	0	0
Measure temprature of tire	0	0	0	1
Measure rotations	0	1	0	0
Calculate if pressure is low	1	0	0	0
Report Tire Pressure Levels	1	0	0	0
Measure Tire Pressure	0	0	0	0

	Scenario 1	Scenario 2
Front ECUMemory Used (MB)	110	90
Front ECU Memory (MB)	100	100
Front ECU Overloaded	1	0
Rear ECU Memory Used (MB)	0	20
Rear ECU Memory (MB)	100	100
Rear ECU Overloaded	0	0

https://www.mathworks.com/help/systemcomposer/ug/allocate-architectures-of-the-tire-pressure-monitoring-system.html

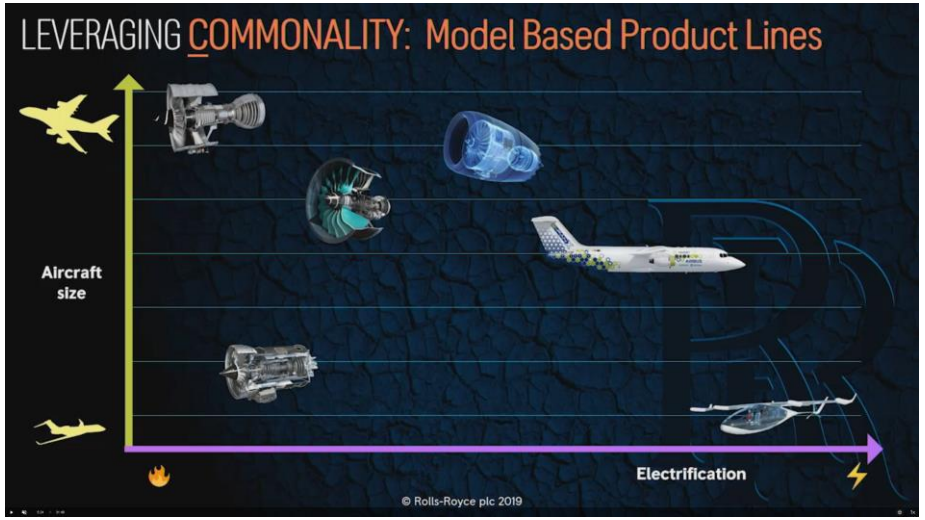
Key Takeaways



- You can import, write, and store textual requirements right **in the same environment** as your architecture and design models.
- You can **understand the impact of changes in your system** by establishing relationships among multiple requirements and architecture artifacts.
- You can **assess the completeness of your system** by visualizing those relationships.

Who is doing Model Based Systems Engineering with MathWorks tools?

Rolls Royce, UK Expo, Oct 2019



https://www.mathworks.com/videos/our-journey-towards-model-based-product-lines-1573233985120.html

System Architecture Modeling for Electronic Systems Using MathWorks System Composer and Simulink

Christopher B. Watkins
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
chris.watkins@gulfstream.com

Jerry Varghese
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
jerry.varghese@gulfstream.com

Michael Knight
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
michael.knight@gulfstream.com

Becky Petteys
The MathWorks, Inc.
Natick, Massachusetts, U.S.
bpetteys@mathworks.com

Jordan Ross
The MathWorks, Inc.
Natick, Massachusetts, U.S.
jordanr@mathworks.com

***Abstract*—Electronic system architectures have traditionally been documented as static block diagrams in tools such as Microsoft® Visio® or through a richer modeling approach such as Systems Modeling Language (SysML). These approaches did not fully meet the modeling needs for the Gulfstream authors, which led to an alternative approach.**

This paper introduces the Electronic System Architecture Modeling (eSAM) method, which leverages a new system architecture modeling tool called System Composer™. eSAM was created by the authors to define a standard method for applying the generic System Composer modeling constructs to build functional, physical, and logical architecture models of electronic systems. The eSAM methods are applied to an example avionics architecture to demonstrate capabilities needed for system modeling, collaborative OEM-supplier workflows, data management and ICD generation, systems integration activities, generation of system architecture deliverables for the avionics

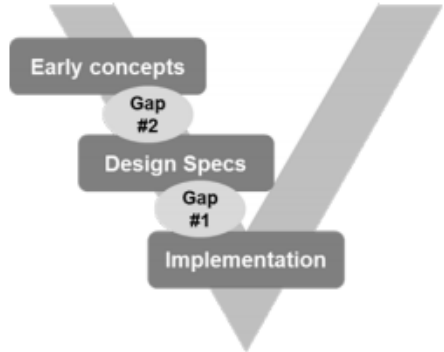


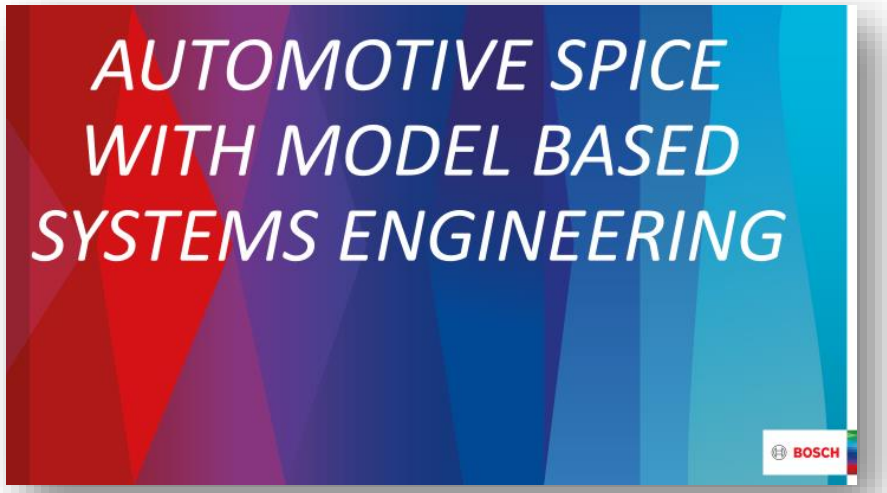
Figure 1: Simplified development process

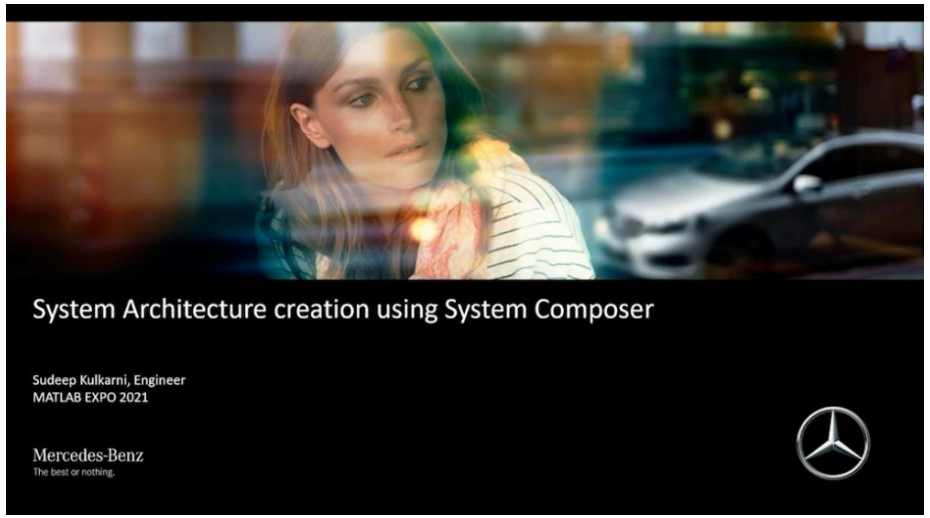
https://ieeexplore.ieee.org/document/9256753

Who is doing Model Based Systems Engineering with MathWorks tools?

MathWorks Automotive Conference 2021

Felix Raab, Bosch



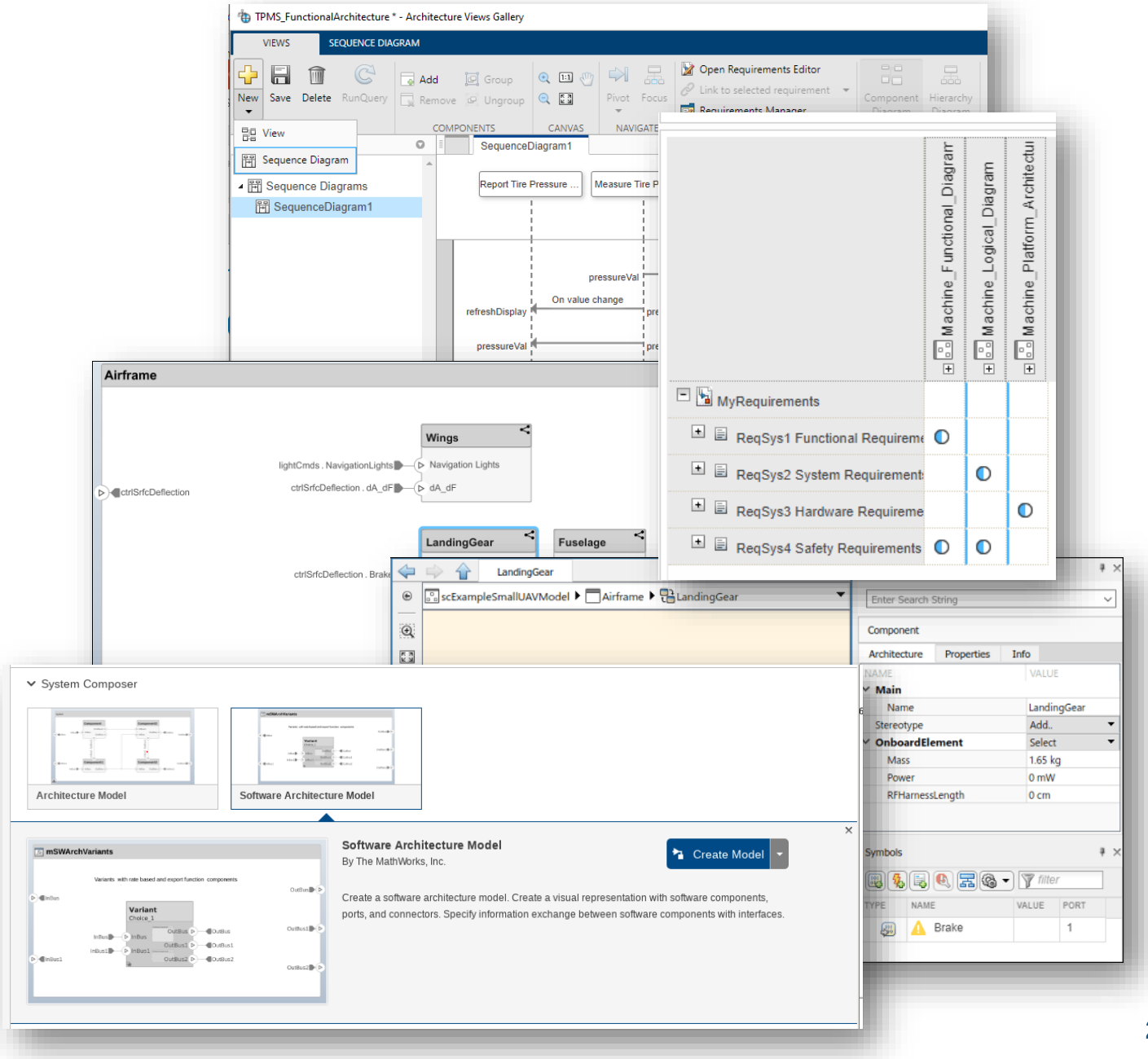


Sudeep Kulkarni,
Mercedes-Benz

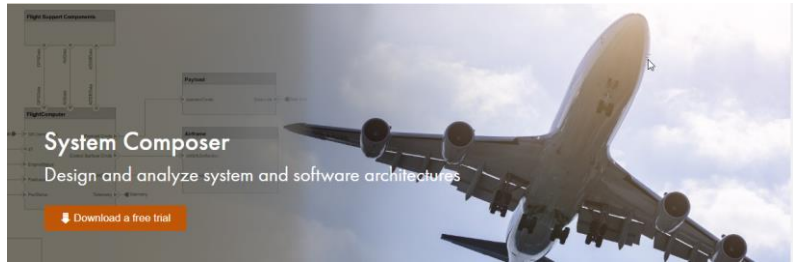
New features in R2021a

- System Composer
  - Sequence diagrams
  - Stateflow charts in components
  - Software architectures

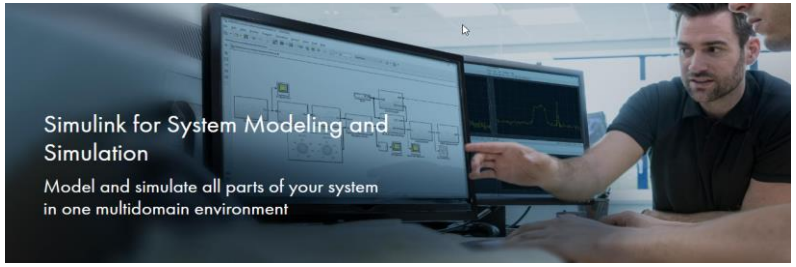
- Simulink Requirements
  - Editor improvements
  - Multi-artifact traceability matrix



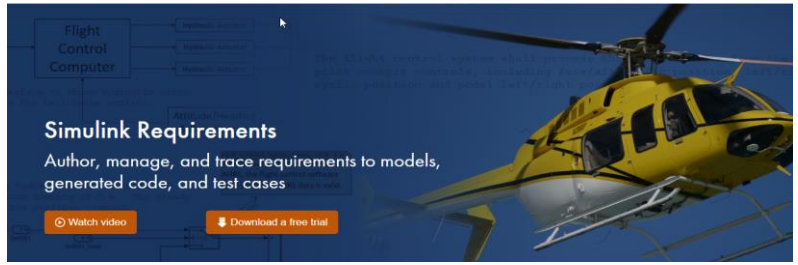
Learn More



[System Composer]

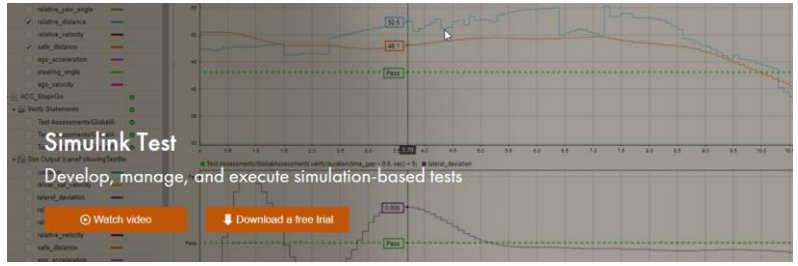


[Model-Based Systems Engineering]



[Simulink Requirements]

[System Modeling and Simulation]



[Simulink Test]

[AUTOSAR]

MATLAB EXPO 2021

Thank you