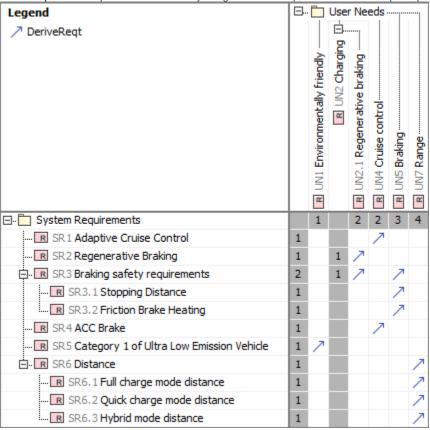# Requirement matrices

The matrices enables you to analyze, create, and modify relationships between Requirements and other design elements. It is especially valuable that you can display relationships that cannot be represented in diagrams, such as representations (classes by lifeline), behavior representations in other diagrams, operation representations by Call Behavior Actions, etc. All Requirement matrices allows you to perform Requirements gap and coverage analysis.

You can create four kind of Requirements matrices:

- Derive Requirement Matrix.
- Refine Requirement Matrix.
- Satisfy Requirement Matrix.
- Verify Requirement Matrix.

The different purposes for each matrix are illustrated below:

- **Derive Requirement Matrix** allows you to analyze, create, and modify Derive relationships between Requirements and other design elements. Rows represent the elements that are the clients of Derive relationship. Columns represent the Requirements that are the suppliers of Derive relationship. The example below is created by using the extract requirement values.mdzip sample model that comes only with SysML Plugin.
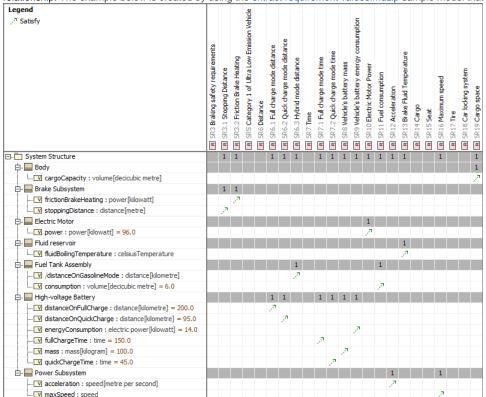

- **Refine Requirement Matrix** allows you to analyze, create, and modify Refine relationships between Requirements and other design elements. Rows represent the elements that are the clients of Refine relationship. Columns represent the Requirements that are the suppliers

of the Refine relationship. The example below is created by using the extract requirement values.mdzip sample model that comes only with SysML Plugin.

Refine Requirement Matrix — Legend: ↗ Refine

	UN1 Environmentally friendly	UN2 Charging	UN2.1 Regenerative braking	UN2.2 Plug-In charge	UN4 Cruise control	UN5 Braking	UN10 Safety	UN11 Power source management
Use Cases	2		3	11	5	8	2	2
⟳ Accelerate	2							2
▸ Provide electric engine power (context Hybrid Vehicle)	2 ↗							↗
▸ Provide power from electric and gasoline engines (context Hybrid Vehicle)	2 ↗							↗
⟳ Brake			2			5	6	
▸ ACC Brake (context Hybrid Vehicle)	3		1 ↗			↗	↗	
▸ Brake (classifier behavior)(context Driving Context)	2		1 ↗				↗	
▸ Detect presence and speed of vehicles (context Hybrid Vehicle)	2					↗	↗	
▸ Pre-charge brake for more aggressive braking (context Hybrid Vehicle)	2					↗	↗	
▸ Provide audible alert (context Hybrid Vehicle)	2					↗	↗	
▸ Send laser signal (context Hybrid Vehicle)	2					↗	↗	
Magic Hybrid Vehicle			1	11			2	2
⟳ Charge vehicle				11				
▸ Charge the vehicle (classifier behavior)(context Power Source Context)	1			1 ↗				

- **Satisfy Requirement Matrix** allows you to analyze, create, and modify Satisfy relationships between Requirements and other design elements. Rows represent the elements that are the clients of Satisfy relationship. Columns represent the Requirements that are the suppliers of the Satisfy relationship. The example below is created by using the extract requirement values.mdzip sample model that comes only with SysML Plugin.

Satisfy Requirement Matrix — Legend: ↗ Satisfy

	SR3 Braking safety requirements	SR3.1 Stopping Distance	SR3.2 Friction Brake Heating	SR5 Category 1 of Ultra Low Emission Vehicle	SR6 Distance	SR6.1 Full charge mode distance	SR6.2 Quick charge mode distance	SR6.3 Hybrid mode distance	SR7 Time	SR7.1 Full charge mode time	SR7.2 Quick charge mode time	SR8 Vehicle's battery mass	SR9 Vehicle's battery energy consumption	SR10 Electric Motor Power	SR11 Fuel consumption	SR12 Acceleration	SR13 Brake Fluid Temperature	SR14 Cargo	SR15 Seat	SR16 Maximum speed	SR17 Tire	SR18 Car locking system	SR19 Cargo space
System Structure	1	1				1	1	1		1	1	1	1	1	1	1	1			1			1
Body																							1
ⓥ cargoCapacity : volume[decicubic metre]																							↗
Brake Subsystem	1	1																					
ⓥ frictionBrakeHeating : power[kilowatt]			↗																				
ⓥ stoppingDistance : distance[metre]		↗																					
Electric Motor														1									
ⓥ power : power[kilowatt] = 96.0														↗									
Fluid reservoir																	1						
ⓥ fluidBoilingTemperature : celsiusTemperature																	↗						
Fuel Tank Assembly								1							1								
ⓥ /distanceOnGasolineMode : distance[kilometre]								↗															
ⓥ consumption : volume[decicubic metre] = 6.0															↗								
High-voltage Battery						1	1			1	1	1	1										
ⓥ distanceOnFullCharge : distance[kilometre] = 200.0						↗																	
ⓥ distanceOnQuickCharge : distance[kilometre] = 95.0							↗																
ⓥ energyConsumption : electric power[kilowatt] = 14.0													↗										
ⓥ fullChargeTime : time = 150.0										↗													
ⓥ mass : mass[kilogram] = 100.0												↗											
ⓥ quickChargeTime : time = 45.0											↗												
Power Subsystem																1				1			
ⓥ acceleration : speed[metre per second]																↗							
ⓥ maxSpeed : speed																				↗			

- **Verify Requirement Matrix** allows you to analyze, create, and modify Verify relationships between Requirements and other design elements. Rows represent the elements that are the clients of Verify relationship. Columns represent the Requirements that are the suppliers of the Verify relationship. The example below is created by using the Categorization requirements.mdzip sample model that comes only with Cameo Requirements Modeler Plugin.

Legend

↗ Verify

	Requirements
	⊟ 🗀 Requirements
	⊟ [R] TWCAT150 Project Categorization
	⊟ [R] TWCAT150.5 Category Management
	⊟ [R] TWCAT150.5.2 User-defined Categor

Test cases	TWCAT150.1 Category Definition	TWCAT150.4 Category Properties	TWCAT150.5.1 System Category Management Restrictions	TWCAT150.5.2.1 Category Management Actions	TWCAT150.5.2.2 Avoiding Name Clashes with System Category	TWCAT150.5.2.3 Ensuring Name Uniqueness	TWCAT150.5.2.4 Automatically Reassigning Projects to System Category	TWCAT150.5.2.5 Removing Category by Multiple Users at the Same Time	TWCAT150.5.2.6 Handling Category Removal While Others Are Renaming It	TWCAT150.5.2.7 Ensuring That Category Name is Entered	TWCAT150.5.2.8 Category Renaming by Multiple Users	TWCAT150.5.2.9 Assigning Project to Category While Others Are Deleting It	TWCAT150.5.2.10 Project Assigning to Categories by Multiple Users
⊟ 🗀 Test cases							1	1				1	
Adding Local Projects to Teamwork													
Assigning Project to Category While Others Are Deleting It	1											↗	
Automatically Reassigning Projects to System Category	1						↗						
Creating New Teamwork Projects													
Exporting Module to Teamwork													
Removing Category by Multiple Users at the Same Time	1							↗					

ⓘ **How to work with Requirement matrices**

Requirement matrices are based on Dependency Matrix. That's why all it's procedures are allowable for Requirement matrices:

📜 Unknown macro: 'list-children'