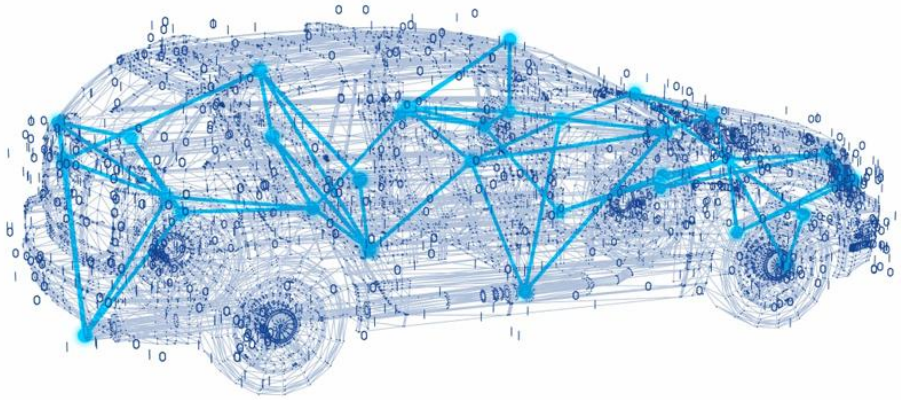
Using Simulatable Requirements Models to improve Team Collaboration

Stephan van Beek
EMEA Principal Application Engineer

What is Systems Engineering?

INCOSE defines Systems Engineering as:

“ Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.

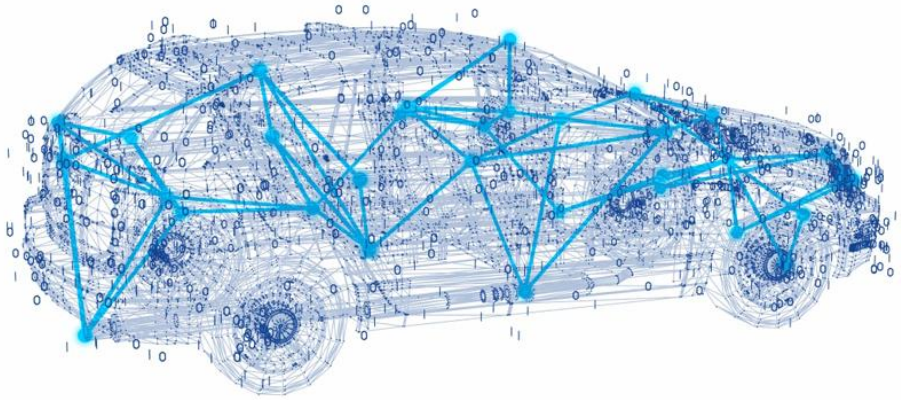


INCOSE: What is Systems Engineering?
INCOSE: Systems Engineering Glossary

What is Model-Based Systems Engineering?

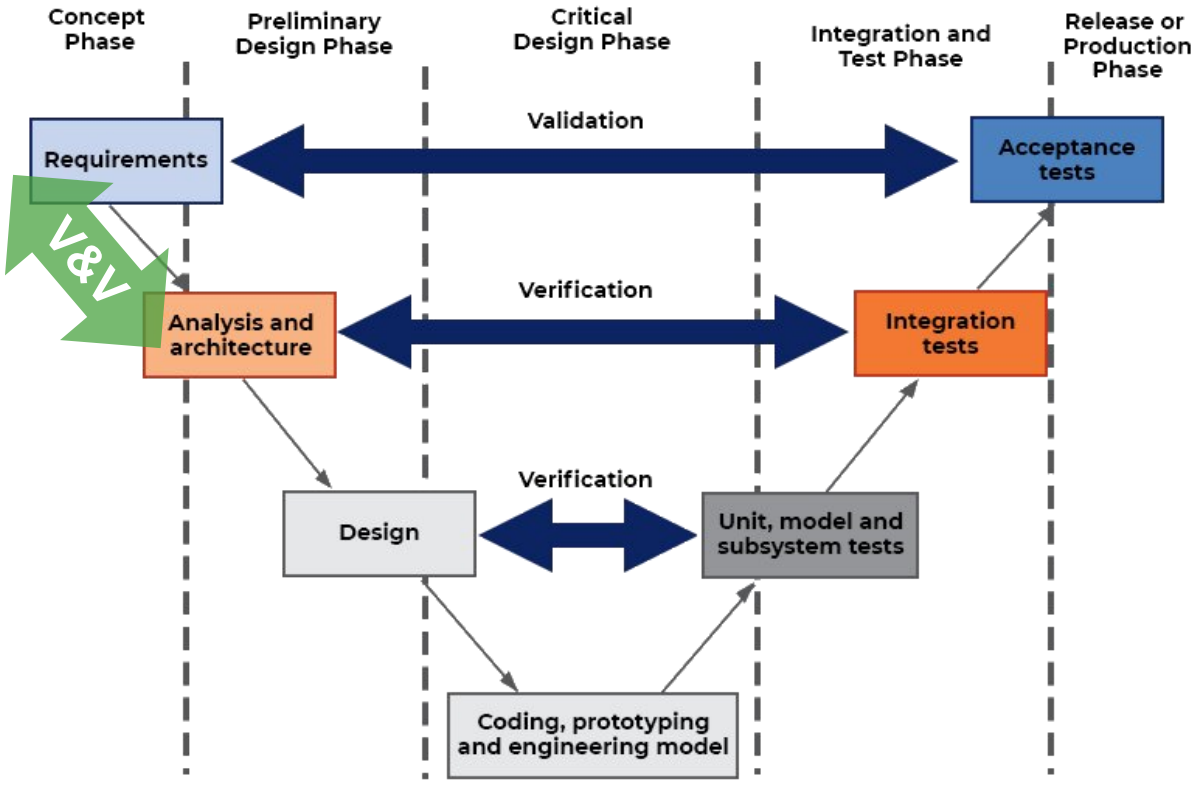
INCOSE defines MBSE as:

“ Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.



INCOSE Model Based Systems Engineering (MBSE) Initiative
Sanford Friedenthal, Regina Griego, Mark Sampson

Verification and Validation from a Systems Engineering viewpoint



Validation: *stakeholders perspective* on a product.

Validation is the process by which engineers ensure that the system will meet stakeholder needs and requirements.

Verification: *engineers perspective* on a product.

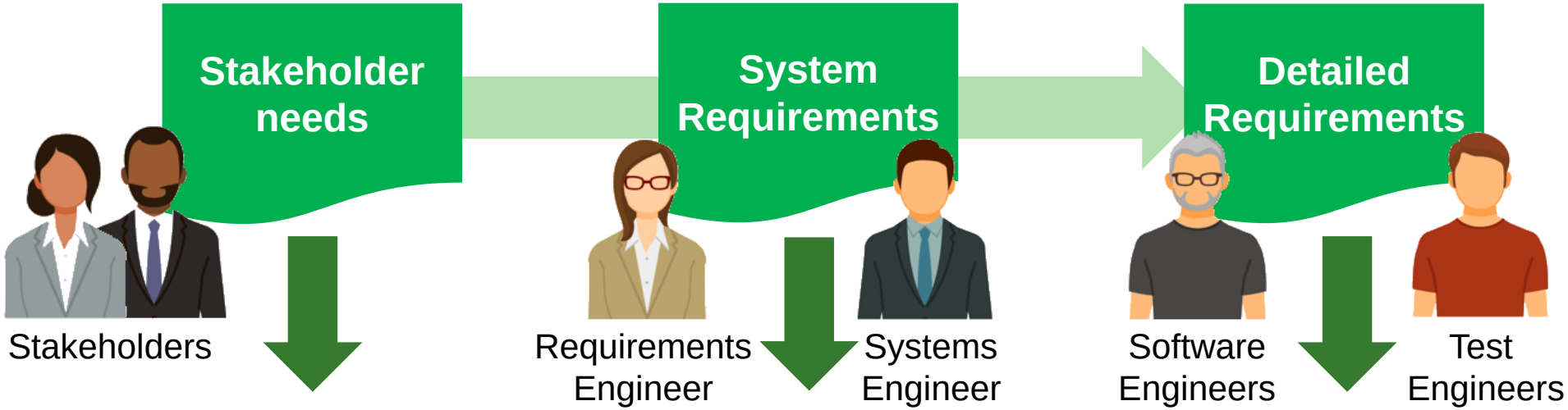
Provides information and evidence that the transformation was made according to the selected and appropriate methods, techniques, standards, or rules.

Model-Based System Engineering also involves both verification and validation activities on the requirements models.

https://sebokwiki.org/wiki/System_Verification
https://sebokwiki.org/wiki/System_Validation

Transform Stakeholder Needs into System Requirements

Requirements flow



Produced artifacts

Requirements set (textual)

Architecture models (descriptive)
Requirements set (textual)

Requirements set (textual)
Design models
Code (C/C++/HDL)
Test harnesses

Tools used

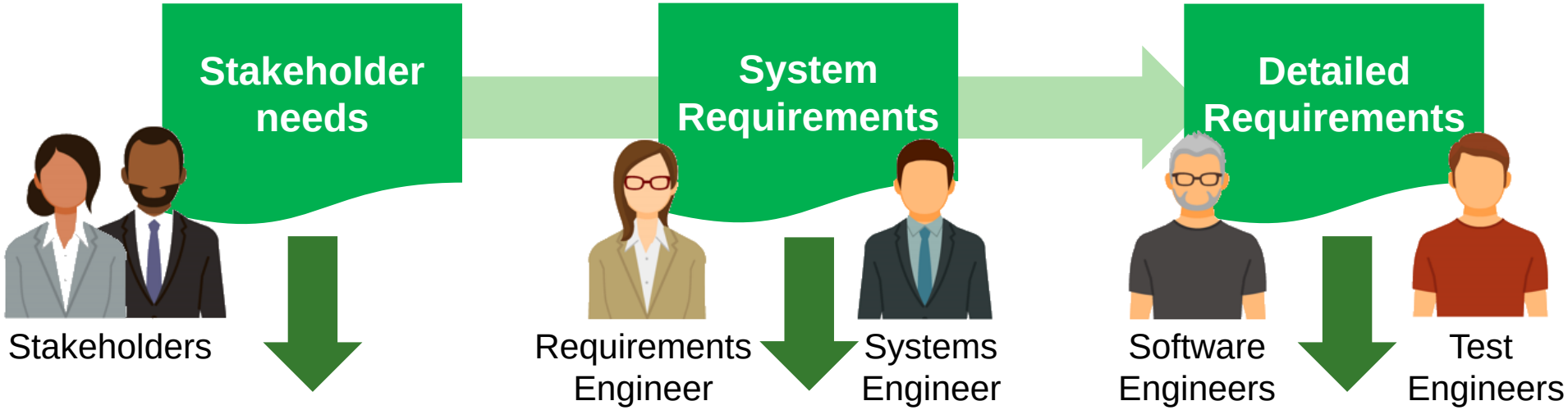
Design - MBD

Architecture - MBSE

Requirements - DOORS, DOORS NG, Polarion, ReqIF

Transform Stakeholder Needs into Simulatable System Requirements

Requirements flow



Produced artifacts

Requirements set (textual)

Architecture models
Requirements models
Test harnesses

Requirements models
Design models
Code (C/C++/HDL)
Test harnesses

Tools used

Design - MBD

Architecture - MBSE (System Composer)

Requirements - DOORS, DOORS NG, Polarion, ReqIF

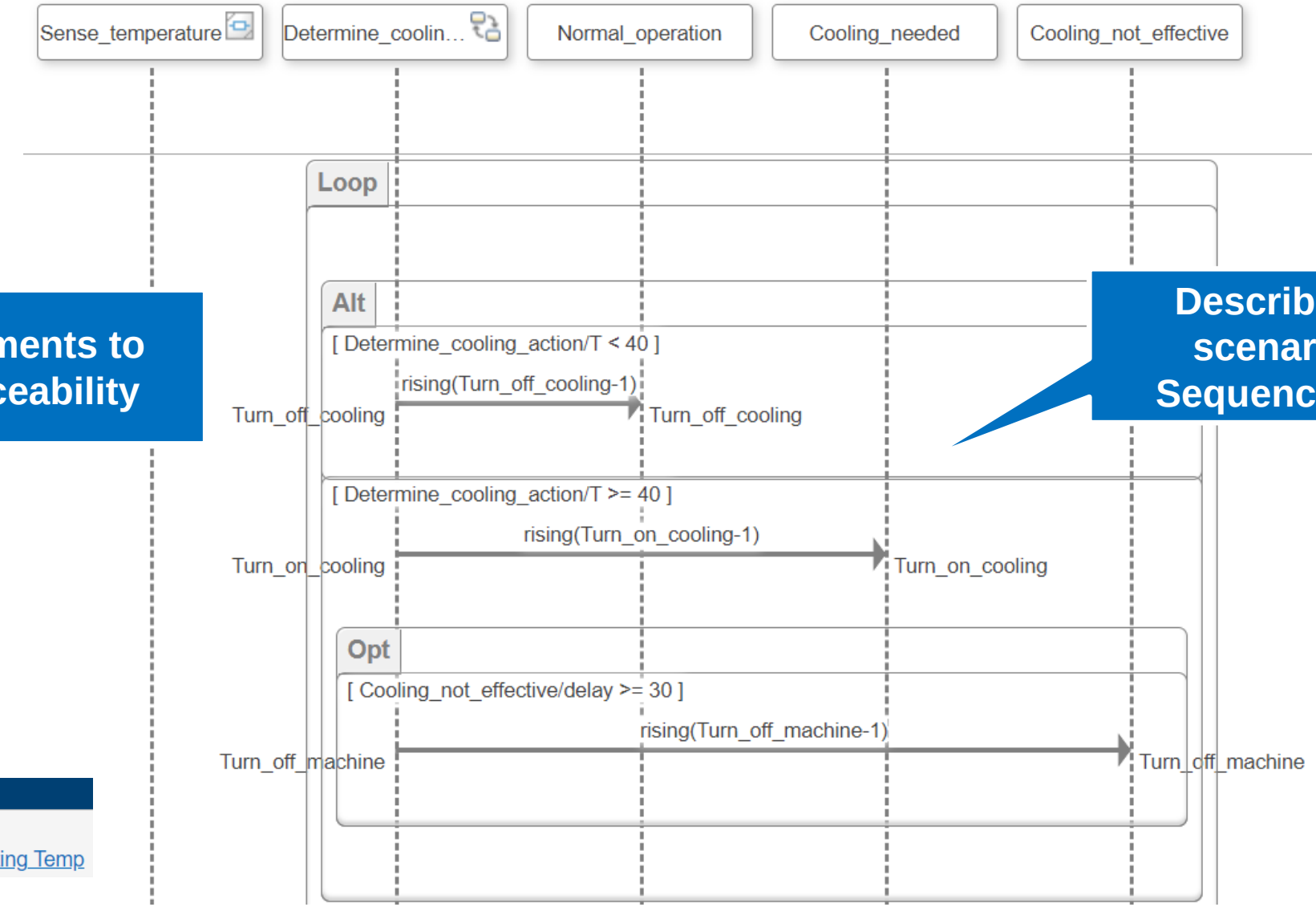
Case Study: Machine Cooling System, stakeholder needs …..

Provide a system which maintains the operating temperature of a machine, avoiding damage to the machine because of overheating.

- **[constraint]** Cooling system needs to maintain operating temperature below 40 degrees.
- **[constraint]** Cooling needs to be effective within a predetermined time.
- **[assumption]** Environmental temperature greater than -10 degrees and smaller than 80 degrees.

Validate and Understand Use-Case Behaviors

By means of **descriptive models**



Link requirements to facilitate traceability

Describe complex scenarios using Sequence Diagrams

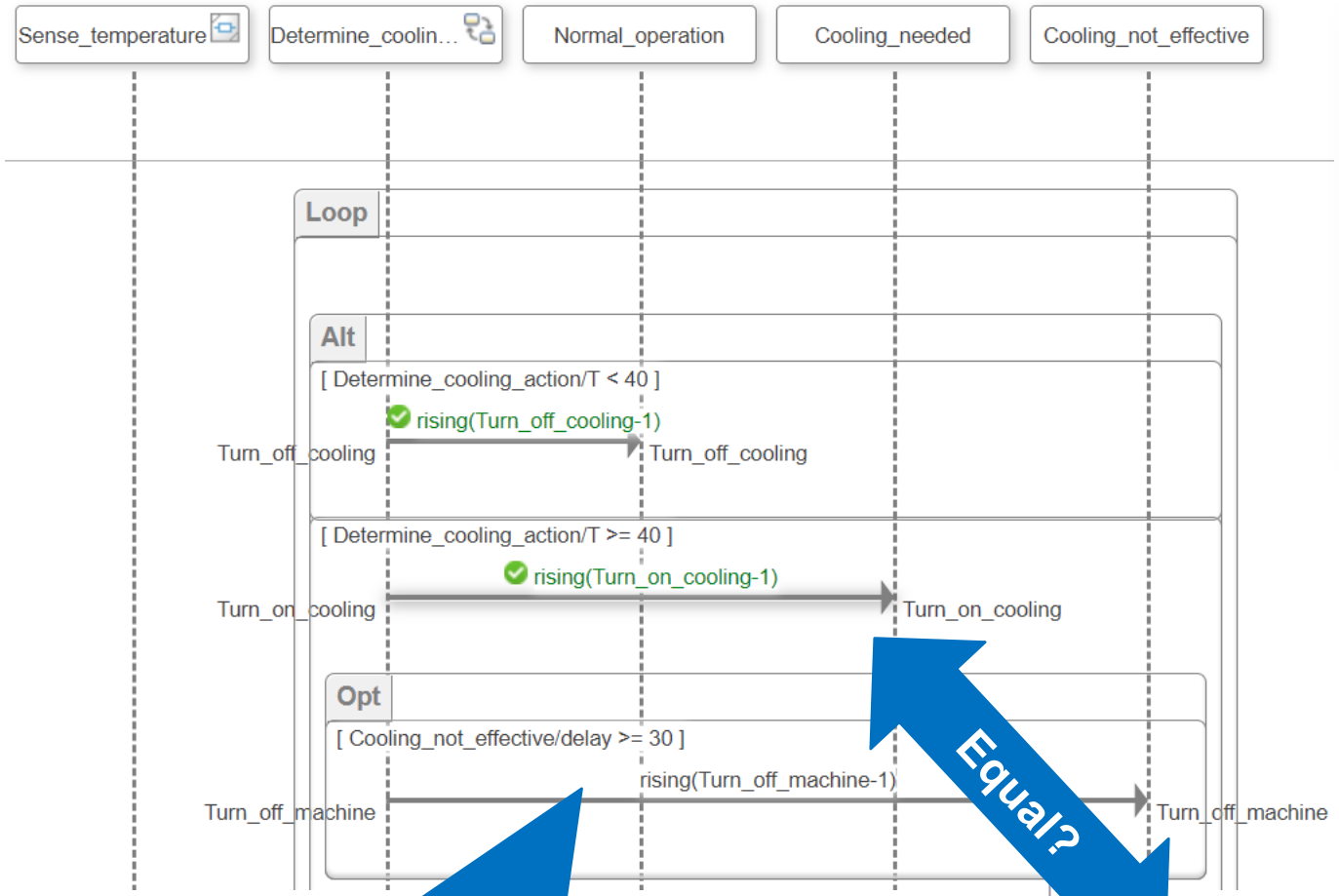
Requirement Links

⇒ **Describes:**

STAKEHOLDER-03 Operating Temp

Validate and Understand Use-Case Behaviors

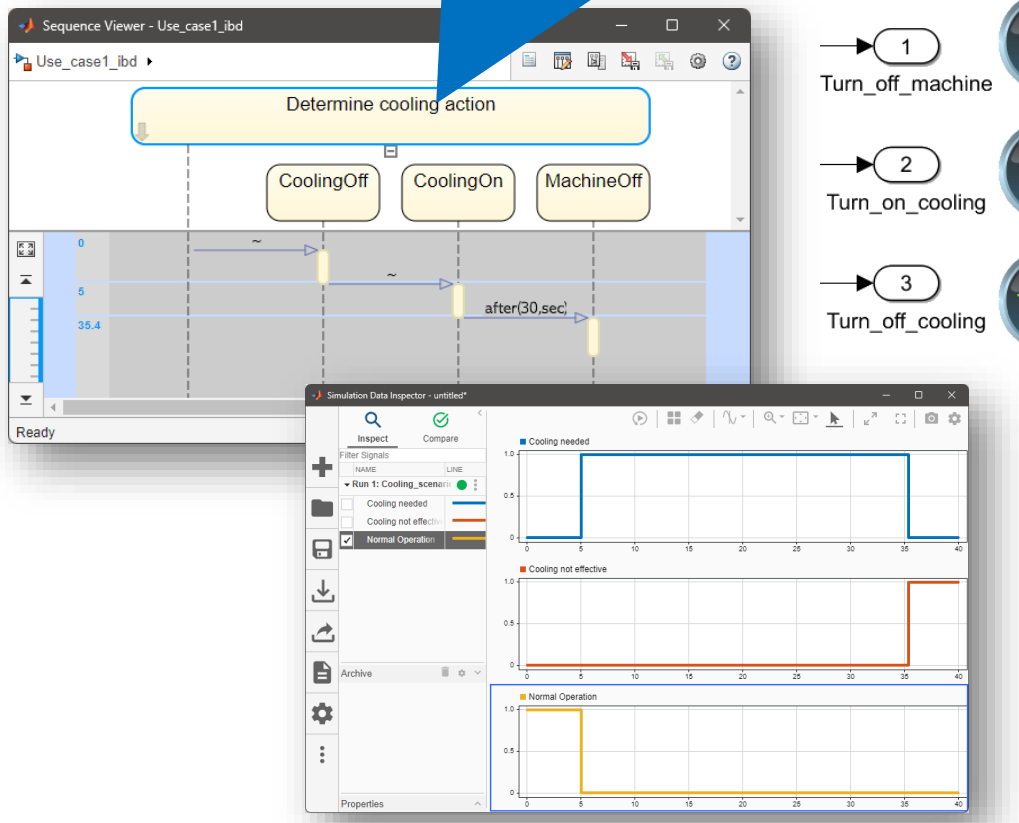
By means of **simulatable descriptions**

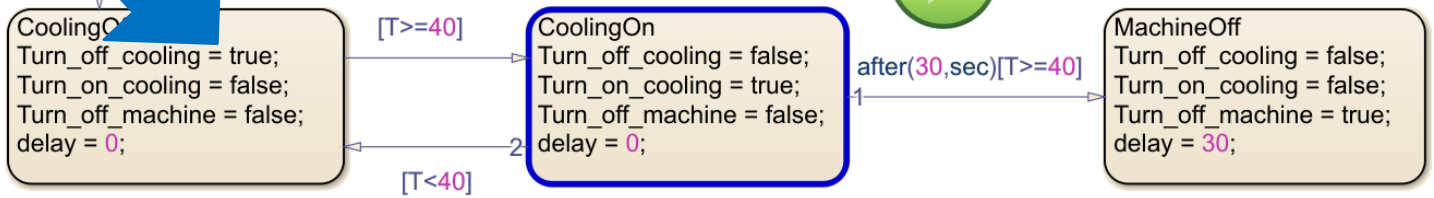


Verify expected behavior through simulation

Equal?

Visualize simulation results

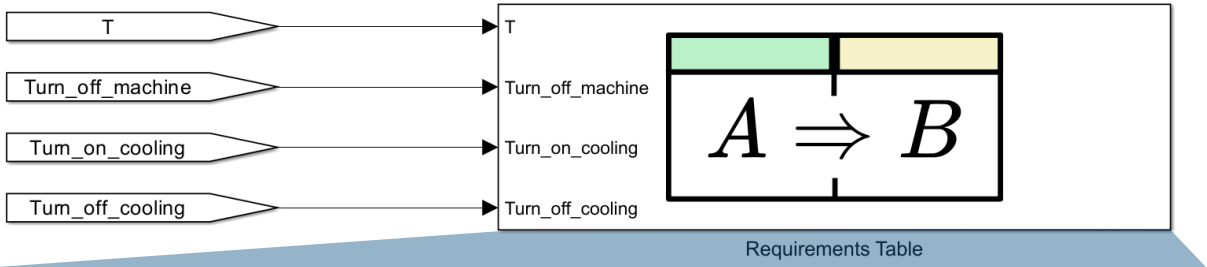




Validate and Understand Use-Case Behaviors

By means of **formal descriptions**

Formal description of requirements



Index	Summary	Precondition		Duration	Postcondition		
		T	prev(Turn_off_machine)		Turn_off_cooling	Turn_on_cooling	Turn_off_machine
1	Cooling off when T<40	<40	false		true	false	false
2	Temperature is T >= 40	>40	false				
2.1	Machine off cooling duration >=30 sec	>40		30	false	false	true
2.2 D	Cooling on cooling done within 30 sec	Else			false	true	false
3	When machine is off, it should stay off		true		false	false	true

Input condition to activate a requirement

Expected outcome of a requirement

Verify and Understand Use-Case Behaviors

By means of **formal descriptions**

Index	Summary	Precondition		Duration	Postcondition		
		T	prev(Turn_off_machine)		Turn_off_cooling	Turn_on_cooling	Turn_off_machine
1	Cooling off when T<40	<40	false		true	false	false
2	Temperature is T >= 40	>40	false				
2.1	Machine off cooling duration >=30 sec	>40		30	false		
2.2 D	Cooling on cooling done within 30 sec	Else			false		
3	When machine is off, it should stay off		true		false		

Formal description

Incompleteness Issues

Incompleteness 1: 'Turn_off_machine' is not specified at time 0 for the following inputs:

Time	0
Step	1
T	40

Incompleteness 2: 'Turn_on_cooling' is not specified at time 0 for the following inputs:

Time	0
Step	1
T	40

Incompleteness 3: 'Turn_off_cooling' is not specified at time 0 for the following inputs:

Time	0
Step	1
T	40

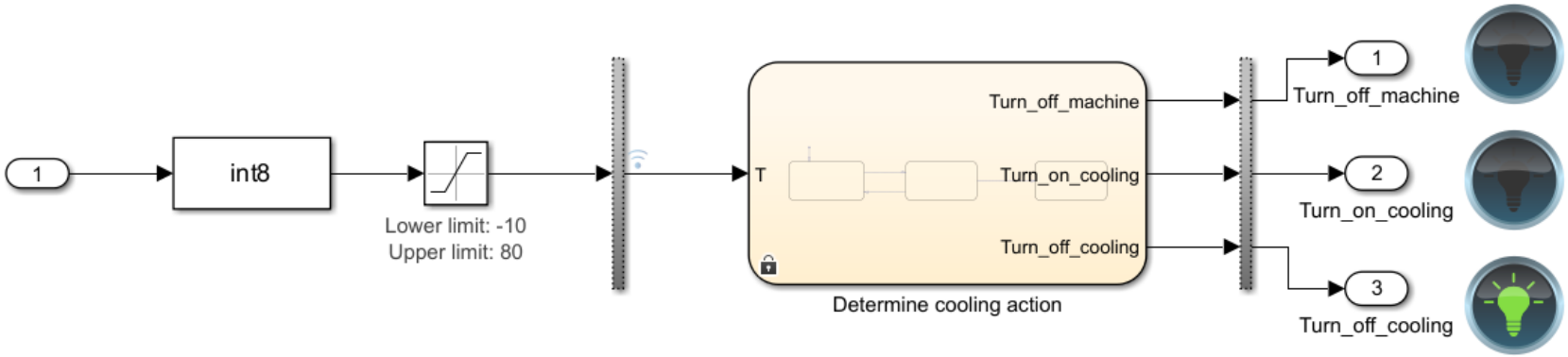
Formal analysis of requirements: completeness and consistency ➔ T = 40 is not specified!

Conclusions

- Requirements Models enable Automation for Verification and Validation
  - Complete, consistent and validated system requirements ➔ improved quality
- Collaboration between different teams
  - Executable requirements models improve communication between different teams
  - Simulation results can be presented from different viewpoints
- What comes next?
  - Re-use requirements models in downstream verification and validation activities
  - Digital thread enables automation

Re-using Requirements Models to Validate Design Models

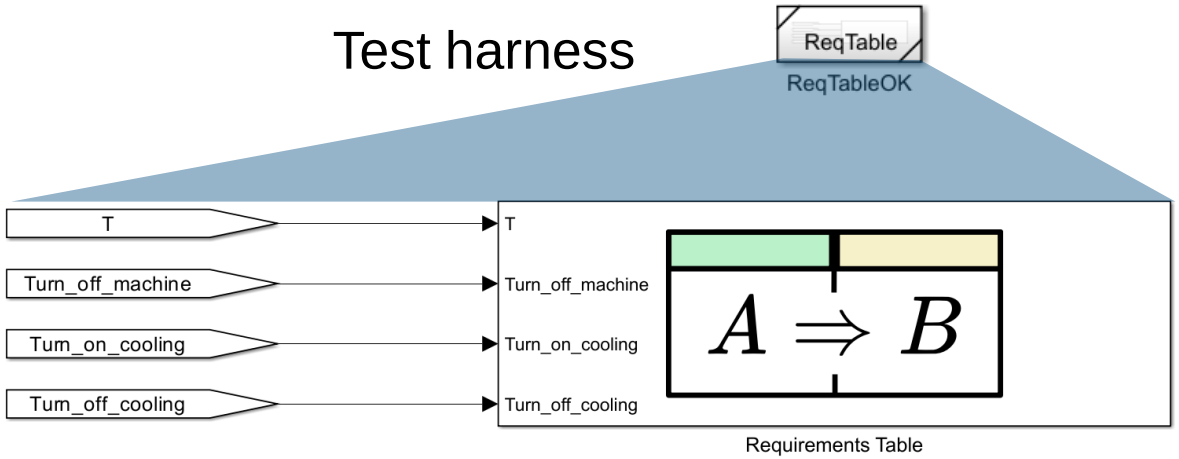
Secure compliance to requirements in the future by unit-testing



Test definition

NAME	STATUS
Results: 2023-Apr-04 10:49:58	2
validateCooling	2
Test cooling behavior	2
Cooling not effective	
Cooling effective	
Verify Statements	
R:1 (Requirements Table)	
R:2.1 (Requirements Table)	
R:2.2 (Requirements Table)	
R:3 (Requirements Table)	
Sim Output (Simulation : normal)	

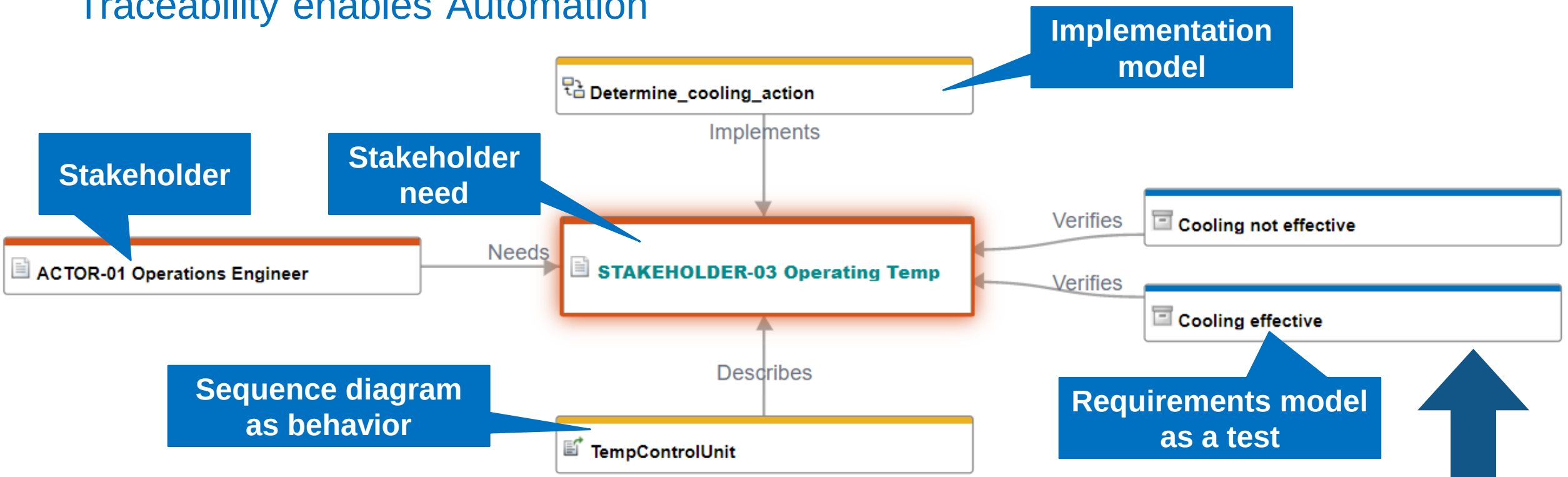
Test harness



Validate compliance to requirements through simulation

Follow the Digital Thread from Stakeholder to Detailed Requirements

Traceability enables Automation



Index	ID	Summary	Implemented	Verified
StakeholderRequirements				
1	-	Stakeholders		
2	-	Operating Environment		
2.1	STAKEHOLDER-03	Operating Temp		
2.2	STAKEHOLDER-05	Noise Pollution		
3	-	Performance		
4	STAKEHOLDER-15	Mean Repair Cost		
5	-	Safety		

- Described by:
  - TempControlUnit
- Implemented by:
  - Determine_cooling_action
- Needed by:
  - ACTOR-01 Operations Engineer
- Verified by:
  - Cooling not effective
  - Cooling effective

Status

MathWorks Value for Model-Based Systems Engineering

Maintain **requirements** as an **authoritative source of truth** throughout the **product development process**, by using (simulation) models to:

1. **Transform stakeholder requirements/needs**
   into design requirements using models, simulation and code generation
2. **Establish traceability**
   between requirements, architectures, designs and testcases
3. **Explore the design space**
   through (reusable) trade-off studies
4. **Manage system complexity**
   through views and traceable architecture models
5. **Connect system architecture**
   with software architecture and component implementations