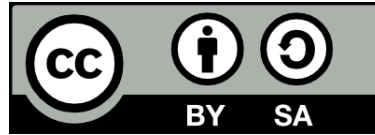


Guide to Tool-Specific S* Patterns Support in Dassault Cameo Systems Modeler™



Licensed under a Creative Commons

Attribution Share Alike-License CC BY SA International 4.0

License Link: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>

Uses are permitted under this license without further permission from the copyright owner, provided each use (1) is clearly marked to attribute the underlying work to “S*Patterns Community”, (2) provides a link to the CC BY SA license, (3) indicates if changes were made, (4) does not suggest the licensor endorses the user or use, (5) does not apply legal terms or technological measures that legally restrict others from doing anything the license permits, and (6) if you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Corporate Officer
ICTT System Sciences
378 South Airport Street
Terre Haute, IN 47803
[812-232-2208](tel:812-232-2208)

Systematica is a registered trademark of System Sciences, LLC.
Cameo Systems Modeler is a trademark of Dassault Systemes

Contents

Section	Pages
1. Introduction	7-11
2. Installation of Companion Assets	12-26
1. Systematica Profile	
2. Systematica Project Template	
3. Systematica Images	
4. Systematica Diagram Styles	
5. Pattern Export Tables Definition	
6. CSV Import Mapping Definition	
3. Creating S* Pattern Classes	27-75
1. Feature Framework	
2. Domain Analysis	
3. Interaction Framework	
4. State Machine	
5. Detail Level Requirements	
6. High Level Design	
7. Interface Context	
8. Attribute Couplings	
9. Failure Analysis	

Contents

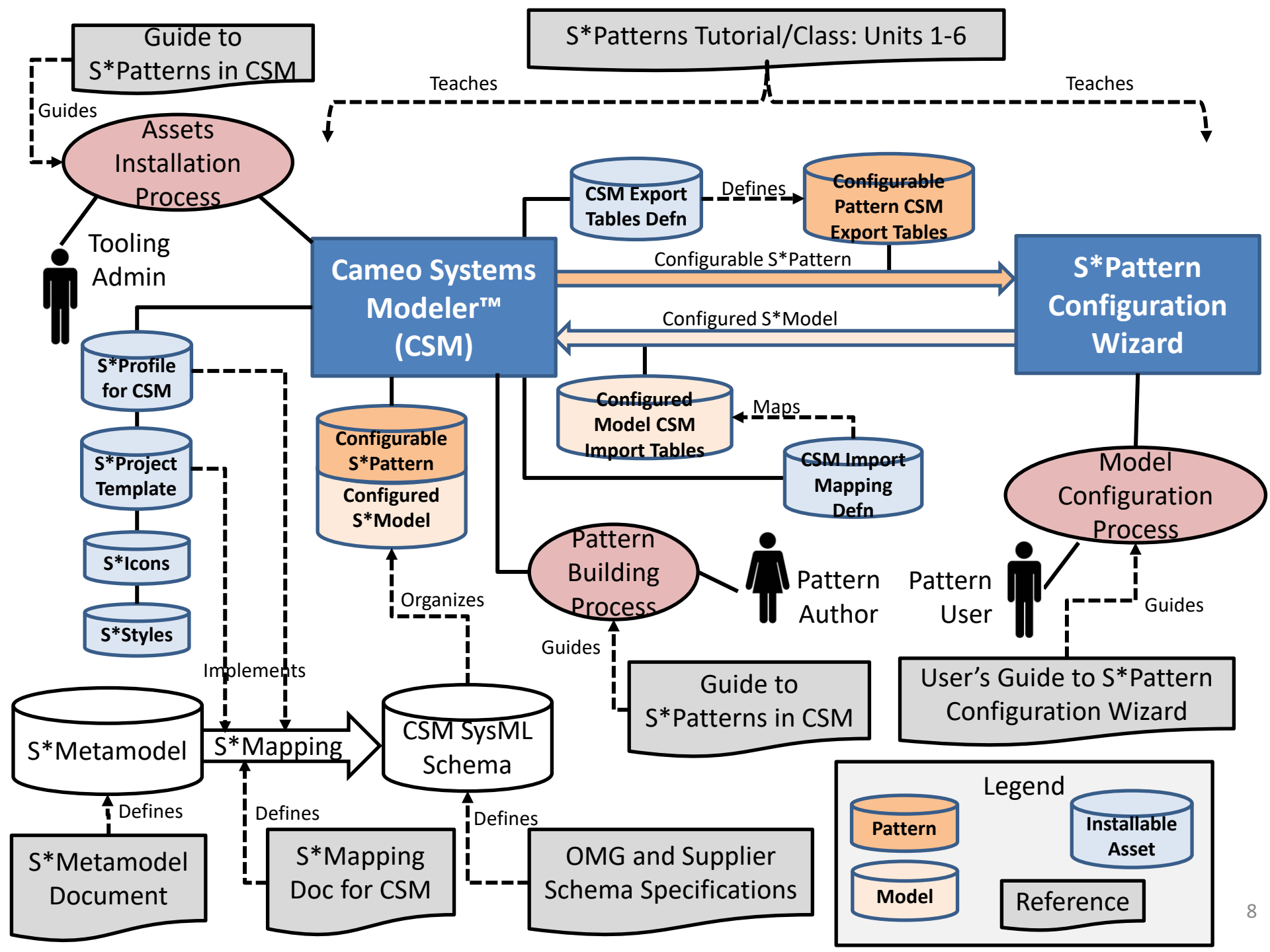
Section	Pages
4. Creating S* Pattern Relationships	76-93
5. Creating S* Pattern Configuration Rules	94-109
6. Exporting and Importing Pattern and Model Data During S* Pattern Configuration	110-114

Change History

Version #	Date	Who	Description
1.3.28	10.19.2023	Stephen Lewis	Inserted updated diagram on slide 7
1.3.29	10.20.2023	Stephen Lewis	Added Change History Slide, Slides 32, 98-99
1.3.30	10.23.2023	Stephen Lewis	Table of Contents and Section descriptions, Slides 35, 71
1.3.31	10.24.2023	Stephen Lewis	Matrix Guidance Slides 69-71
1.3.32	10.25.2023	Stephen Lewis	Slide 6 Diagram legend
1.3.33	10.26.2023	Stephen Lewis	Companion Assets details, Slide 38, 75
1.3.34	10.27.2023	Stephen Lewis	Slides 8, 17, 19-20, additional Section descriptions
1.3.35	10.31.2023	Stephen Lewis	Slides 2-3, 5-9, 17, 23
1.3.36	11.1.2023	Stephen Lewis	Slides 3-4, 9, 24-28,
1.3.37	11.2.2023	Stephen Lewis	Slides 70-2, 74-7, 94-5, 97, 99, 100
1.3.38	11.3.2023	Stephen Lewis	Import/Export Installation Slides
1.3.39	11.6.2023	Stephen Lewis	Import/Export Execution Slides, Role Overview Table

Purpose, scope, companion references, other companion assets

- The purpose of this document is limited to providing detailed directions for (1) installation and use of the S* Profile and S* Project Template, and, (2) with examples, for entry of S*Pattern data into, and specific to, Dassault Cameo Systems Modeler™ (CSM), and (3) tool-specific aspects of pattern and model data export and import between CSM and the Configuration Wizard.
- See the References for other information:
 - To understand the overall significance of that S*Pattern data, refer to the generic S*Patterns class/tutorial reference media.
 - To understand the details of the generic S*Metamodel, refer to the generic S*Metamodel reference.
 - To understand the details of the mapping of the generic S*Metamodel into the CSM-specific SysML schema, refer to the S*Mapping Document specific to CSM.
 - To understand the use of the S*Pattern Configuration Wizard with CSM, refer to the User's Guide to the S*Pattern Configuration Wizard.



Roles Overview

Who	Tool	Task	When
Pattern Author	Cameo Systems Modeler	Pattern Building Process	Pattern creation and updating
Pattern User	Cameo Systems Modeler, Configuration Wizard	Model Configuration Process	For each configuration
Tooling Admin	Cameo Systems Modeler	Assets Installation Process	One-time initial setup

Companion References

1. Tutorial/Class: Advancing the Practice, Units 1-6. Introduction to S*Models, S*Metamodel, S*Patterns, Mappings to Languages & Tools, S*Configuration Wizard". Contact ICTT System Sciences.
2. "Systematica Metamodel", Metamodel Version 8.0, 04/07/2022.
3. "S*Metamodel Mapping for MagicDraw/Cameo Systems Modeler Version 19", Version 1.11.4, 2/13/2023.
4. "Guide to the S*Pattern Configuration Wizard", V1.2.10, 1/18/2023.

*Refer to Figure 1, Page 6

Companion Assets

1. S* Project Template Version 12142023, *Systematica.mdzip*.
2. S* Profile Version 05082023, *Systematica Profile.mdzip*.
3. S* Diagram Styles, *Systematica.stl*.
4. S* Images, *Systematica.zip*.
5. Pattern Export Tables Definition
6. CSV Import Mapping Definition
7. A general example Pattern or client-specific Pattern may be part of the Companion Asset Package, if not starting a new Pattern Project.

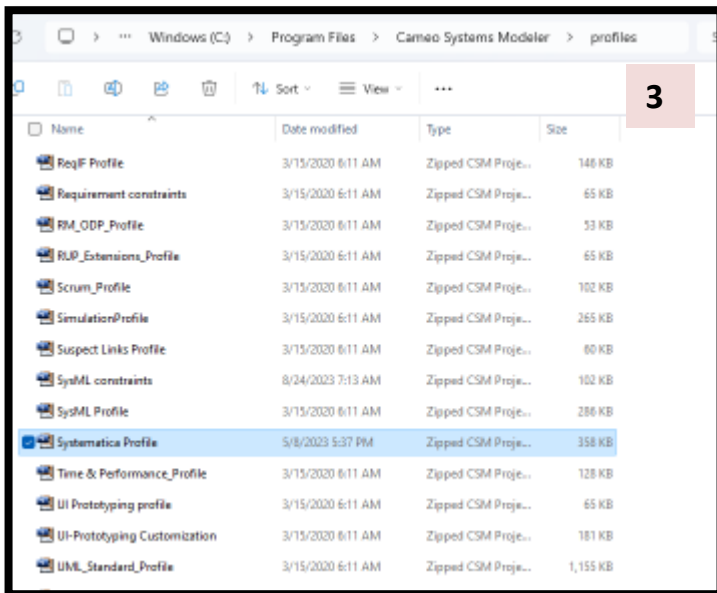
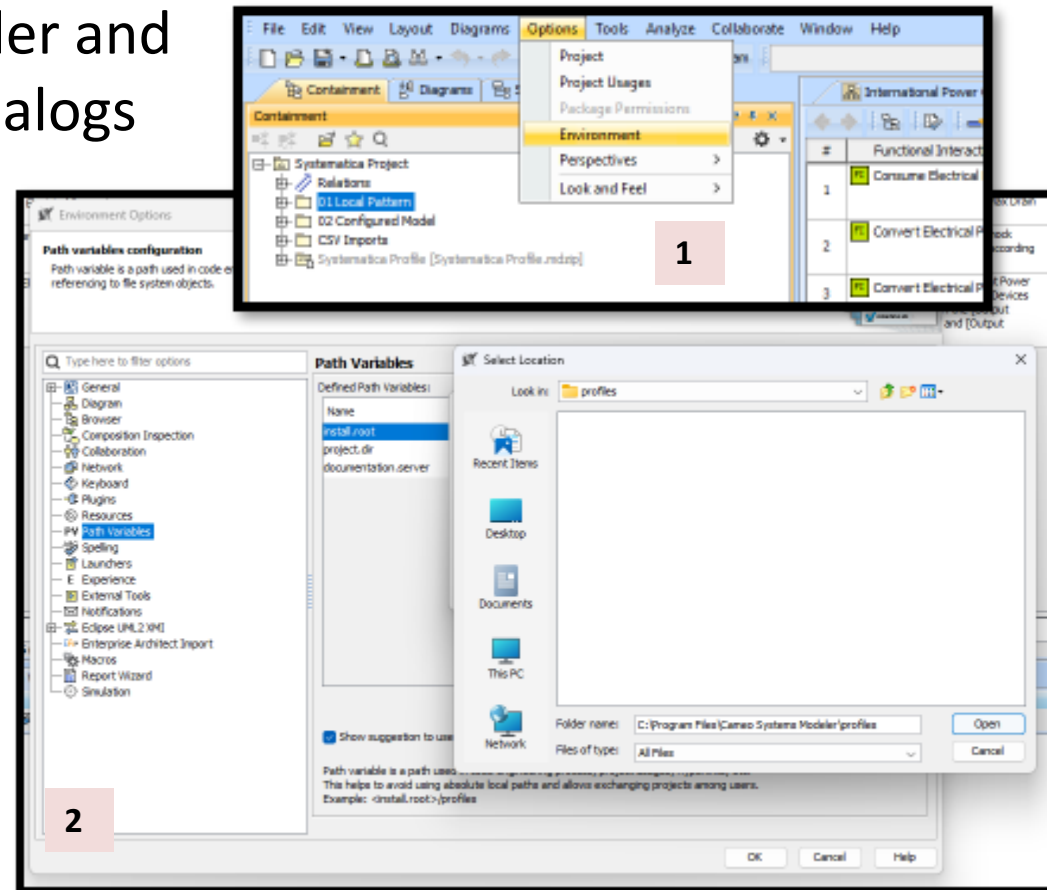
Refer to Figure 1, Page 8

Installing Systematica Profile

- Open Cameo Systems Modeler and use the menu options and dialogs described below.

1. Use Options~Environment~Path Variables to find install.root folder

2. Navigate to the install.root folder using Windows Explorer

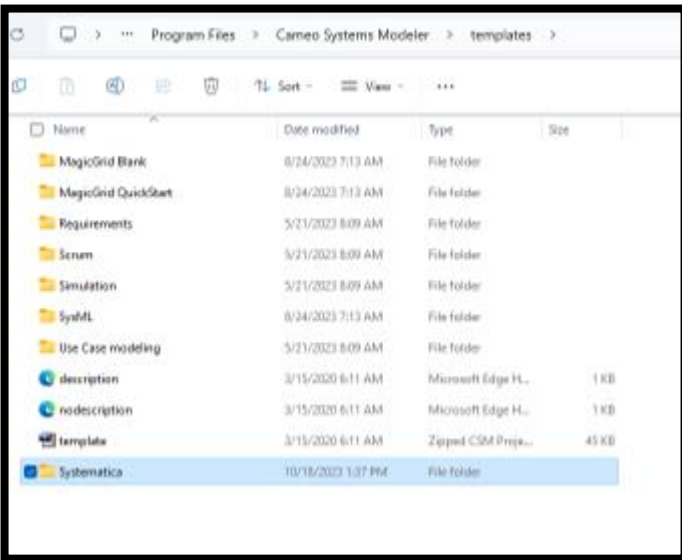
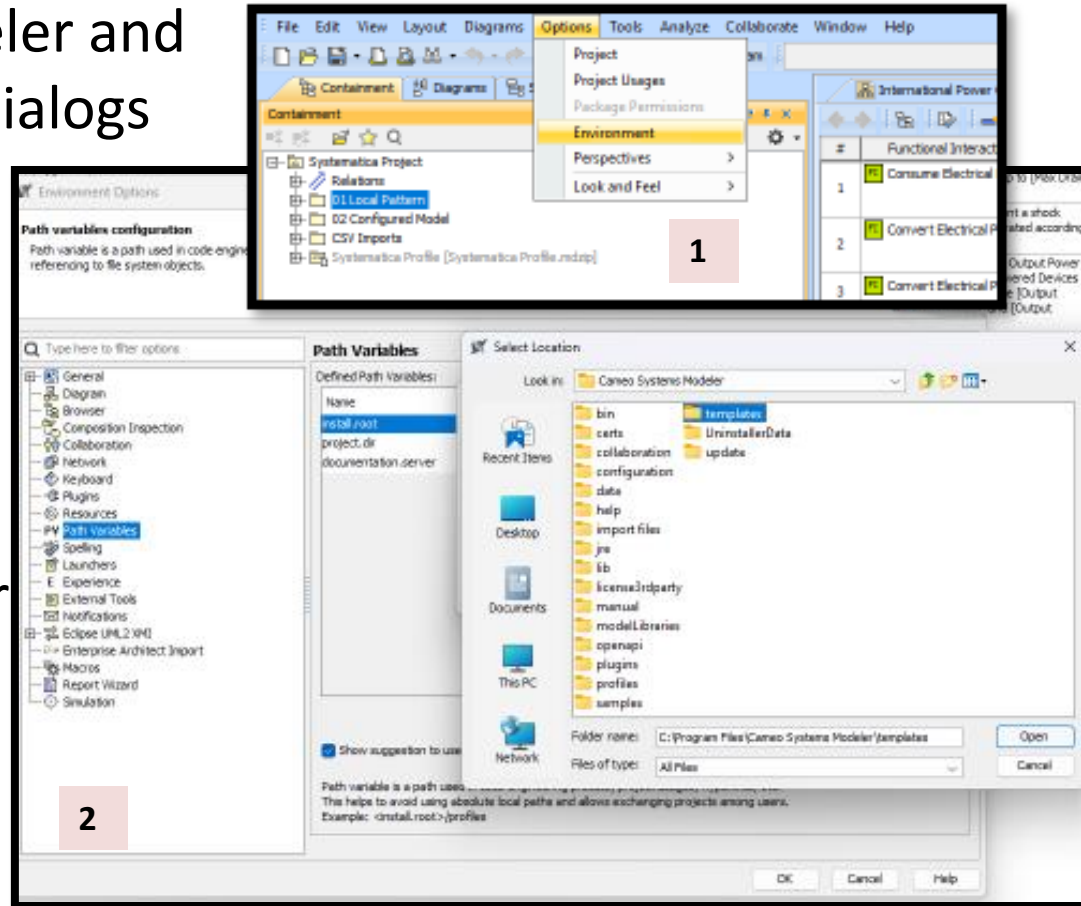


3. Copy Systematica Profile.mdzip to profiles folder from the Deliverables Folder

Installing Systematica Project Template

- Open Cameo Systems Modeler and use the menu options and dialogs described below.

1. Use Options~Environment~Path Variables to find install.root folder
2. Navigate to the install.root folder using Windows Explorer

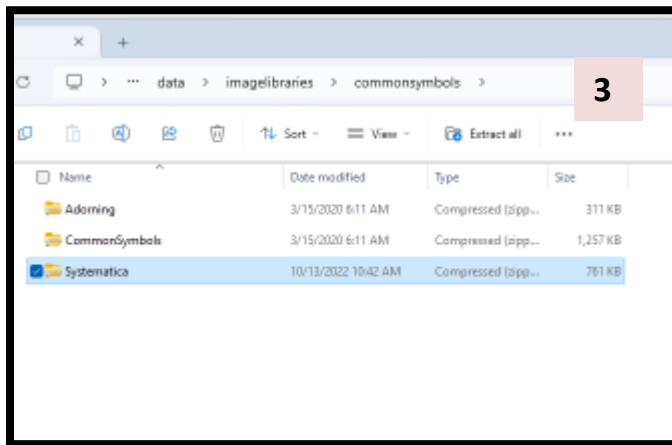
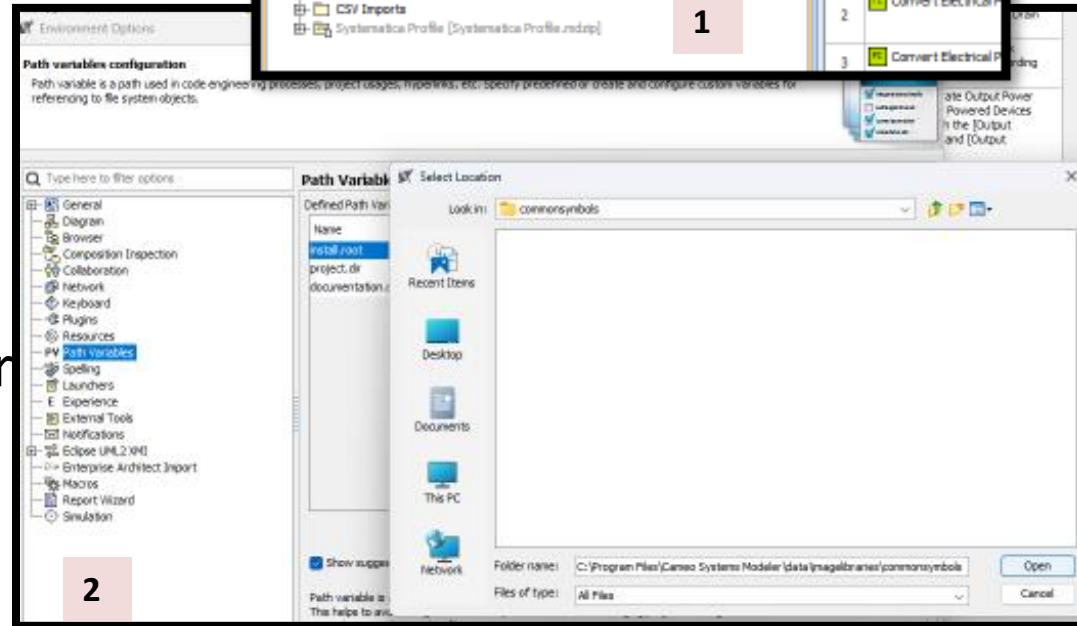
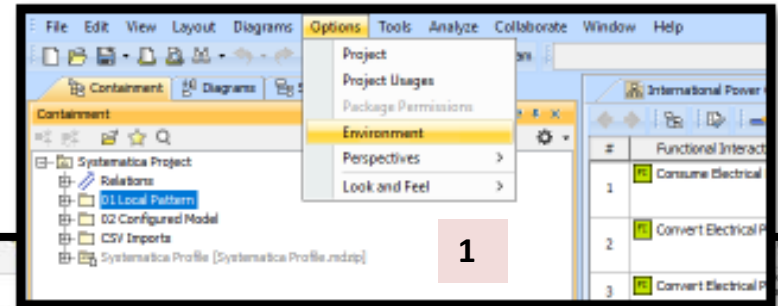


3. Copy Systematica folder to templates folder from the Deliverables Folder

Installing Systematica Images

- Open Cameo Systems Modeler and use the menu options and dialogs described below.

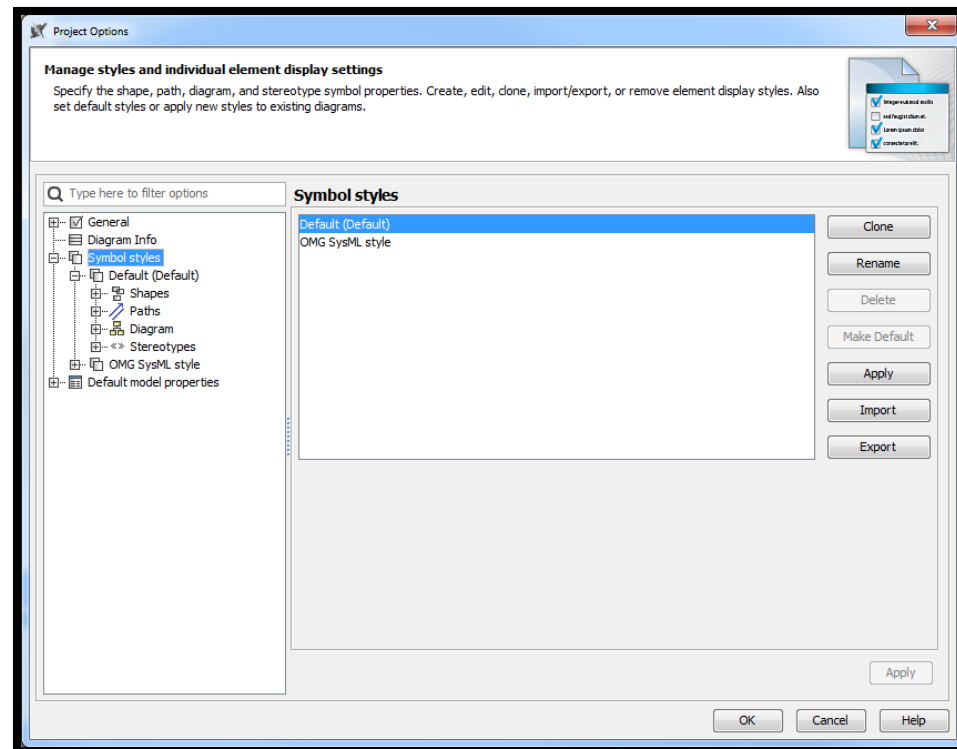
1. Use Options~Environment~Path Variables to find install.root folder
2. Navigate to the install.root folder using Windows Explorer



3. Copy Systematica zip file to data\imagelibraries\commonsymbols folder from the Deliverables Folder

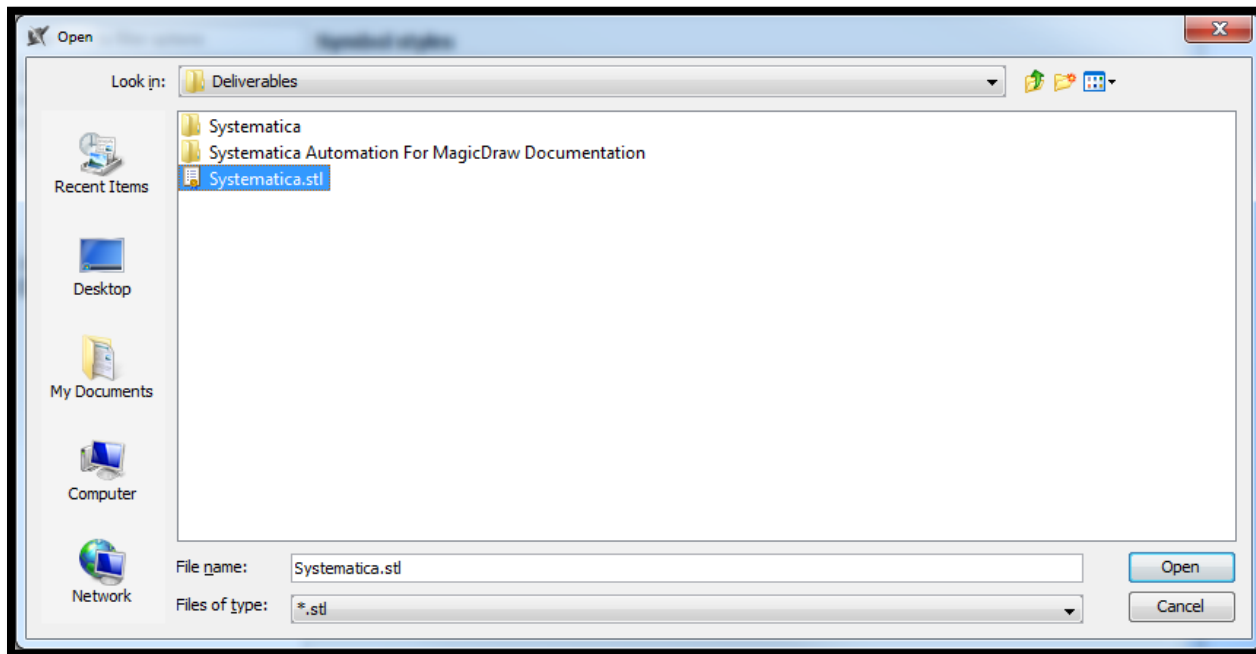
Installing Systematica Diagram Styles

- Open a project in MagicDraw
- Select the menu option ~Options~Project
- Click on ‘Symbol styles’ in tree in left pane.



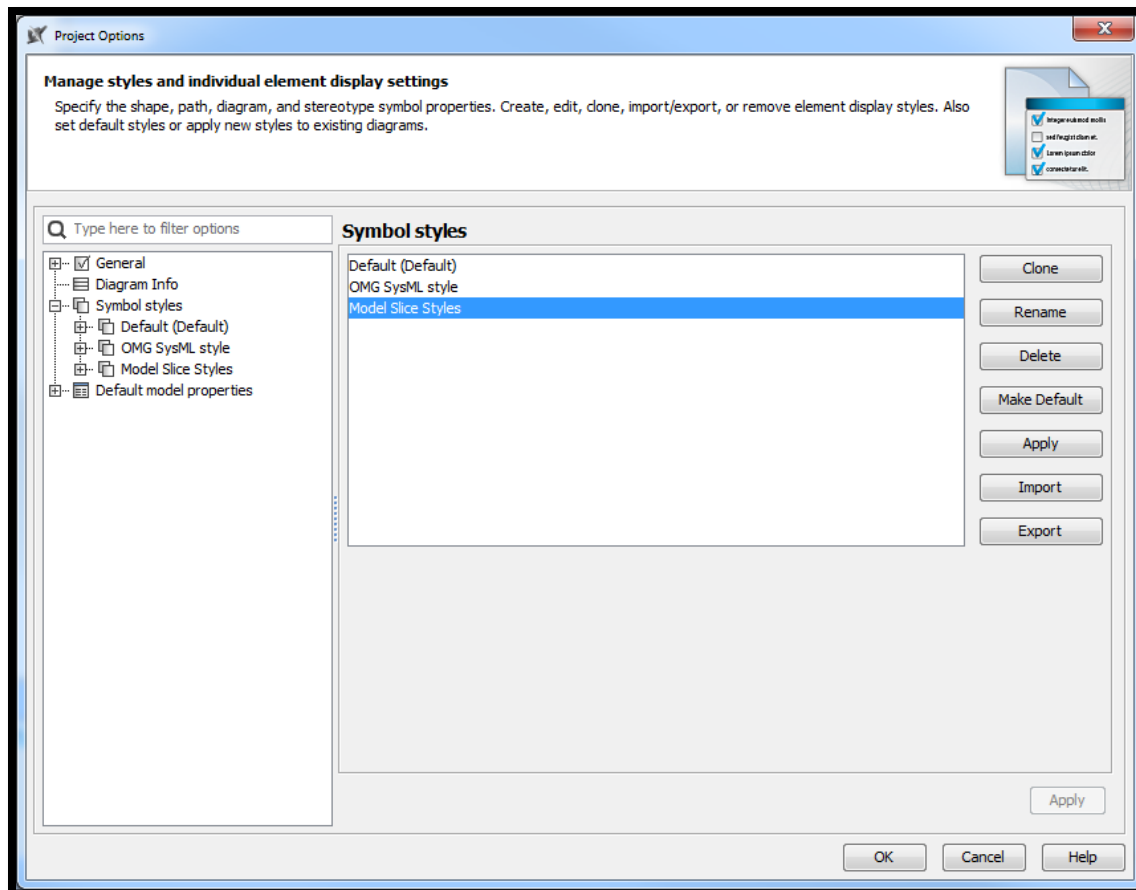
Installing Systematica Diagram Styles, cont'd

- Click the 'Import' button
- Navigate to the Deliverables Folder (see image)
- Select the Systematica.stl file (see image)
- Click the 'Open' button (see image)



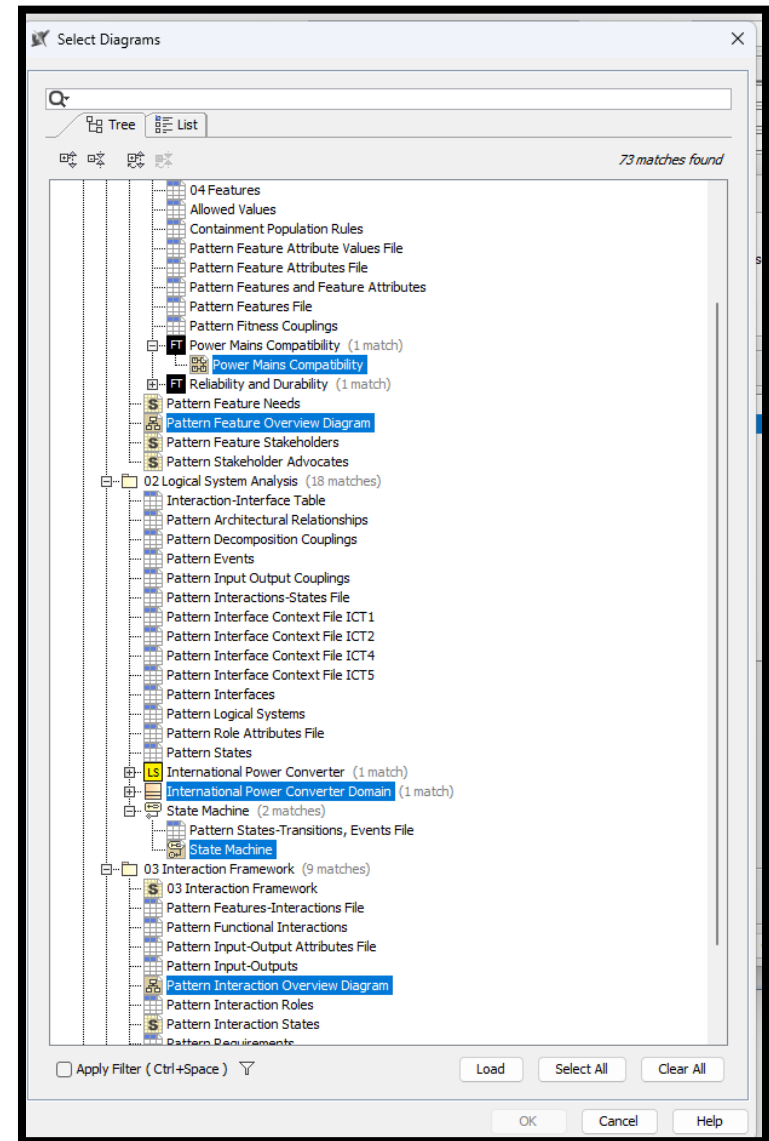
Installing Systematica Diagram Styles, cont'd

- Expand the 'Symbol styles' node in tree in the left pane.



Installing Diagram Systematica Styles, cont'd

- Select the 'Systematica Styles' node in tree in the left pane.
- Click the 'Apply' button
- Select which diagrams to apply style to (i.e., Domain Diagram, Pattern Feature Overview Diagram, Interaction Overview Diagram, State Machine, Attribute Coupling, and Slice Diagrams)
- Click the 'OK' button



Defining Pattern Export Tables Using the Report Wizard

Use the Report Wizard to create the report of Pattern Tables.

Select a report template

Select a report template from which you would like to generate a report. In this dialog, you can also create new templates, edit, delete, open, clone, or import/export existing templates. Additionally, you can "Attach" a template to your project so that it can be viewed in the model containment tree. To edit an attached template, select one and click "Extract". This will create a saved copy in your local reports directory.

Select Template

- Dependency Report
- Diagram Presentation
- Diagram Presentation PowerPoint
- Element Summary Table
- Element Table
- Package Element Table
- Project Element Article
- Table Report
- Use Case Diagram Presentation
- Requirements

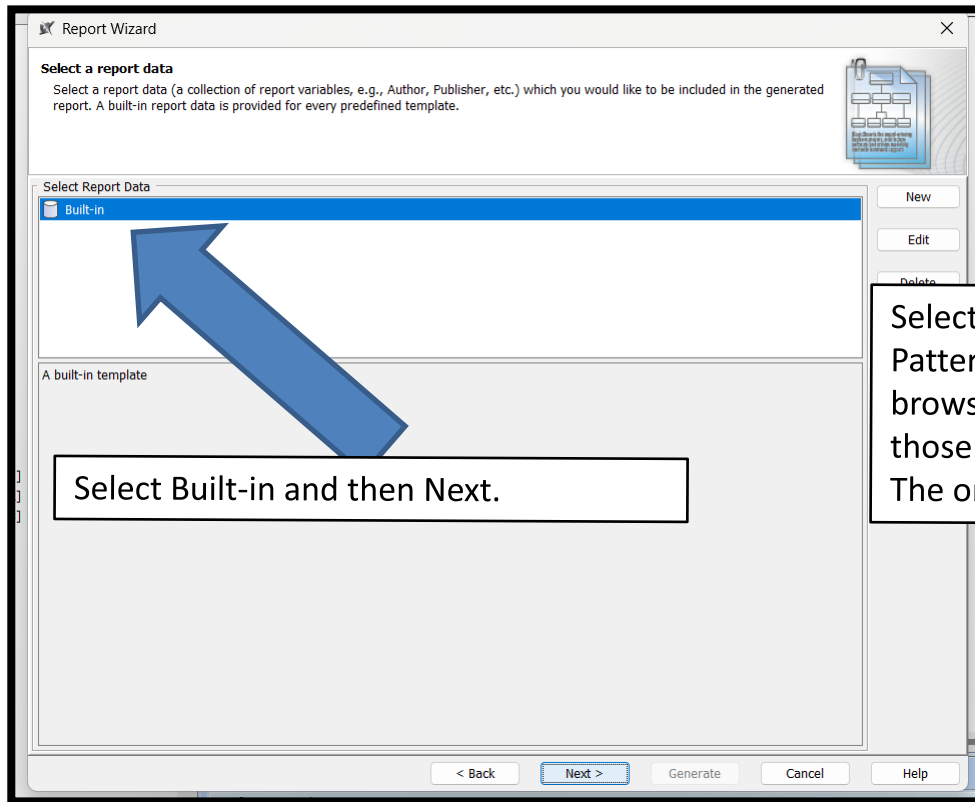
The excel spreadsheet is generated from diagram table

Next >

Report Wizard

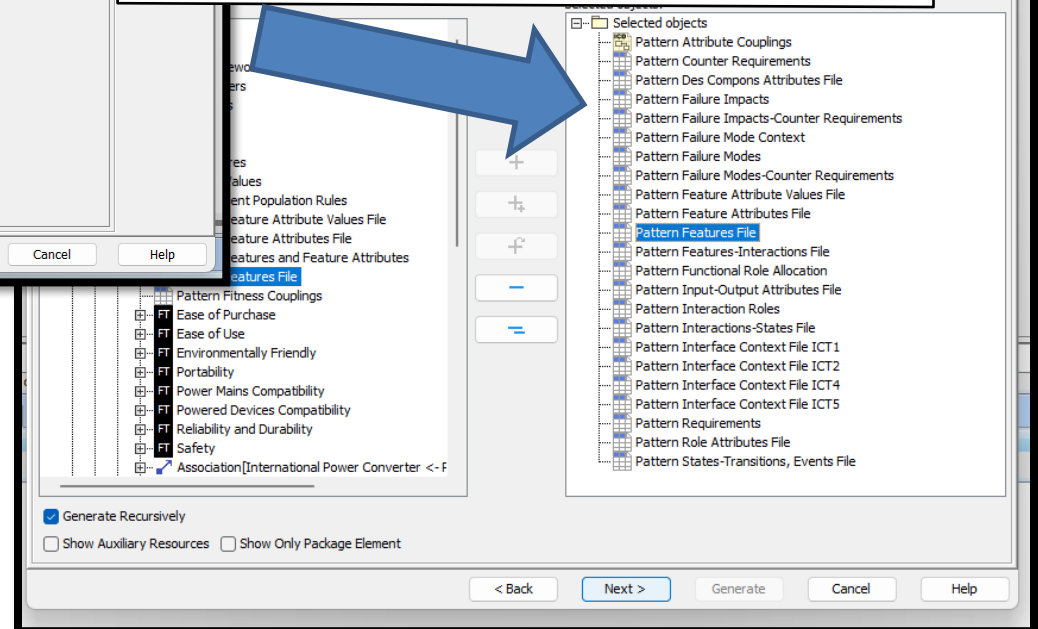
Buttons: New, Edit, Delete, Open, Variable, Clone, Attach, Import, Export, < Back, Next >, Generate, Cancel, Help

Defining Pattern Export Tables Using the Report Wizard



Select Built-in and then Next.

Select this list of Pattern Tables from the 01 Local Pattern Package. See next slides for Table list and browser locations. Ensure the table names match those in the list exactly. There are 23 tables to select. The order of selection does not matter.



**Note: If only Packages appear in the list, select the Search For: Any Elements radio button option.

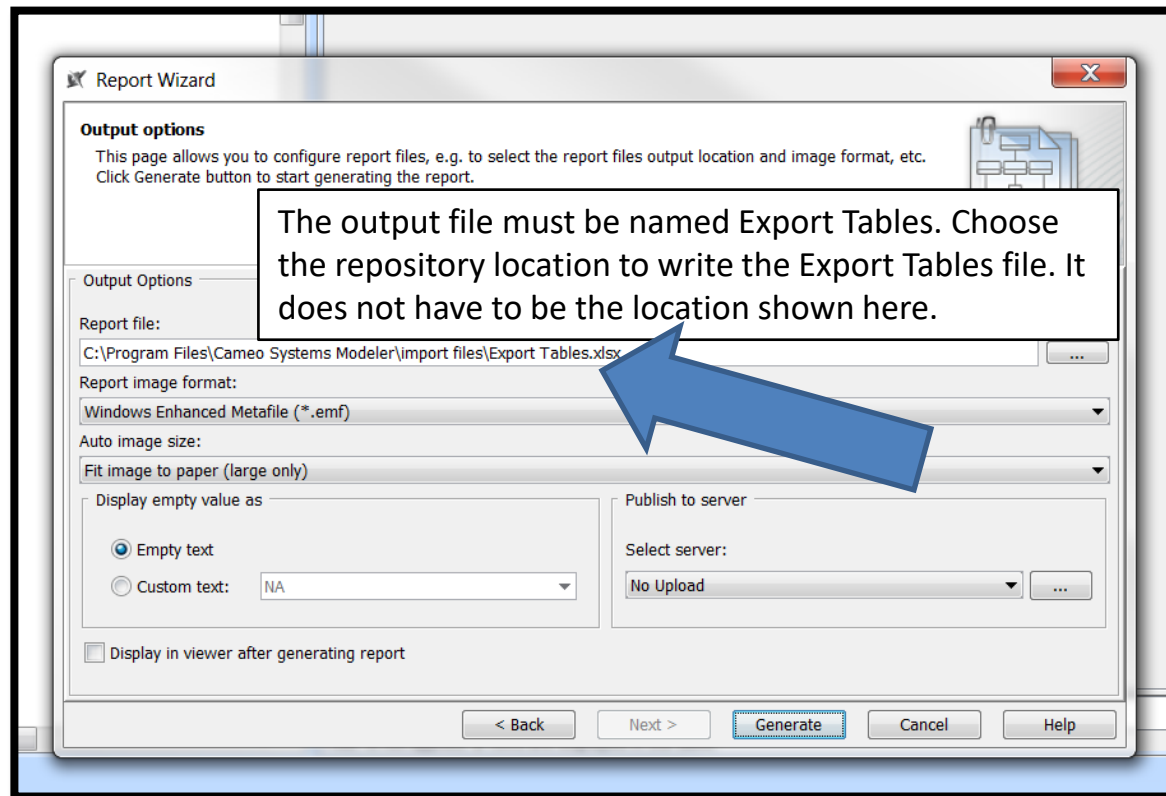
Defining Pattern Export Tables Using the Report Wizard

Table Name	Table Browser Location
Pattern Features File	01 Local Pattern>01 Feature Framework>04 Features
Pattern Feature Attributes File	01 Local Pattern>01 Feature Framework>04 Features
Pattern Feature Attribute Values File	01 Local Pattern>01 Feature Framework>04 Features
Pattern Role Attributes File	01 Local Pattern>02 Logical System Analysis
Pattern Interactions-States File	01 Local Pattern>02 Logical System Analysis
Pattern States-Transitions, Events File	01 Local Pattern>02 Logical System Analysis> State Machine
Pattern Interface Context File ICT1	01 Local Pattern>02 Logical System Analysis
Pattern Interface Context File ICT2	01 Local Pattern>02 Logical System Analysis
Pattern Interface Context File ICT4	01 Local Pattern>02 Logical System Analysis
Pattern Interface Context File ICT5	01 Local Pattern>02 Logical System Analysis
Pattern Interaction Roles	01 Local Pattern>03 Interaction Framework
Pattern Requirements	01 Local Pattern>03 Interaction Framework
Pattern Features-Interactions File	01 Local Pattern>03 Interaction Framework
Pattern Input-Output Attributes File	01 Local Pattern>03 Interaction Framework

Defining Pattern Export Tables Using the Report Wizard

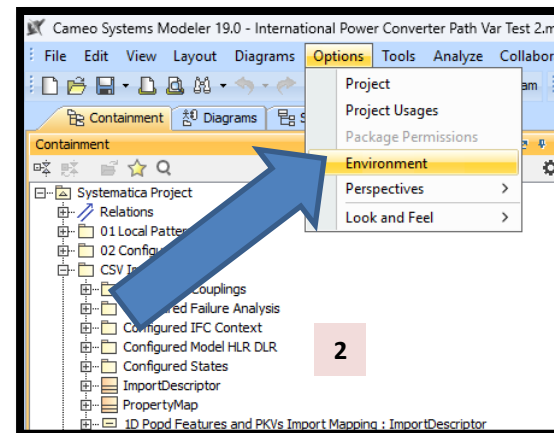
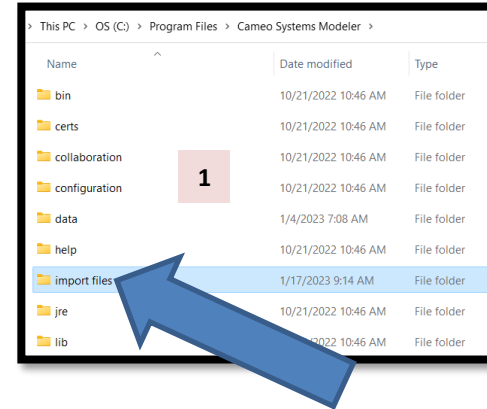
Table Name	Table Browser Location
Pattern Functional Role Allocation	01 Local Pattern>04 Physical Architecture
Pattern Des Compons Attributes File	01 Local Pattern>04 Physical Architecture
Pattern Failure Impacts	01 Local Pattern>05 Risk Framework
Pattern Failure Modes	01 Local Pattern>05 Risk Framework
Pattern Counter Requirements	01 Local Pattern>05 Risk Framework
Pattern Failure Mode Context	01 Local Pattern>05 Risk Framework
Pattern Failure Modes-Counter Requirements	01 Local Pattern>05 Risk Framework
Pattern Failure Impacts-Counter Requirements	01 Local Pattern>05 Risk Framework
Pattern Attribute Couplings	01 Local Pattern

Defining Pattern Export Tables Using the Report Wizard



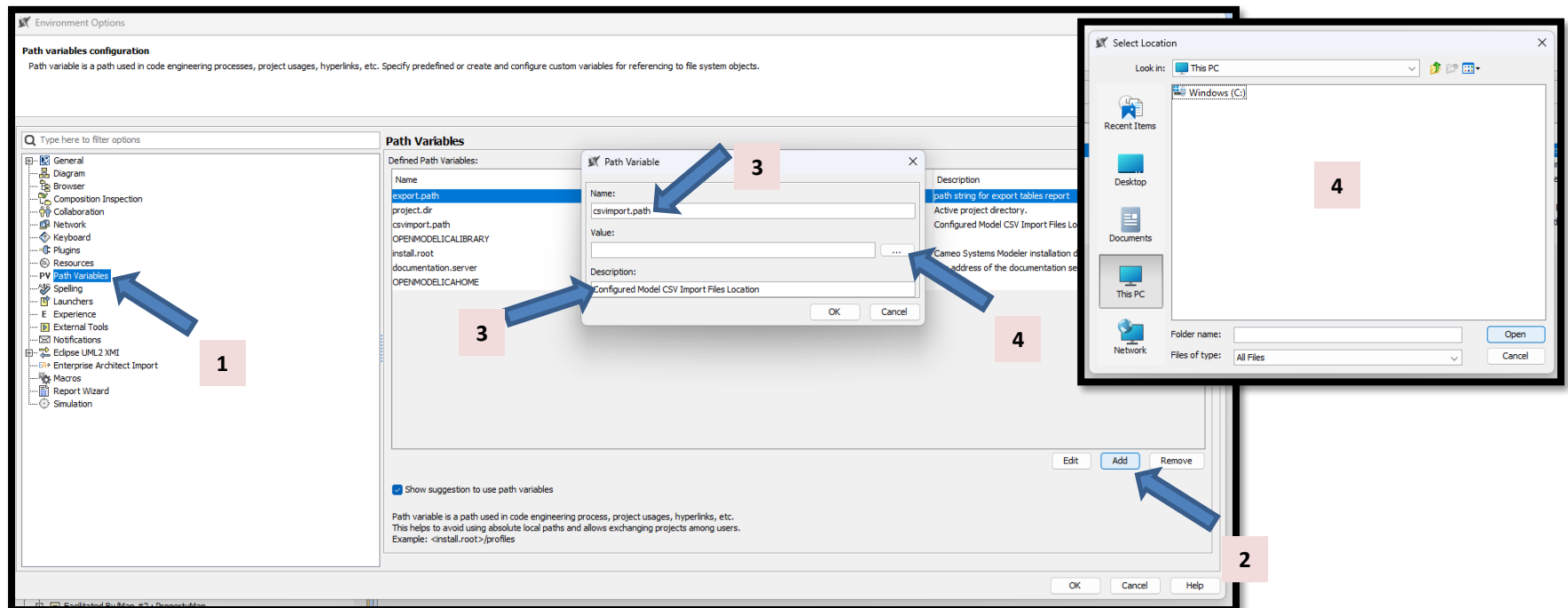
CSV Import Mapping Definition for Configured Model Data

1. In Windows Explorer, create a folder in an accessible location. This is the same location as specified using Button 5 in the Configuration Wizard to store the configured model output files. This folder location and name can be of your creation and choosing as long as read/write privileges are available.
2. In Cameo Systems Modeler, from the Menu toolbar, select Options → Environment.



CSV Import Mapping Definition for Configured Model Data

1. Select Path Variables from the list of Environment Options.
2. Select Add to open the Path Variable Dialog Window.
3. Enter “csvimport.path” as the Name and “Configured Model CSV Import Files Location” as the Description.
4. Select the ellipses to open the location dialog and choose the accessible file location discussed in the previous slide.

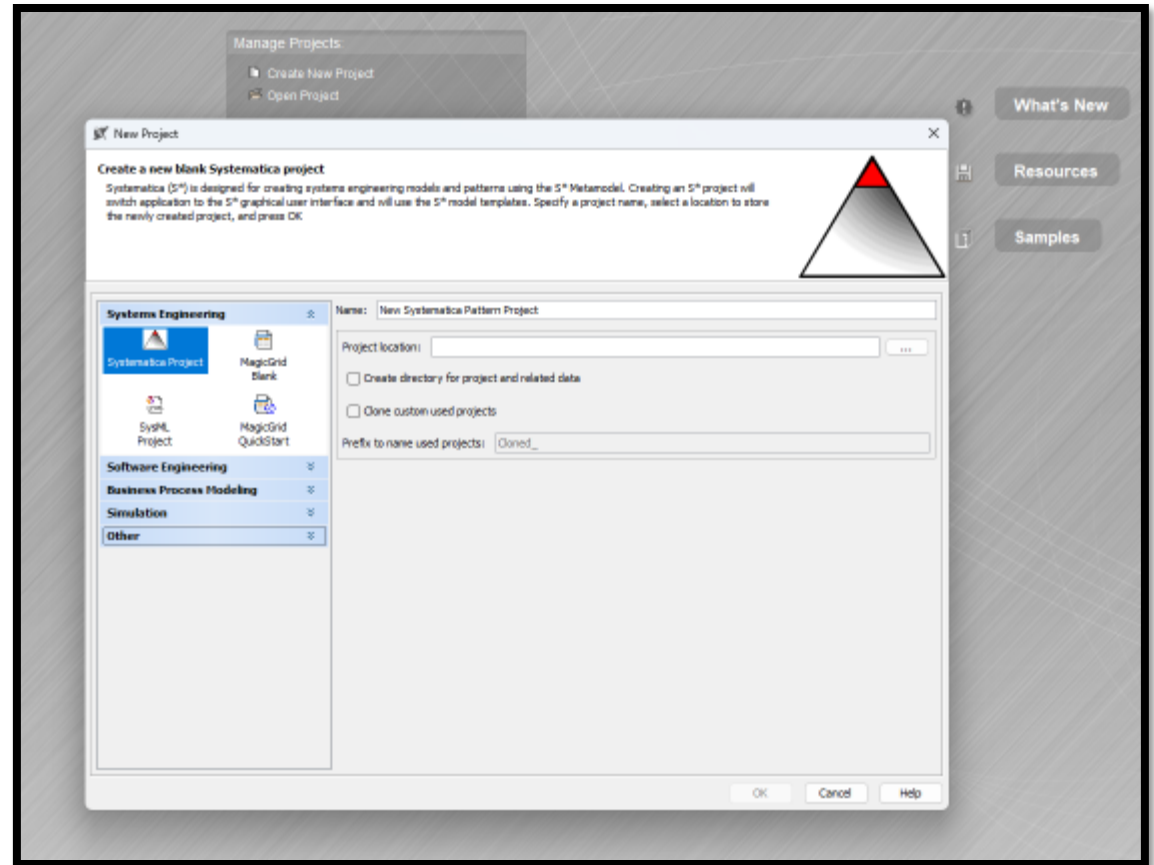


3. Creating S* Pattern Classes

This section of the guide details the steps for creating formally modeled S* Pattern Classes (metaclasses) as mapped and implemented in Cameo Systems Modeler.

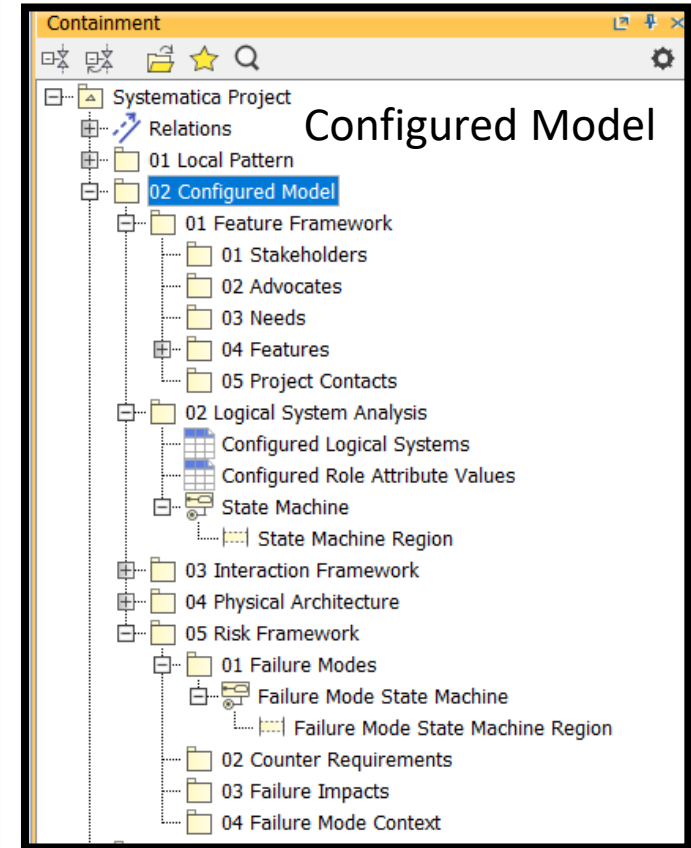
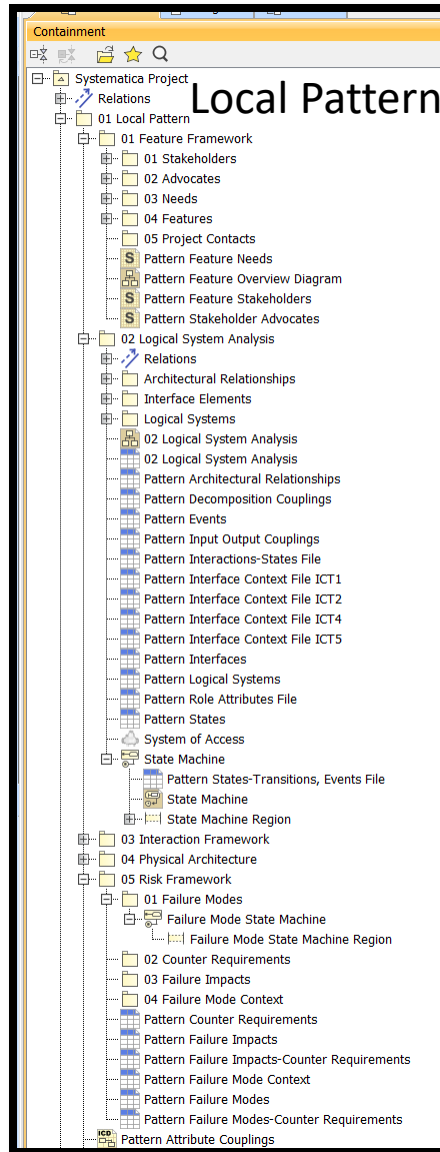
Creating a New S* Pattern Project

- Start CSM and Select Create New Project.
- Select the Systematica Project.
- Name the Project and Select a File Location.



Systematica Project Template Packages

- S* Projects include 2 main packages
 - Local Pattern: An entire configurable model from which a Configured Model is configured
 - Configured Model: An entire model that is a specific configuration of the Local Pattern
- All of a pattern's classes and relationships should be located within the Local Pattern package
- Package Numbers/Names are pre-defined in the Project Template and Pre-Existing Pattern Files and should not be changed.

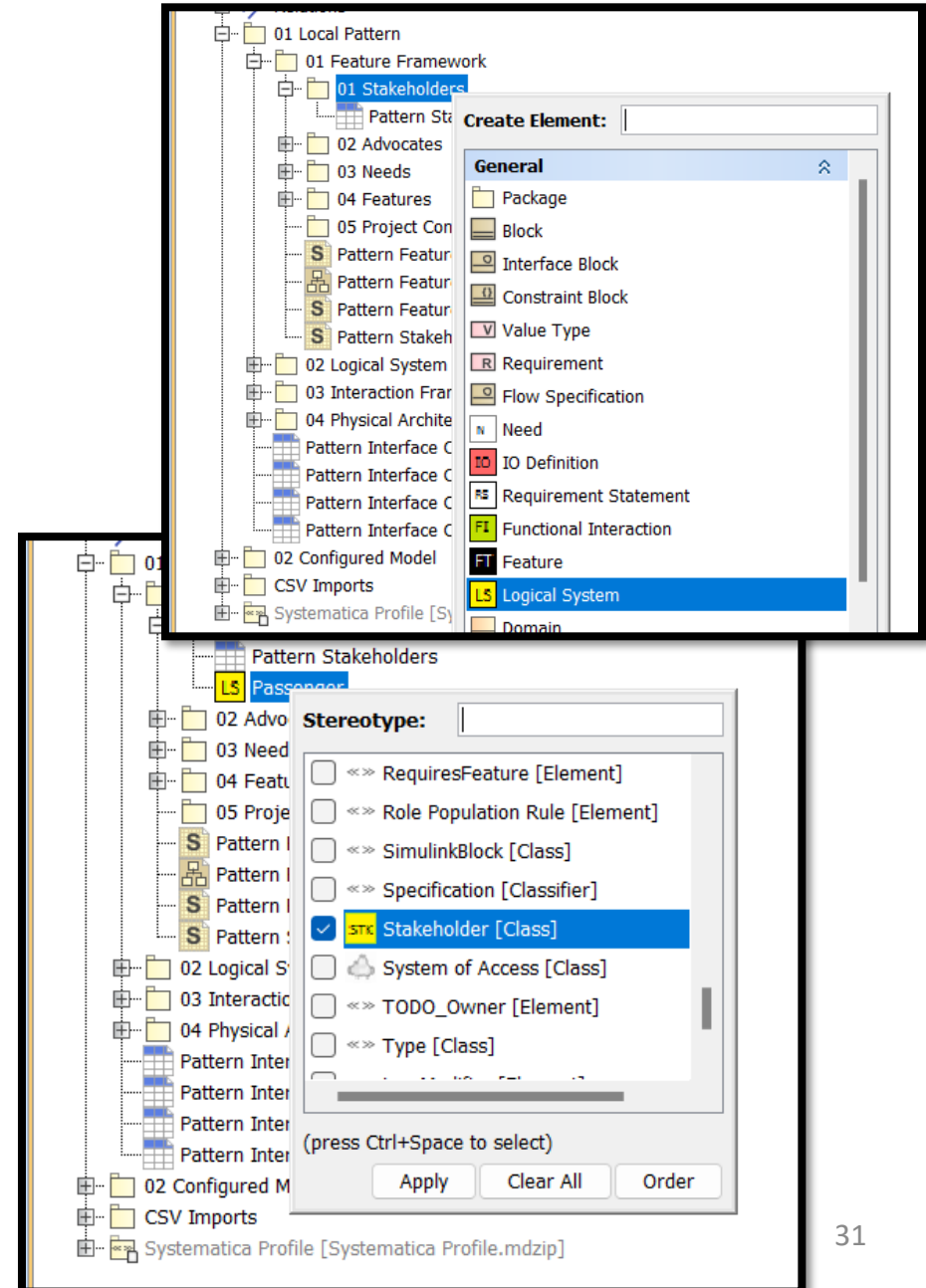


Pattern data entry methods and details

- MBSE tools provide multiple means of entering the same information, in different modes.
- This document focuses on pattern data entry via tool browser GUI and tabular entries.
- Additional methods of entry for the same pattern data include diagram entry and bulk import, not discussed in this document.
- Class Names should not include special characters such as []/\-”;
- Use of tilde (~) symbol on following pages indicates use of a mouse right-click or equivalent.

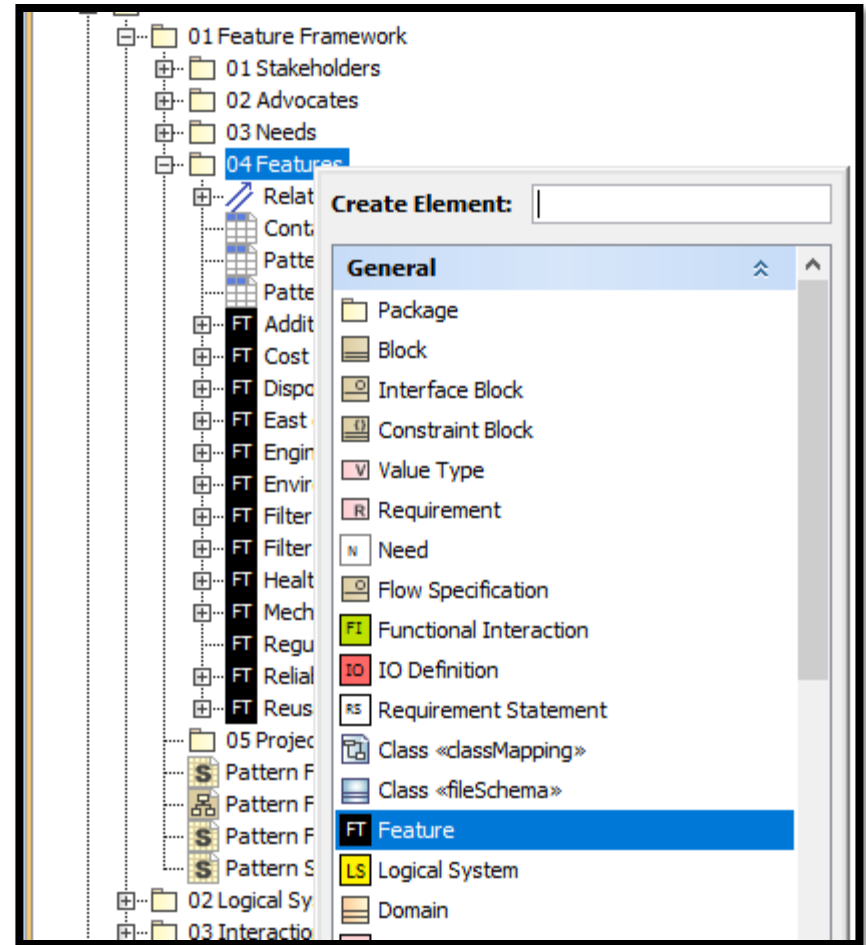
Feature Framework: Creating Stakeholders

- Pattern Stakeholders are created in package 01 Local Pattern::01 Feature Framework::01 Stakeholders
- Use ~Create Element and choose Logical System because Stakeholder is a type of Logical System.
- Right-click on the newly created Logical System, select Stereotype and change the Stereotype to Stakeholder and remove the Logical System Stereotype.



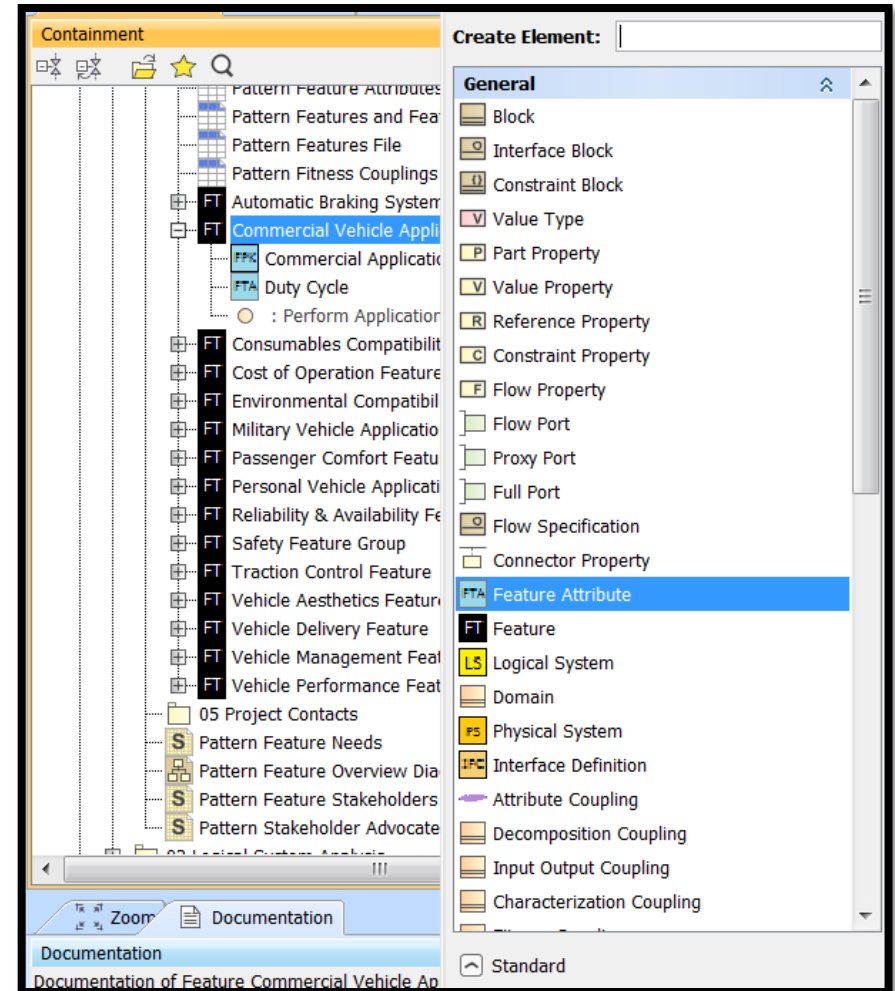
Feature Framework: Creating Features

- Pattern Features are created in package 01 Local Pattern::01 Feature Framework::04 Features
- Use ~Create Element and choose Feature to create an S* Feature



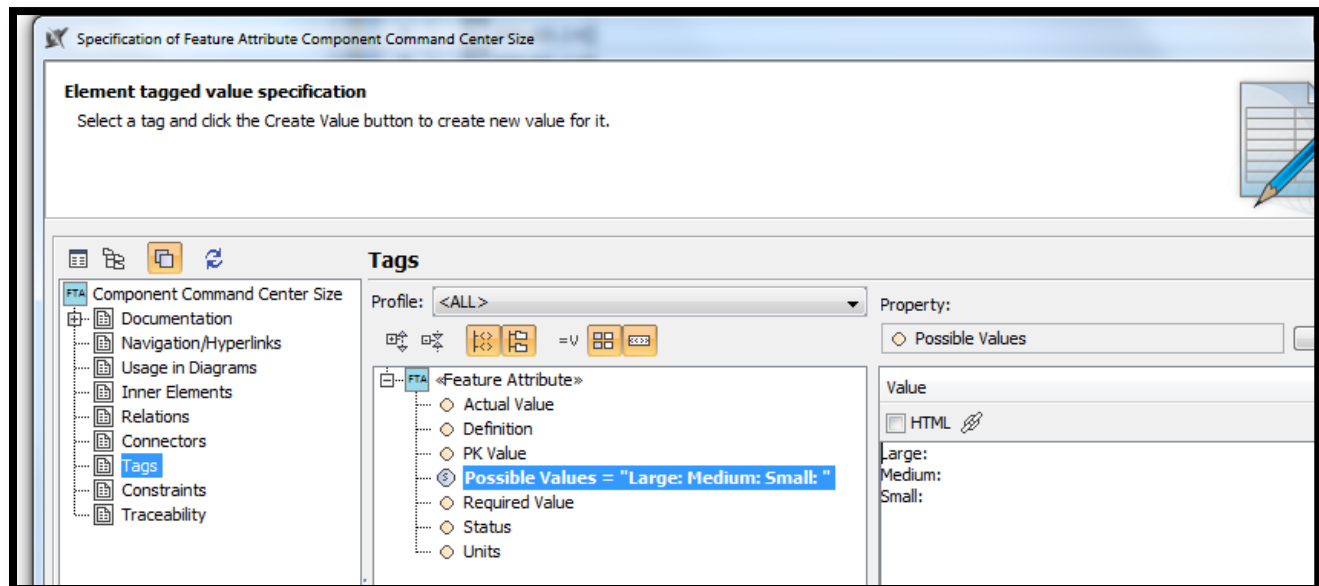
Feature Framework: Creating Feature Attributes

- Features may have Feature Attributes. A special type of Feature Attribute is called a Feature Primary Key, which is the Feature Attribute that is used to differentiate Configured Features originating from the same parent Feature in the Pattern.
- Select a Feature and use ~Create Element and choose either Feature Attribute or Feature Primary Key
- If adding a Feature Primary Key, enter the name of the Feature Primary Key into its Feature's Primary Key Name tag.



Feature Framework: Feature Attribute Possible Values


- Use the new Feature Attribute's specification to define Possible Value options in the Possible Values tag.
- Each option should:
 - Be on its own line
 - Be followed by a colon and a space
 - May have a value meaning after the colon and space




Feature Framework: Feature Attribute Possible Values

- The list of colon separated Feature Attributes can also be entered directly into the Possible Values column of the Pattern Feature Attributes Table.

The screenshot displays a software interface with a table of Feature Attributes. The table has five columns: #, Feature, Attribute Name, Applied Stereotype, and Possible Values. A dialog box titled 'Possible Values' is open, showing a text area with the following content:

HTML 
Power Output 1:
Power Output 2:
Power Output 3: |

#	Feature	Attribute Name	Applied Stereotype	Possible Values
1	FT Reliability and Durability	Design Life	FTA Feature Attribute [Property]	
2	FT Power Mains Compatibility	Power Mains Type	FTA Feature Attribute [Property]	
3	FT Powered Devices Compatibility	Power Output Capacity	FTA Feature Attribute [Property]	
4	FT Powered Devices Compatibility	Power Output Interface ID	FRK Feature Primary Key [Property]	Power Output 1: Power Output 2: Power Output 3: 
5	FT Safety	Safety Risk Type	FRK Feature Primary Key [Property]	Electrical Shock:
6	FT Power Mains Compatibility	Max Drain on Mains	FTA Feature Attribute [Property]	

Feature Framework: Feature Attribute Allowed Values

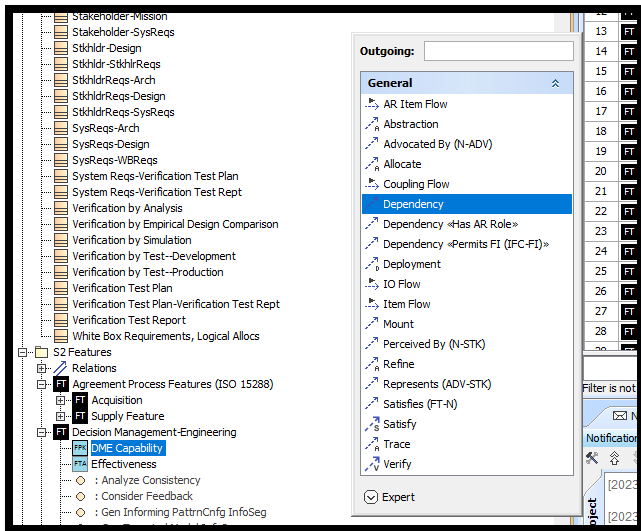
- An alternative method for creating Feature Attributes and connecting them to Features for configuration includes the use of allowed values.
- Enter the complete list of Feature Attribute Values in the Allowed Values Table.

The screenshot displays a software interface with a project tree on the left and a table of allowed values on the right. The project tree shows a hierarchy starting with 'Systematica Project' and 'Relations', leading to '01 Local Pattern' and '04 Features'. Under '04 Features', there is a sub-folder 'Feature Attribute Values', which contains 'S2 Features' and 'S3 Features'. The 'S3 Features' folder is expanded, and 'Allowed Values' is selected. The table on the right, titled 'Criteria', has columns for '#', 'Name', and 'Scope (optional)'. It lists 28 items, each with a small icon and a name.

#	Name	Scope (optional)
1	Alternatives, Trades, Selection	
2	Basic Integration	
3	Basic Stakeholder Validation	
4	Black Box Reqs Consistent	
5	Component Capabilities, Margins	
6	Components and Alternatives	
7	Design Feedback to Reqs	
8	Development Unit Test	
9	Enterprise Unit Assignment	
10	Experience Pattern Informed, Configurable	
11	Experience Pattern Informed, Personal	
12	Experience Pattern Informed, Standard	
13	Experience Pptrn, Config-Mission	
14	Experience Pptrn, Personal-Mission	
15	Experience Pptrn, Standard-Mission	
16	Facilities Construction	
17	Feasibility, Fault, Risk Analysis	
18	Flow Down Requirements Informed	
19	Hardware Fabrication	
20	Hardware Prototyping	
21	Identification and Authorization	
22	Implementation Report	
23	Information Navigation and Access	
24	Integration Plan and Tooling	
25	Integration Report	
26	Logical Architecture & Alternatives	
27	Materials Management	
28	MBSE Black Box Capability	

Feature Framework: Feature Attribute Allowed Values

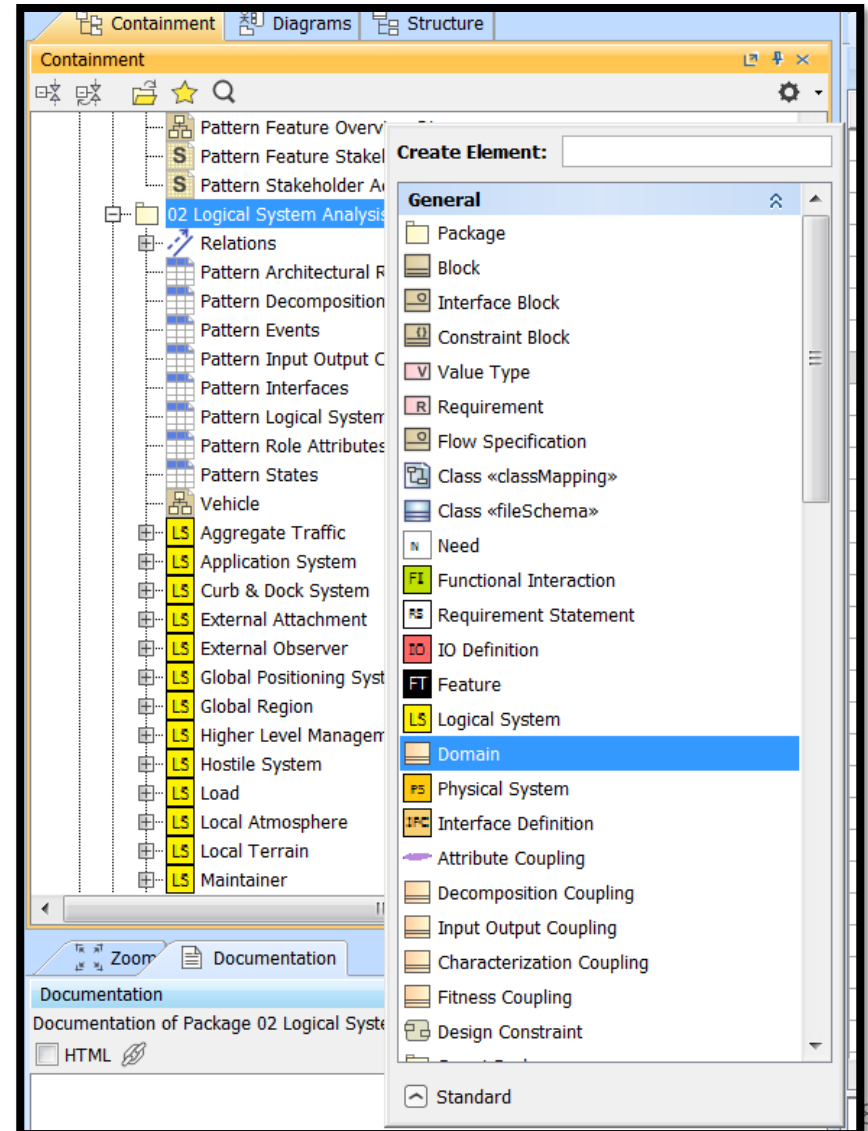
- Create a Dependency relationship between the Feature Primary Key and Feature Attribute Value.
- Set the Stereotype of the relationship to Can Have Value.
- View the result in the Pattern Feature Attribute Values Table.



#	Feature	Feature Primary Key	Allowed Value
1	Architectural Definition	Architectural Definition Capability	Alternatives, Trades, Selection
2	Architectural Definition	Architectural Definition Capability	Black Box Reqs Consistent
3	Architectural Definition	Architectural Definition Capability	Experience Pattern Informed, Configurable
4	Architectural Definition	Architectural Definition Capability	Experience Pattern Informed, Personal
5	Architectural Definition	Architectural Definition Capability	Experience Pattern Informed, Standard
6	Architectural Definition	Architectural Definition Capability	Logical Architecture & Alternatives
7	Architectural Definition	Architectural Definition Capability	Observation Informed
8	Architectural Definition	Architectural Definition Capability	Stakeholder Constrained
			White Box Requirements, Logical Allocs
			Experience Pattern Informed, Configurable
			Experience Pattern Informed, Personal
			Experience Pattern Informed, Standard
			Simulation Informed
			Stakeholder Informed
			Experience Pttm, Config-Mission
			Experience Pttm, Personal-Mission
			Experience Pttm, Standard-Mission
			Observation-Mission
			Simulation-Mission
			Stakeholder-Mission
			ComparDesign-Design
			Constraints-Arch
			DesAnalysis-SysReqs
			Experience Pttm, Config-Arch
			Experience Pttm, Config-Design
			Experience Pttm, Config-StkhdrReqs
			Experience Pttm, Config-SysReqs

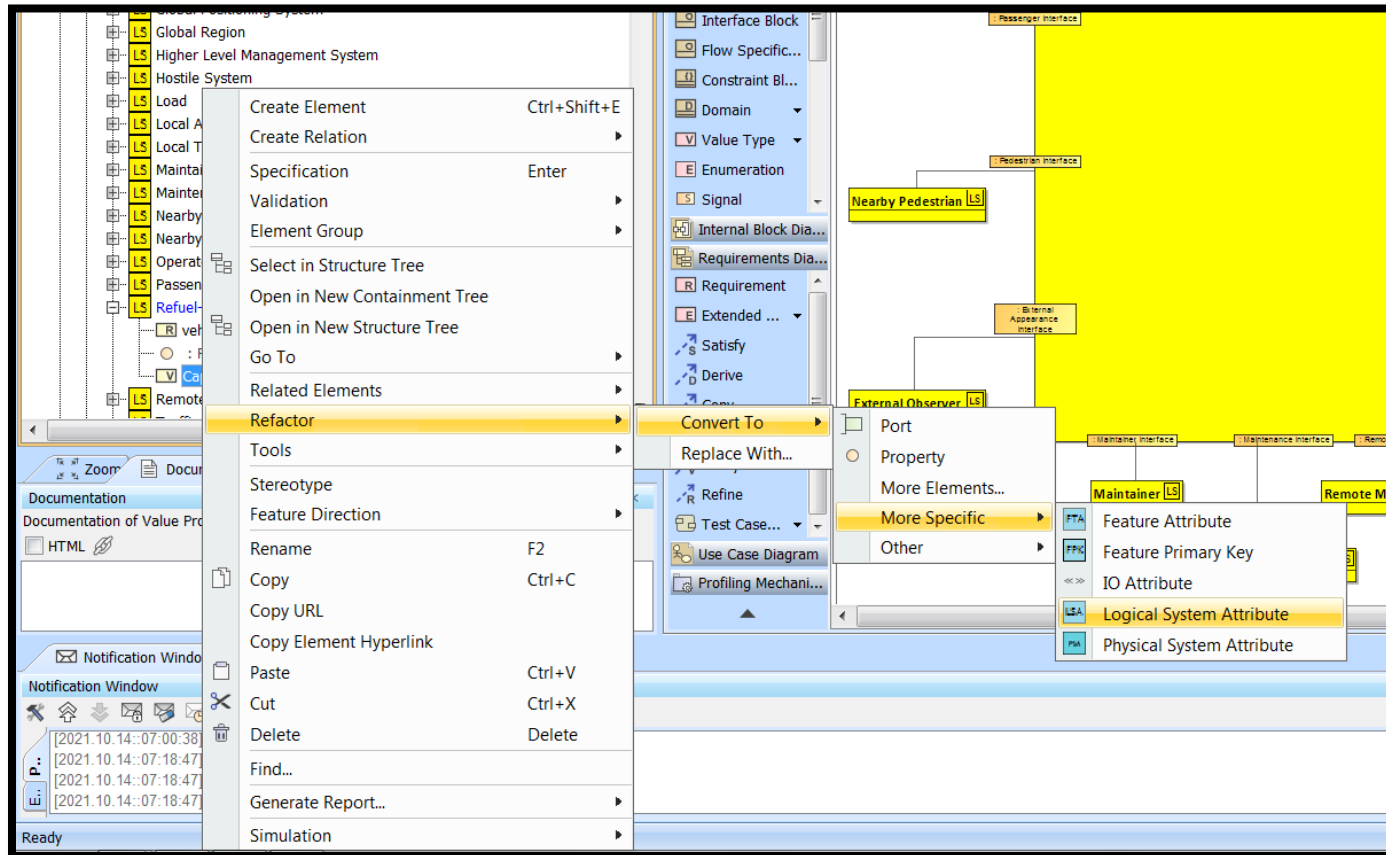
Domain Analysis: Creating Domains and Logical Systems

- Pattern Domains and Logical Systems are created in package 01 Local Pattern::02 Logical System Analysis
- Use ~Create Element and choose Domain or Logical System



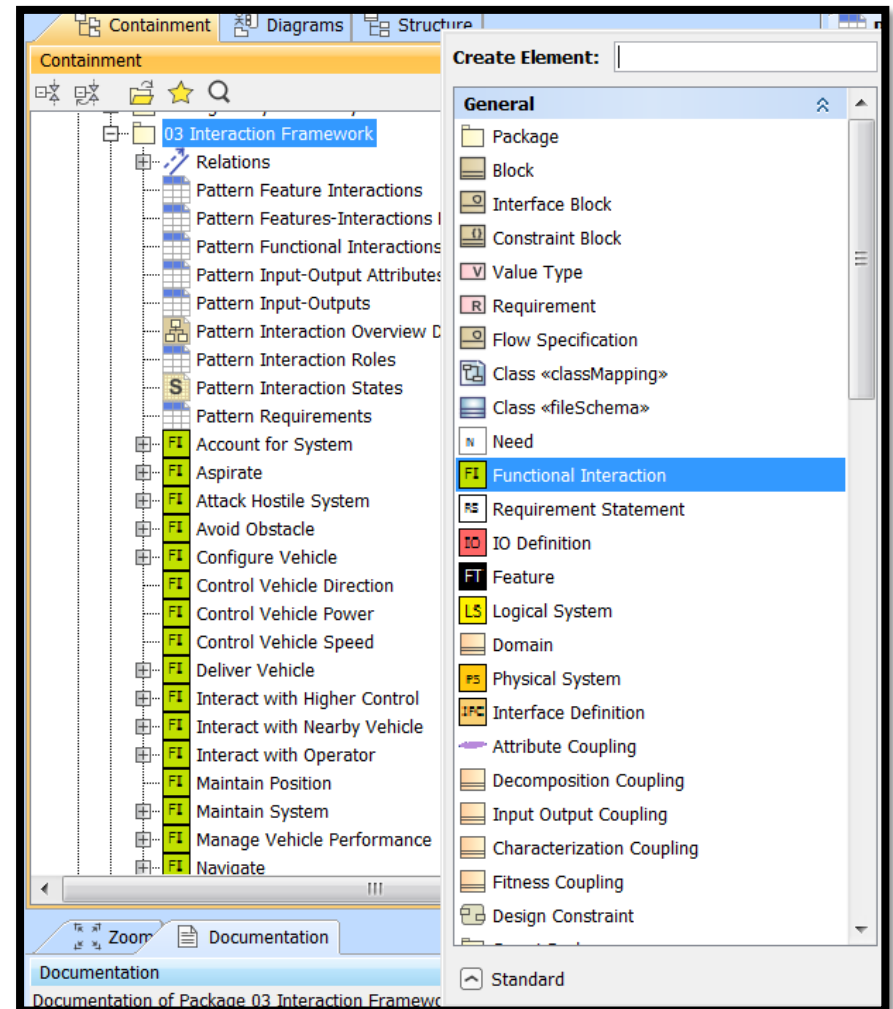
Domain Analysis: Creating Logical System Attributes

- Select a Logical System under the 01 Local Pattern::02 Logical System Analysis package
- Use ~Create Element and choose Value Property
- Right click and refactor the new Value Property as a Logical System Attribute



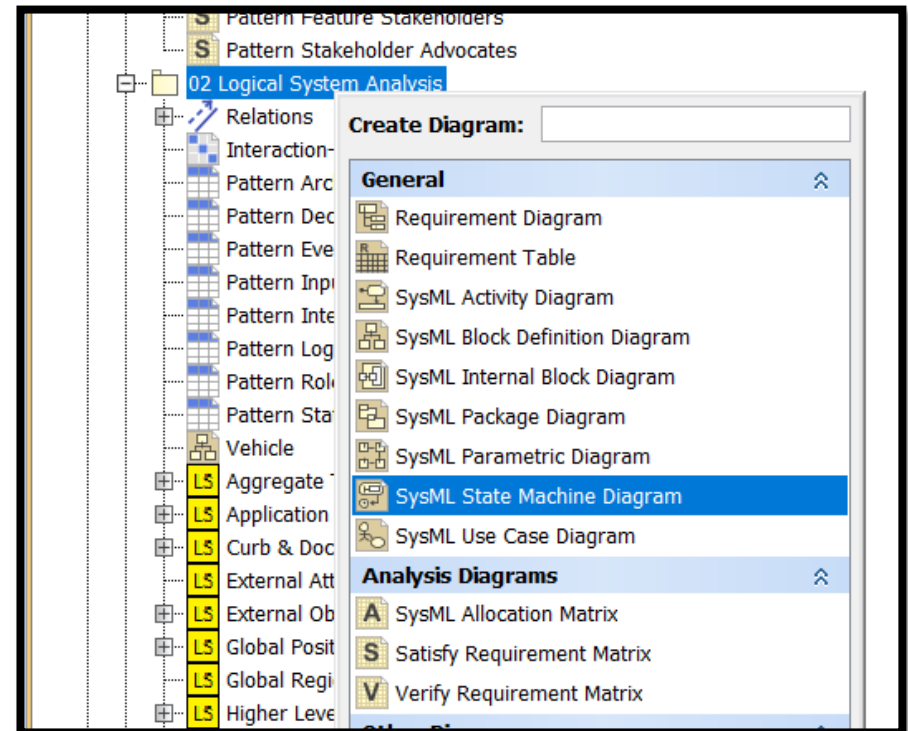
Interaction Framework: Creating Functional Interactions

- Pattern Functional Interactions are created in package 01 Local Pattern::03 Interaction Framework
- Use ~Create Element and choose Functional Interaction



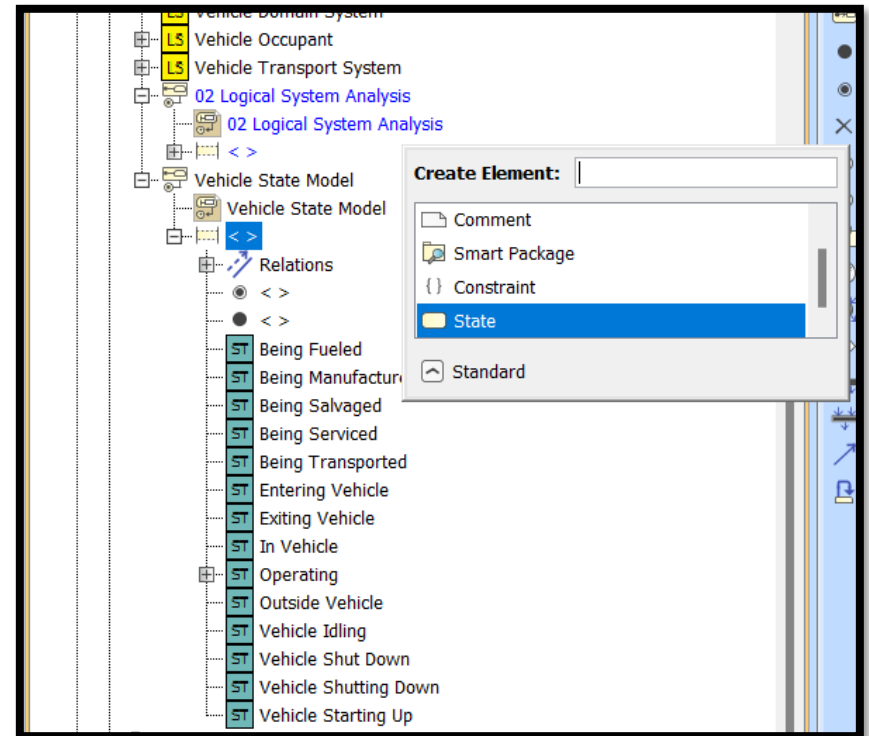
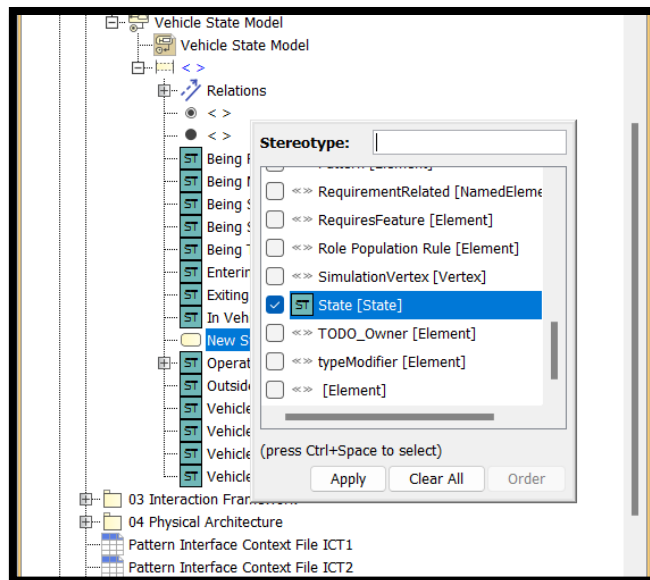
State Machine: Creating State Machine and States

- Pattern State Diagram and States are created in package 01 Local Pattern:::02 Logical System Analysis
- Use ~Create Diagram and choose SysML State Machine Diagram



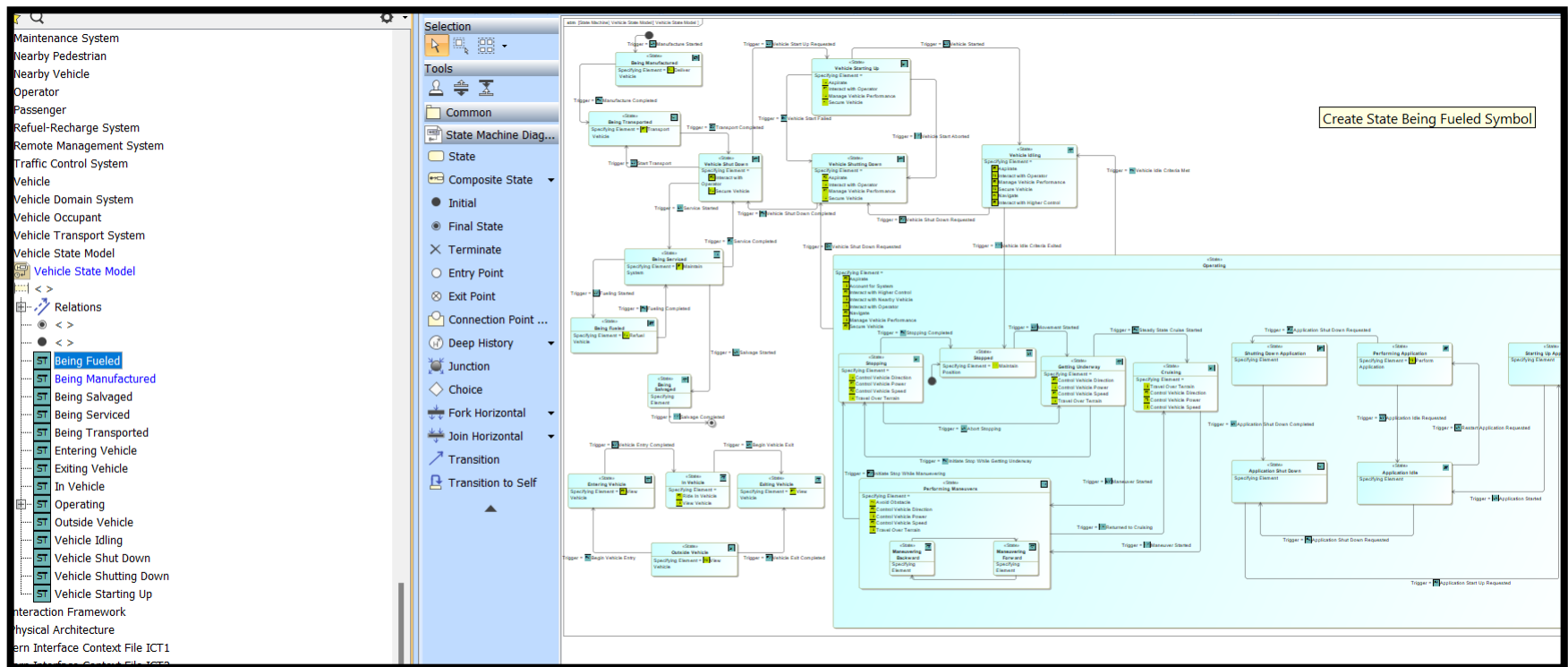
State Machine: Creating State Machine and States

- Right-click the automatically created Region under the State Model and use ~Create Element and choose State. Make sure the menu is expanded to Expert.
- Right-click the newly created State, select Stereotype, and change it to State as shown below.



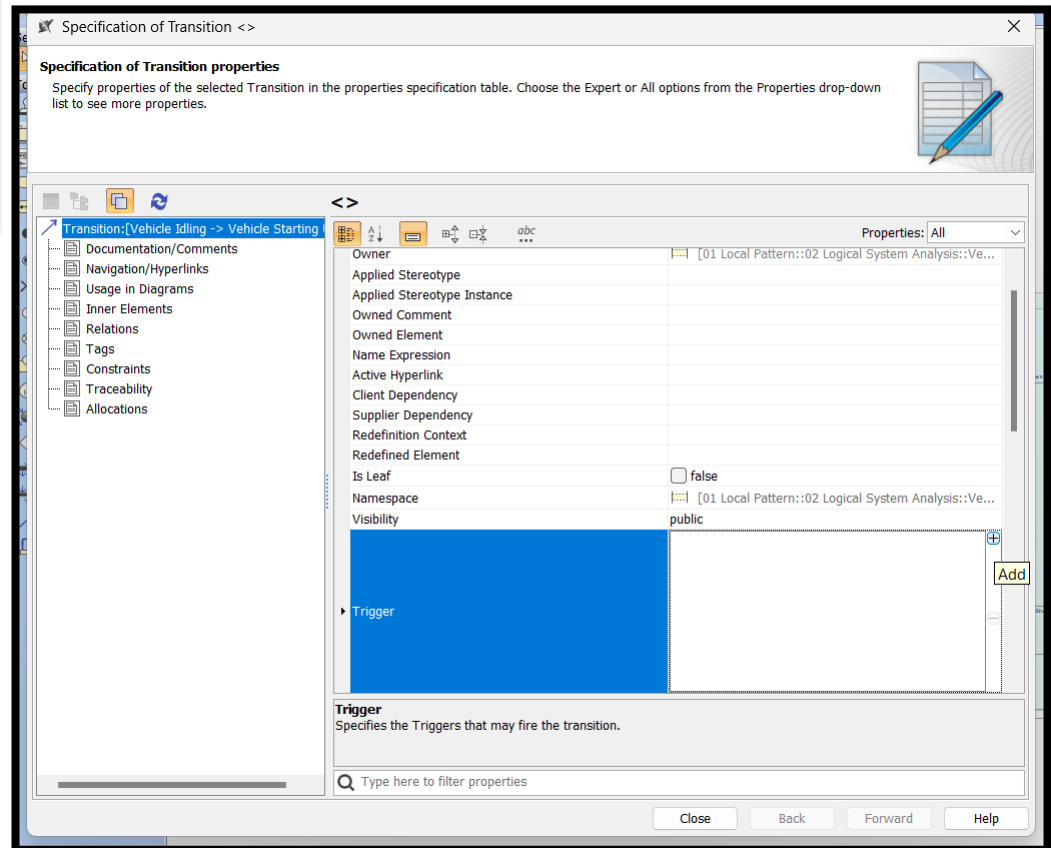
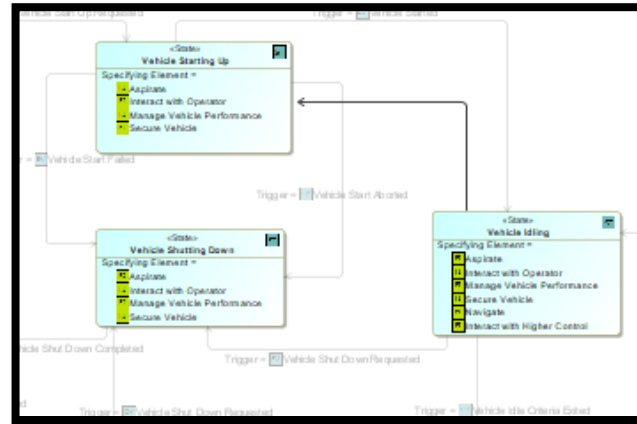
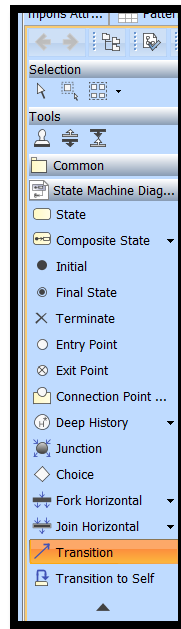
State Machine: Creating the Diagram

- Open the State Machine Diagram created in the earlier step.
- Populate the State Machine Diagram with States by dragging them from the containment browser onto the diagram.



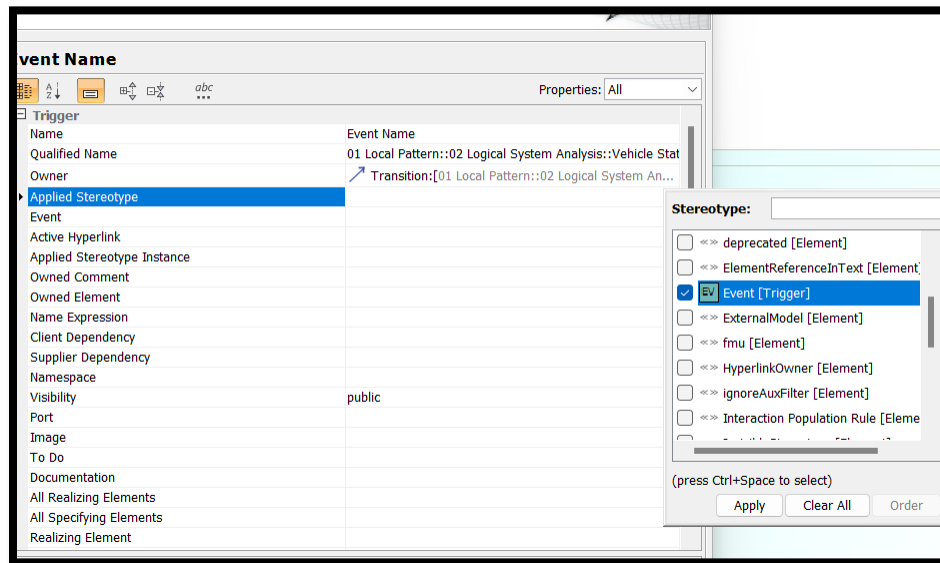
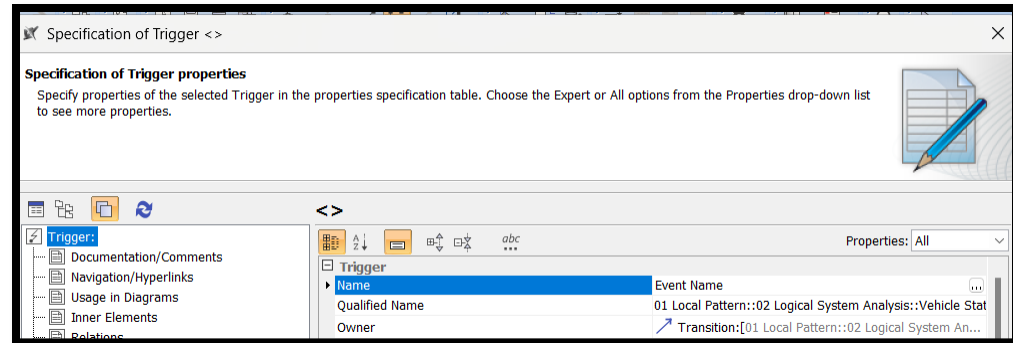
State Machine: Creating State-State Transitions, Events

- Select the Transition item from the State Machine Diagram Toolbar.
- Select the two states on the diagram.
- Right-click the new transition and open its specification.
- Select the Trigger property and click the + (Add) button.



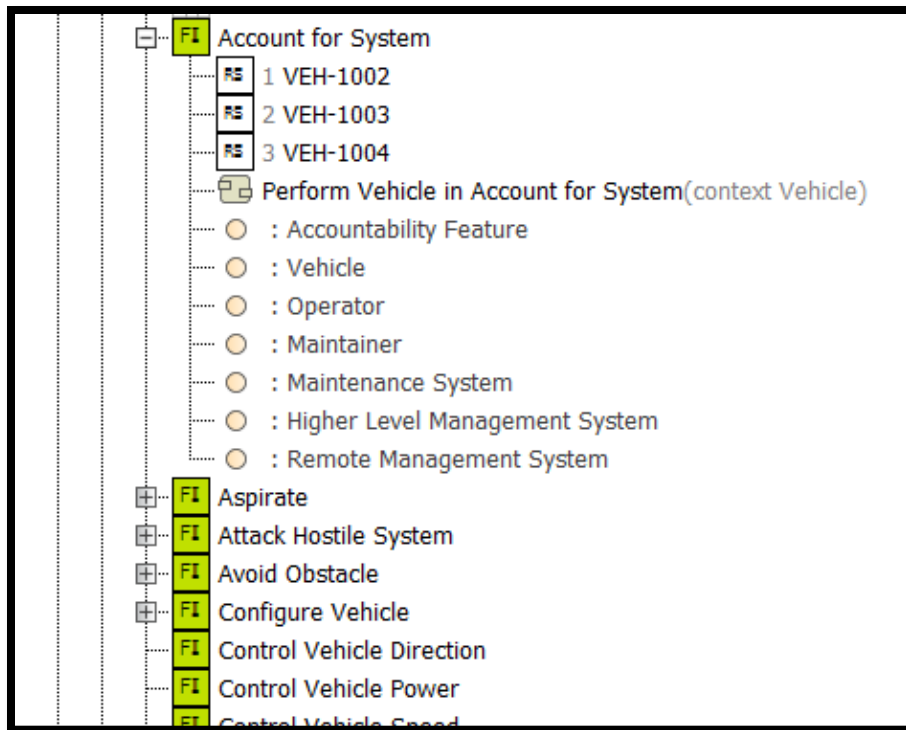
State Machine: Creating State-State Transitions, Events

- In the new Trigger Specification window that opens enter the name of the event in the Name property.
- Select the Applied Stereotype property and choose the Event stereotype option.



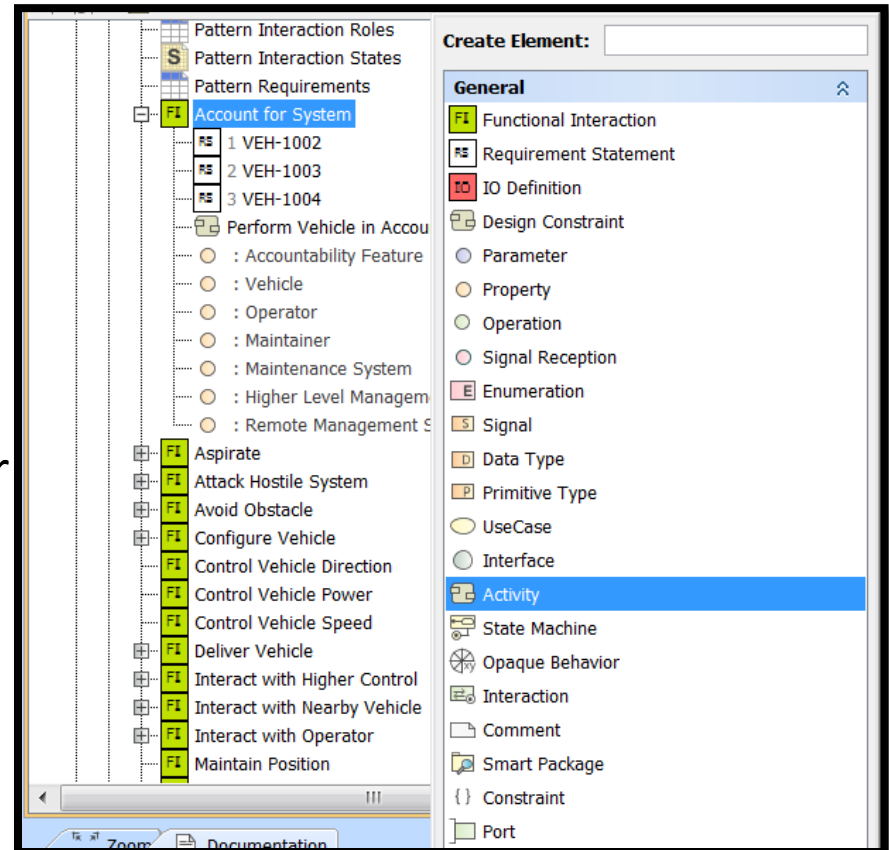
Detail Requirements: Overview

- Requirement Statements and the Requirement Transfer Functions they specify are created under their respective Functional Interaction.
- There should be at least one Requirement Transfer Function for each Interaction-Role pair.



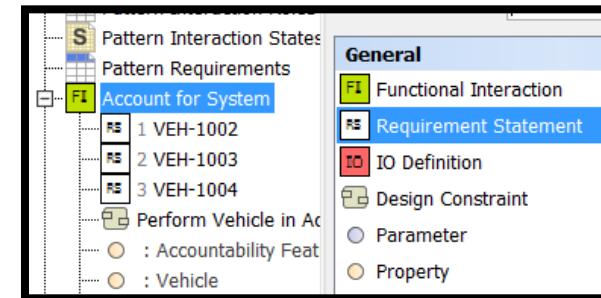
Detail Requirements: Creating Requirement Transfer Functions

- Use ~Create Element and choose Activity
- Add the stereotype “Requirement Transfer Function” to the new Activity
- Create an Allocate dependency from the new Requirement Transfer Function to the Logical System will perform it.



Detail Requirements: Creating Requirement Statements

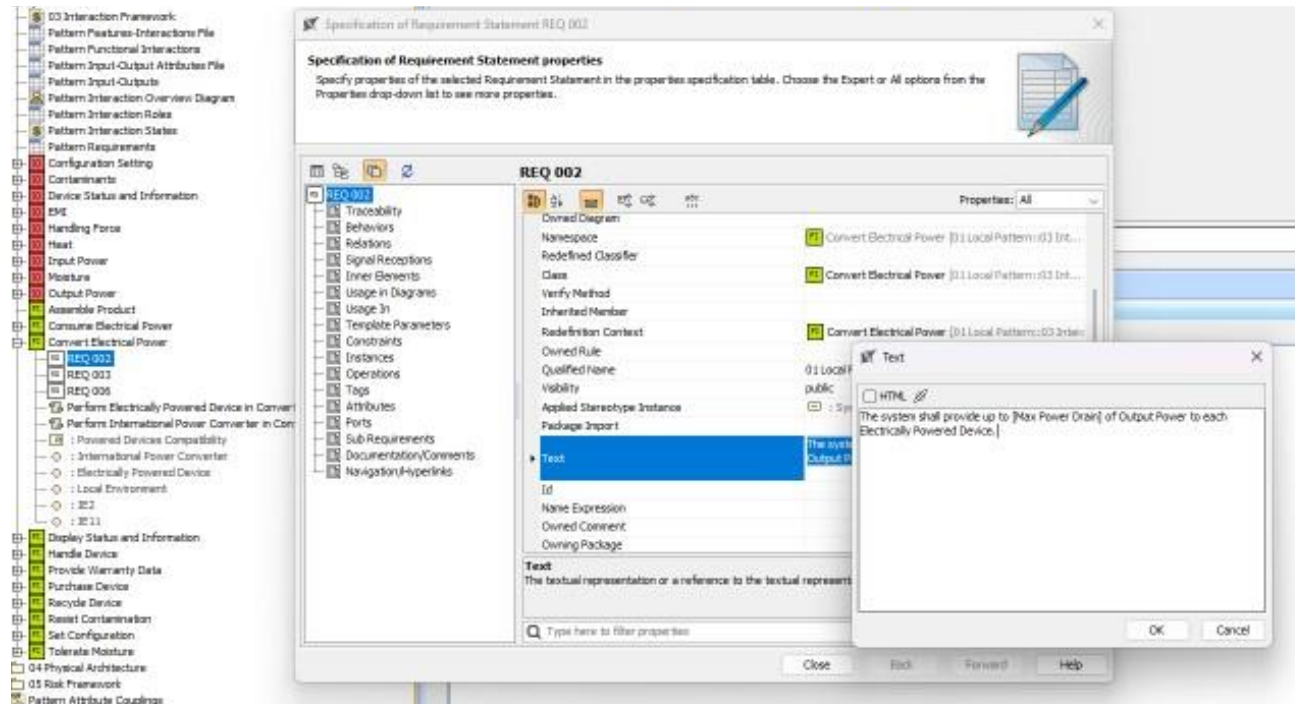
- Use ~Create Element and choose Requirement Statement
- Name the new Requirement Statement with its Requirement ID.
- Create a Satisfy dependency to the new Requirement Statement from the Requirement Transfer Function it specifies.
- At this point, the Requirement row will appear in the Pattern Requirements Table. The Requirement Statement can be entered in the Text column.
- Alternatively, open the Specification for the new Requirement Statement and enter the statement details in the Text property field.



#	Functional Interaction	IPK Value	Functional Role	RPK Value	Req ID	RSPK Rule	Text
1	FI Consume Electrical Power		LS International Power Converter		REQ 001		The system shall consume not more than [Max Drain on Mains] of Input Power from the Local Power Distribution System.
2	FI Convert Electrical Power	*ANY*	LS International Power Converter	*ANY*	REQ 002	IPK	The system shall provide up to [Max Power Drain] of Output Power to each Electrically Powered Device.
3	FI Convert Electrical Power	*ANY*	LS Electrically Powered Device	*ANY*	REQ 003	IPK	The system shall not consume more than [Max Power Drain] of Output Power.
4	FI Consume Electrical Power		LS Local Power Distribution System		REQ 004		The system shall provide up to [Max Drain on Mains] of Input Power.
5	FI Handle Device		LS International Power Converter		REQ 005		The system shall not present a shock hazard to users when operated according to its instructions.
6	FI Convert Electrical Power	*ANY*	LS International Power Converter	*ANY*	REQ 006	IPK	The system shall generate Output Power to attached Electrically Powered Devices which is compatible with the [Output Voltage-Power Profile] and [Output Frequency Profile].

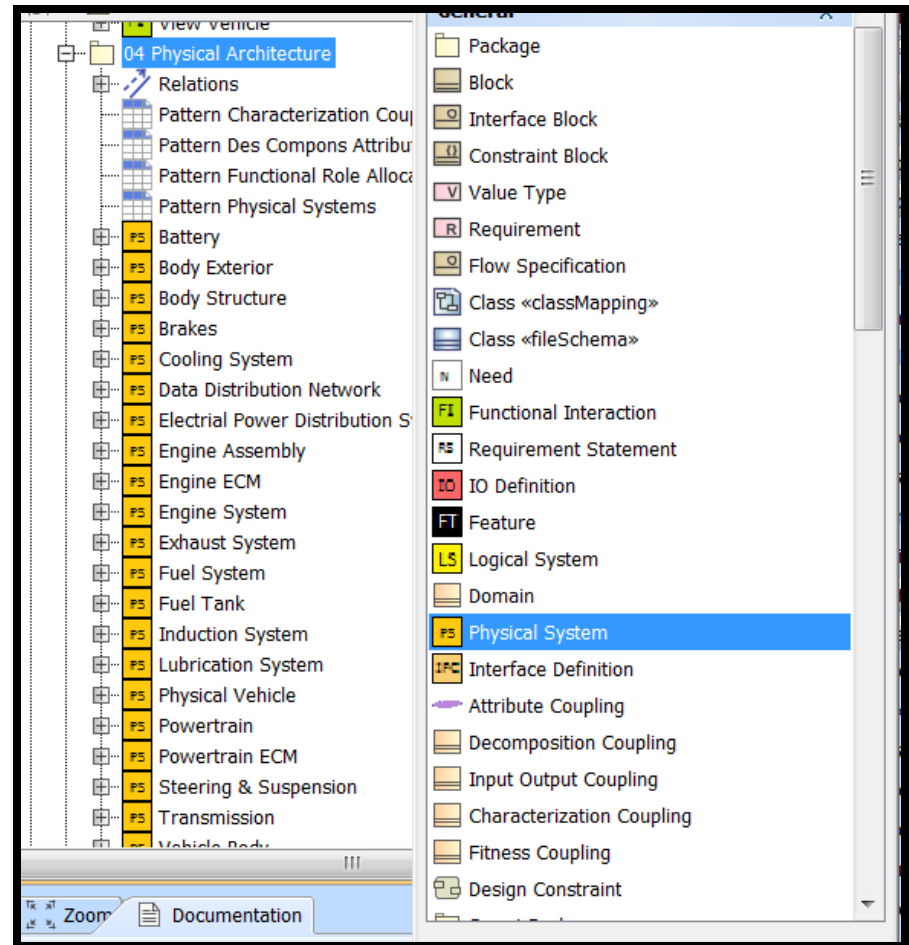
Detail Requirements: Creating Requirement Statements

- Alternatively, open the Specification for the new Requirement Statement and enter the statement details in the Text property field.



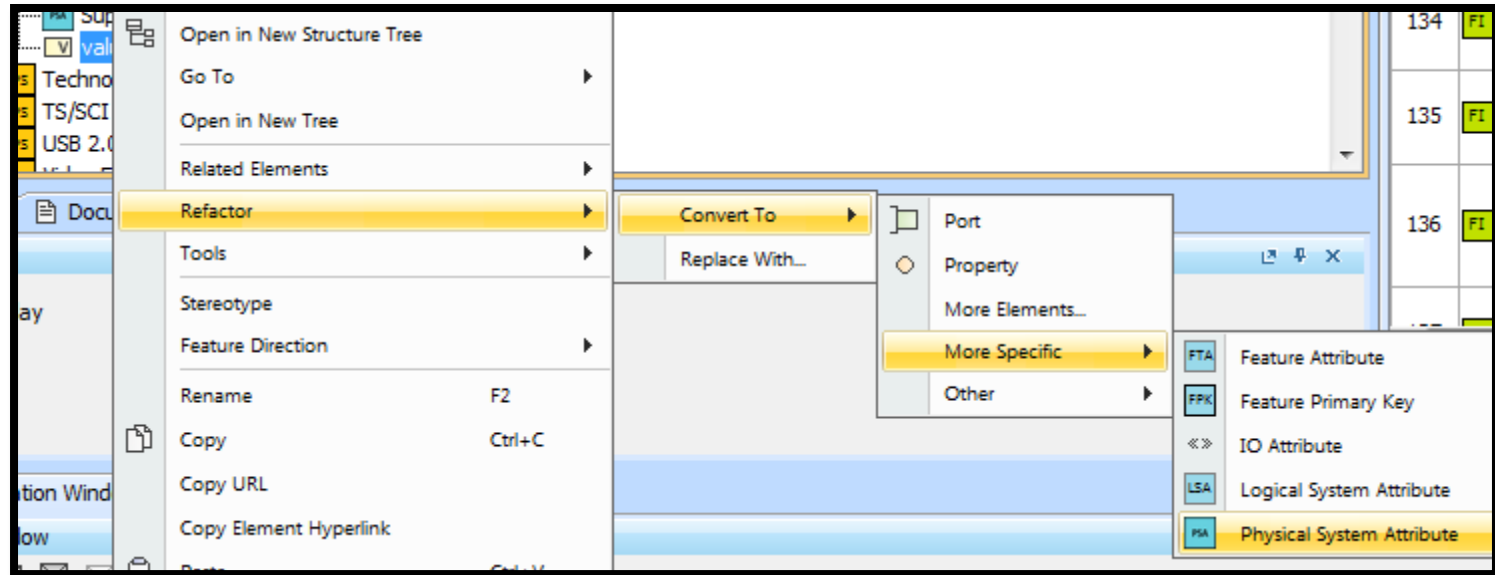
High Level Design: Creating Design Components (Physical Systems)

- Pattern Design Components are created as Physical Systems in package 01 Local Pattern::04 Physical Architecture
- Use ~Create Element and choose Physical System



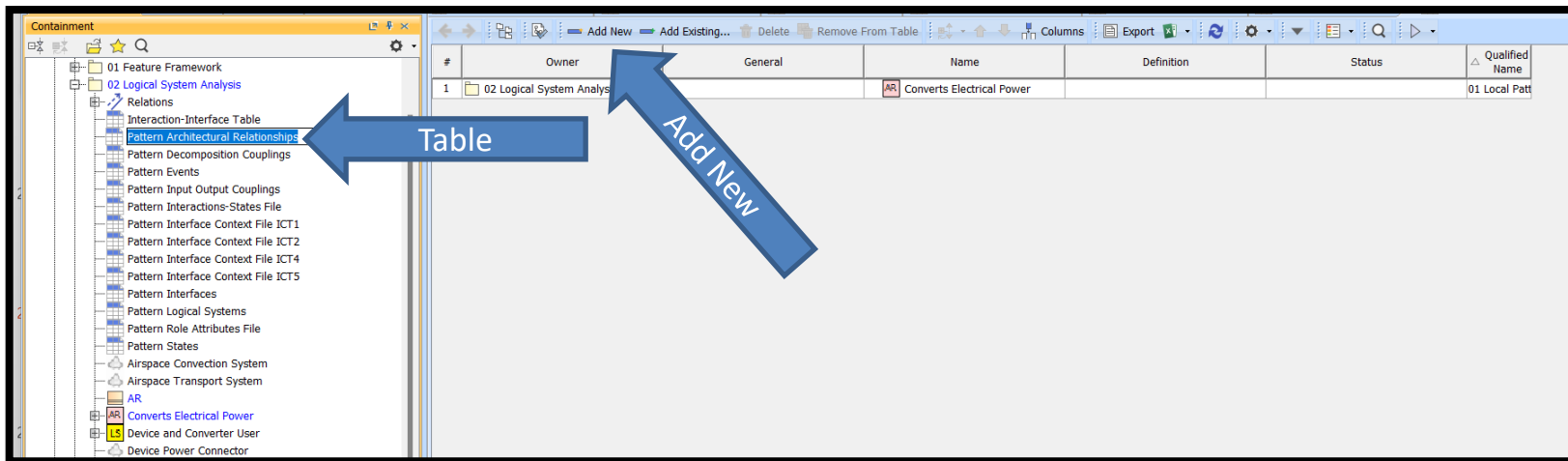
High Level Design: Creating Design Component (Physical System) Attributes

- Use ~Create Element and choose Value Property
- Right click and refactor the new Value Property as a Physical System Attribute



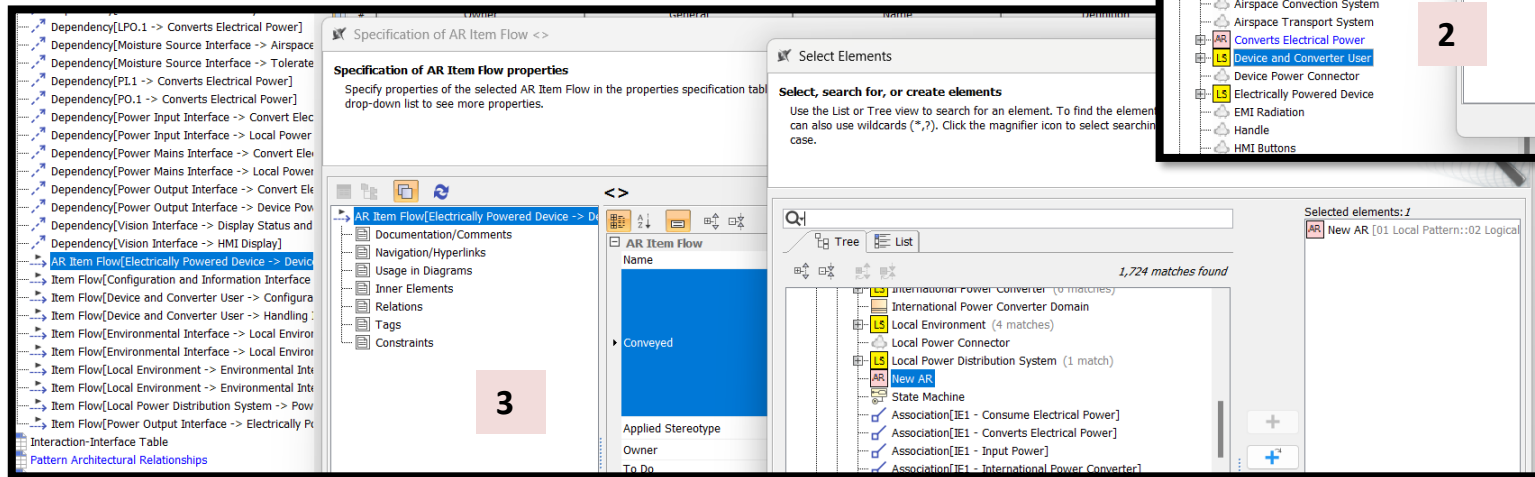
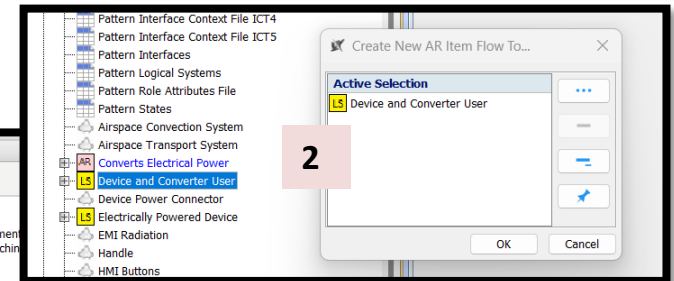
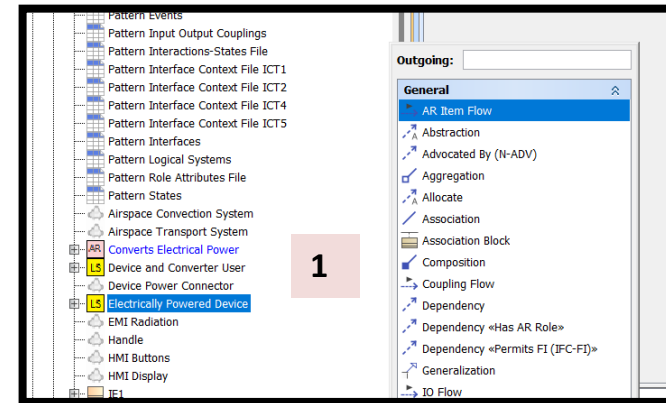
Interface Context: Architectural Relationships

- Architectural Relationships are created in package 01 Local Pattern::02 Logical System Analysis.
- Select and Open the Pattern Architectural Relationships Table and then Select Add New.
- The newly added Architectural Relationship will show up in the table and the Containment Browser



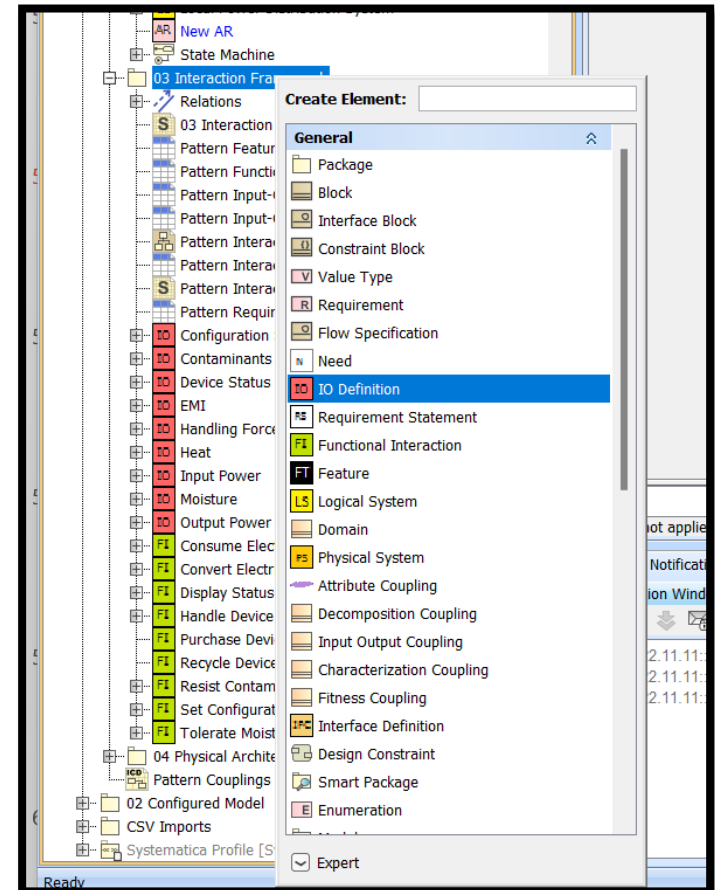
Interface Context: Architectural Relationships

1. Select the Logical System or Physical System that conveys the Architectural Relationship > Create Relation > Select Outgoing or Incoming > Select AR Item Flow.
2. Select the other Logical System or Physical System involved in the Architectural Relationship.
3. Select the new AR Item Flow relation in the browser, open its specification, select the Conveyed property and choose the newly created Architectural Relationship.



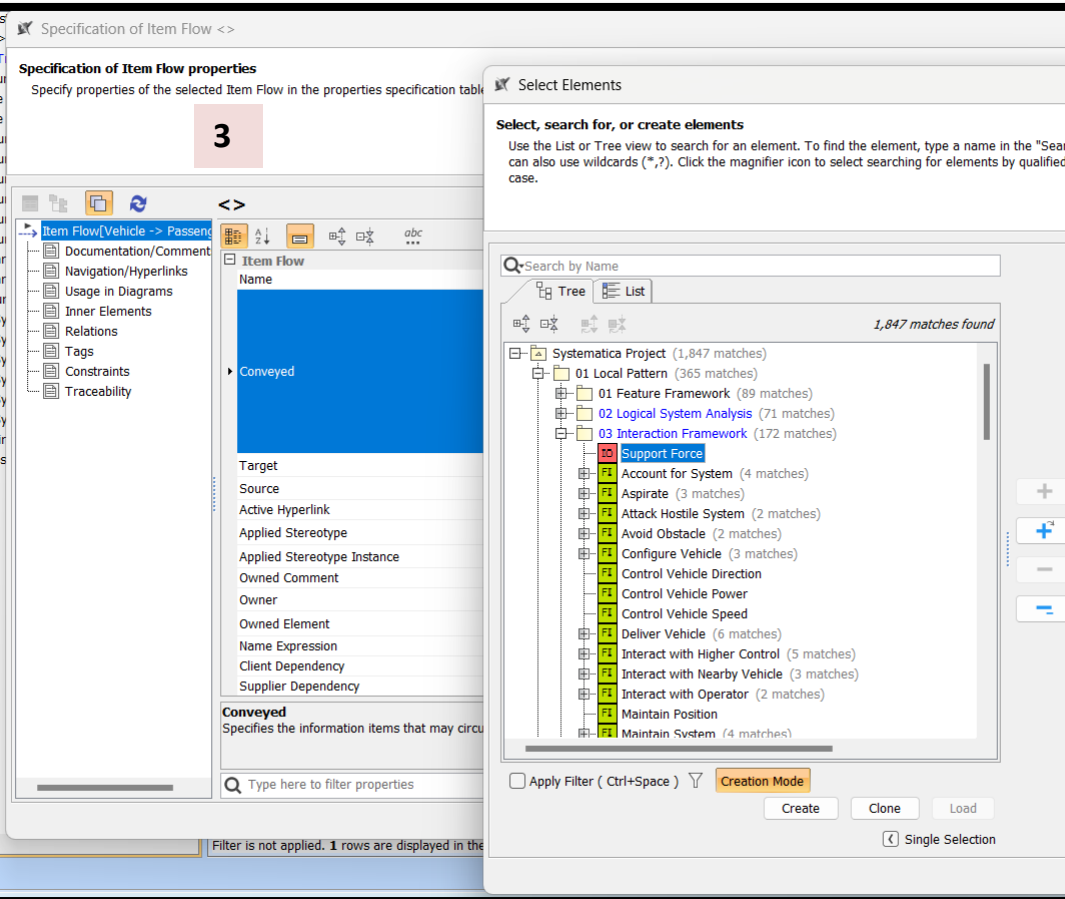
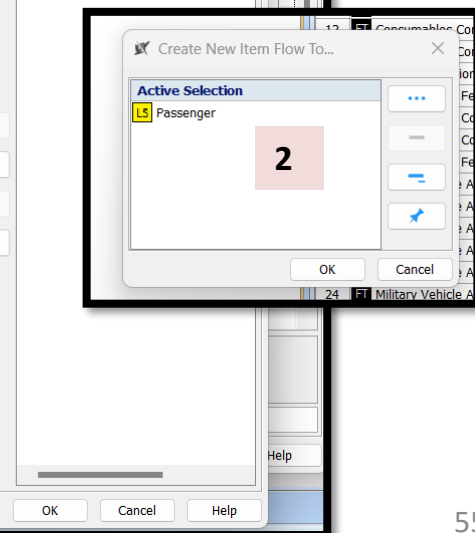
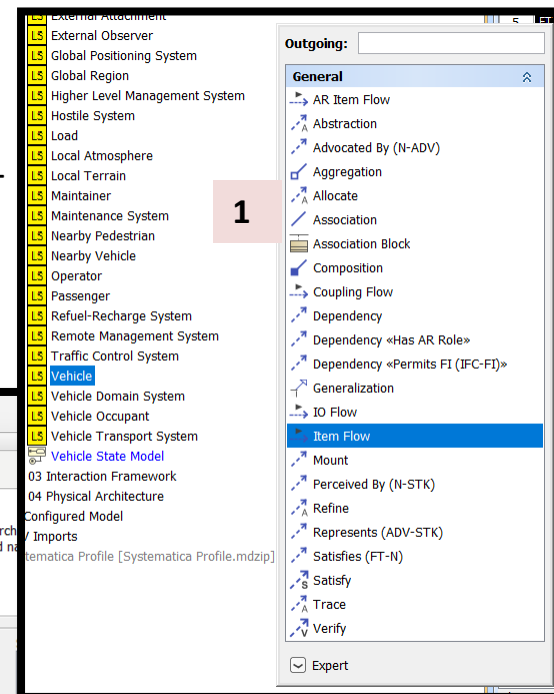
Interface Context: Populating Input-Outputs

- Input-Outputs are created in package 01 Local Pattern::03 Interaction Framework.
- Use ~Create Element and choose IO Definition



Interface Context: Populating Input-Outputs continued

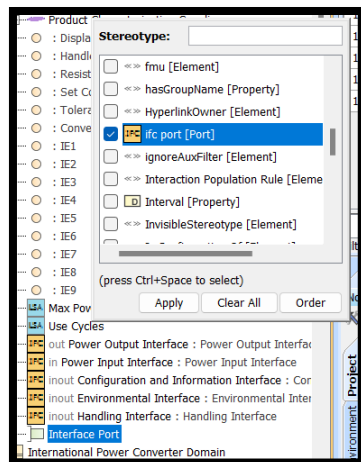
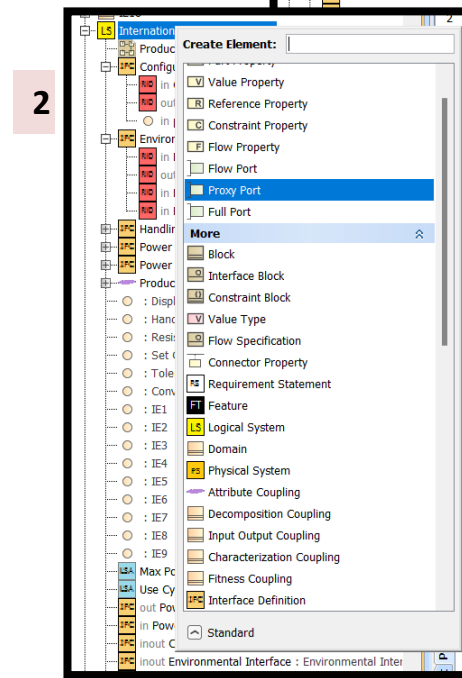
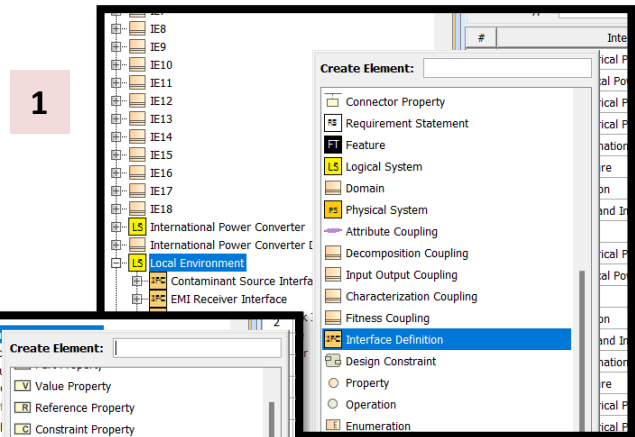
1. Select the Logical System or Physical System that conveys the Input-Output > Create Relation > Select Outgoing or Incoming > Select Item Flow.
2. Select the other Logical System or Physical System involved in the Input-Output.
3. Select the new Item Flow relation in the browser, open its specification, select the Conveyed property and choose the newly created Input-Output.



Interface Context: Populating Interfaces

- Interface Definitions are created in package 01 Local Pattern:::02 Logical System Analysis under the Logical System that owns them.

1. Use ~Create Element and choose Interface Definition; give the new Interface a name.
2. Next, use ~Create Element and choose Proxy Port; give it the Proxy Port the same name as the Interface Definition created in the above step.
3. Select the Proxy Port in the browser, select the Stereotype option, then add ifc port stereotype.



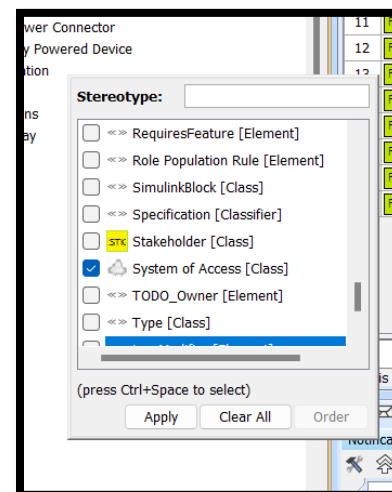
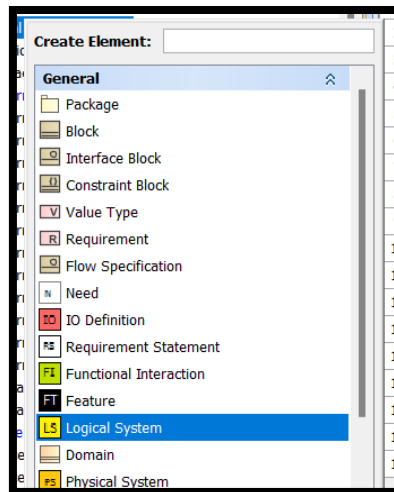
Interface Context: Populating Interfaces, continued

- Select the new ifc port in the browser, open its specification, select the Type property and choose the associated Interface Definition with the same name.

The screenshot displays a software interface for defining interface properties. On the left, a tree view shows the 'International Power Converter' domain with various interfaces like 'Configuration and Information Interface', 'Environmental Interface', and 'Handling Interface'. The right pane, titled 'Specification of Proxy Port Interface Port', shows the 'Interface Port' properties. The 'Type' property is selected, and a dropdown menu is open, listing several interface definitions such as 'Configuration and Information Interface', 'Contaminant Source Interface', and 'Device Power Interface'. The 'Type' property is currently set to '<UNSPECIFIED>'.

Interface Context: Populating Systems of Access (SOAs)

- Systems of Access are created in package 01 Local Pattern::02 Logical System Analysis
- Use ~Create Element and choose Logical System (because System of Access is a type of Logical System).
- Right-click on the newly created Logical System, select Stereotype and change the Stereotype to System of Access and remove the Logical System Stereotype.

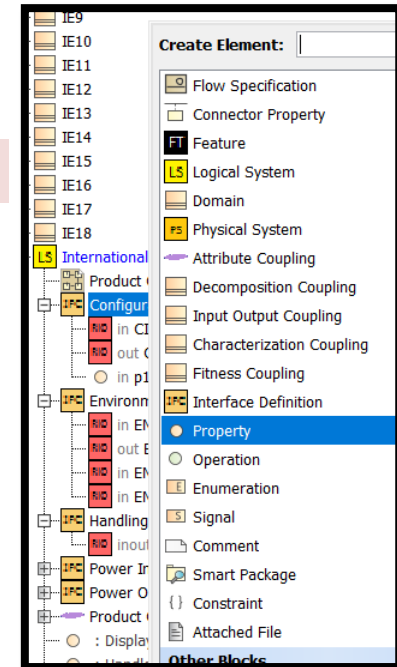


Interface Context: Populating Ports

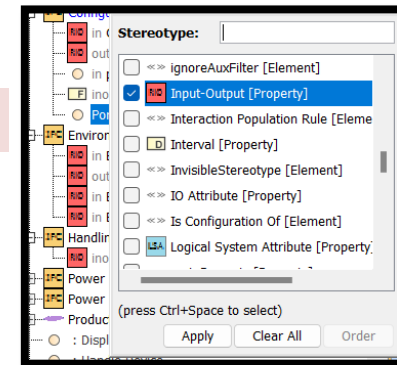
- Ports are created in package 01 Local Pattern::02 Logical System Analysis under the Interface Definition that owns the Port:

1. Use ~Create Element and choose Property.
2. Select and right-click the property in the containment browser, choose Stereotype, choose Input-Output [Property].
3. Select the Port/IO Property in the browser, open its specification, select the Type property and choose the associated IO Definition.

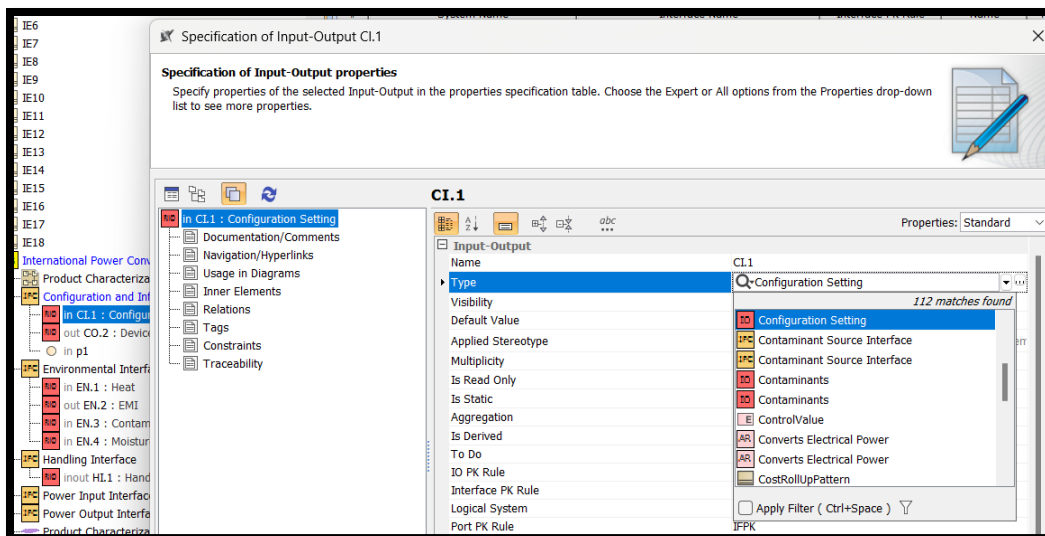
1



2

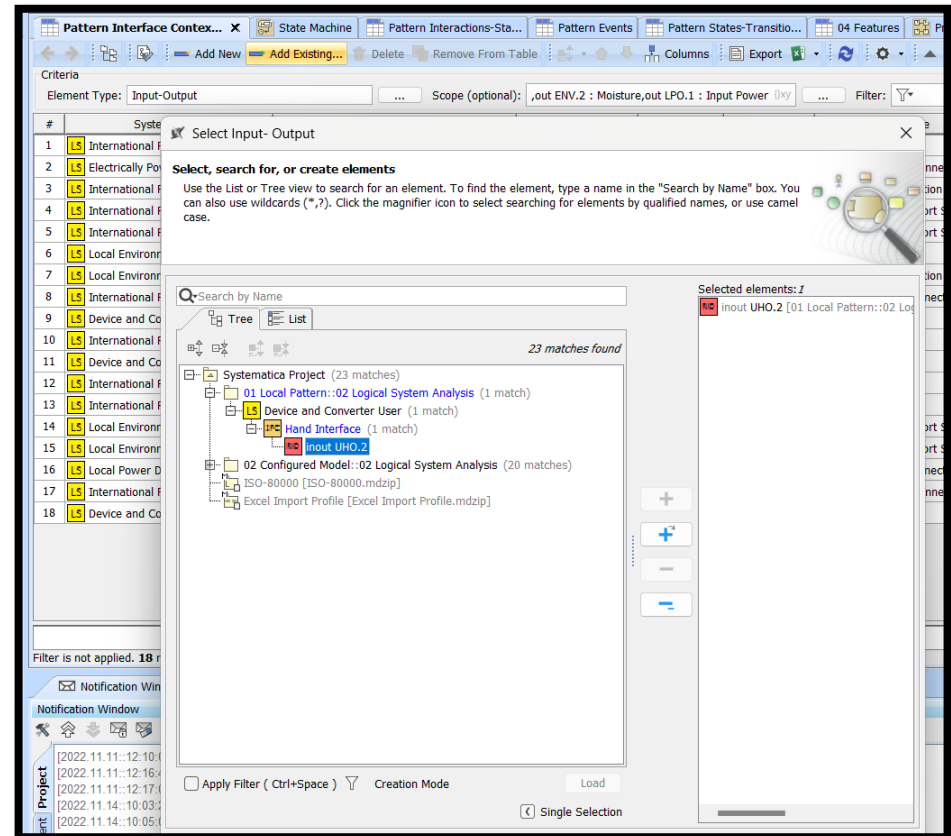


3



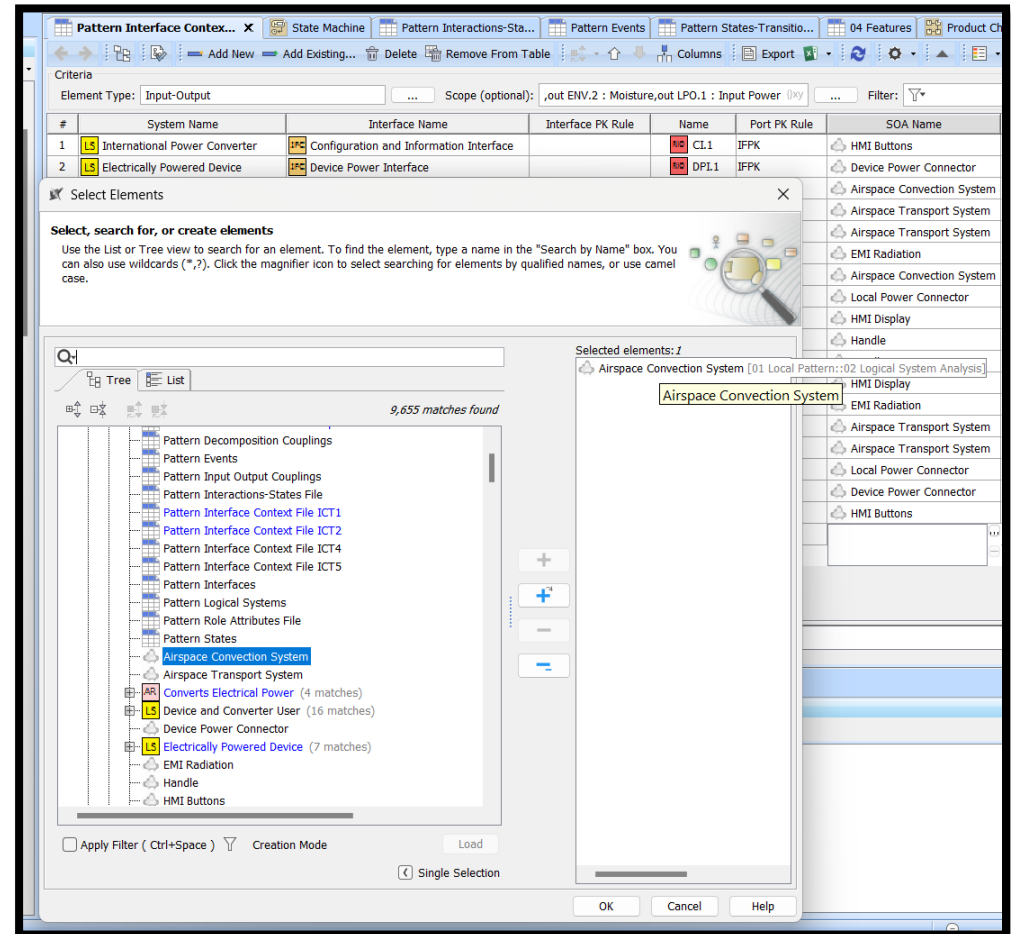
Interface Context: Ports, Table ICT1

- Pattern Table ICT1 shows relationships to Ports and also administers the related configuration rules.
- To add a Port to the table, select Add Existing and select the Port from its browser location.
- The Role, Interface, and Input-Output column will have content automatically populated based on the steps on the previous slide.



Interface Context: Ports, Table ICT1

- To create the relationship to Systems of Access (SOAs), double-click the cell, then the ellipses to bring up the selection dialog window.
- Choose the related SOA from the containment browser.



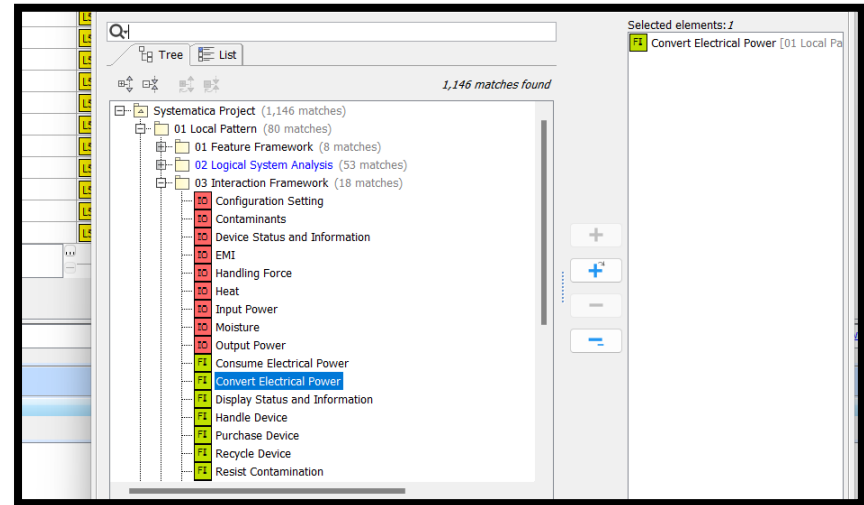
Interface Context: Ports, Table ICT1

#	System Name	△ Interface Name	Interface PK Rule	Input/Output	IO PK Rule	Direction	Port	Port PK Rule	SOA Name	SOA PK Rule	SOA Internal or External
1	International Power Converter	Configuration and Information Interface		Configuration Setting		in	CI.1	IFPK	HMI Buttons		
2	International Power Converter	Configuration and Information Interface		Device Status and Information		out	CO.2	IFPK	HMI Display		
3	Local Environment	Contaminant Source Interface		Contaminants		out	ENW.1	IFPK	Airspace Transport System		
4	Electrically Powered Device	Device Power Interface		Output Power	IPK	in	DPI.1	IFPK	Device Power Connector	IPK	
5	Local Environment	EMI Receiver Interface		EMI		in	ENV.3	IFPK	EMI Radiation		
6	Local Environment	Environment Thermal Sink Interface		Heat		in	ENW.4	IFPK	Airspace Convection System		
7	International Power Converter	Environmental Interface		Heat		in	EN.1	IFPK	Airspace Convection System		
8	International Power Converter	Environmental Interface		Contaminants		in	EN.3	IFPK	Airspace Transport System		
9	International Power Converter	Environmental Interface		Moisture		in	EN.4	IFPK	Airspace Transport System		
10	International Power Converter	Environmental Interface		EMI		out	EN.2	IFPK	EMI Radiation		
11	Device and Converter User	Finger Interface		Configuration Setting		out	UCO.1	IFPK	HMI Buttons		
12	Device and Converter User	Hand Interface		Handling Force		inout	UHO.1	IFPK	Handle		
13	International Power Converter	Handling Interface		Handling Force		inout	HI.1	IFPK	Handle		
14	Local Environment	Moisture Source Interface		Moisture		out	ENV.2	IFPK	Airspace Transport System		
15	International Power Converter	Power Input Interface		Input Power		in	PI.1		Local Power Connector		
16	Local Power Distribution System	Power Mains Interface		Input Power		out	LPO.1	IFPK	Local Power Connector		
17	International Power Converter	Power Output Interface	IPK	Output Power	IPK	out	PO.1	IFPK	Device Power Connector	IPK	
18	Device and Converter User	Vision Interface		Device Status and Information		in	UCI.1	IFPK	HMI Display		

Example Pattern Interface Context File ICT1

Interface Context: Ports, Table ICT2

- Pattern Table ICT2 shows relationships among Interactions, Roles, Input-Outputs, and Architectural Relationships.
- Each row is an Interface Element. To add a row, select Add New, double-click the new row and give the Interface Element a name.
- To create the relationships to other elements, double-click the cell, then the ellipses to bring up the selection dialog window.
- Choose the related element from the containment browser.



Add New → Add New → Add Existing... → Delete → Remove From Table → Columns → Export → Filter

Criteria
 Element Type: Interface Element Scope (optional): IE1 Filter: ▼

#	Name	Interaction Name	System Name	IO Name	Arch Relat
1	IE1	Consume Electrical Power	International Power Converter	Input Power	Converts Electrical Power
2	IE2	Convert Electrical Power	International Power Converter	Output Power	Converts Electrical Power
3	IE3	Consume Electrical Power	International Power Converter	Heat	
4	IE4	Consume Electrical Power	International Power Converter	EMI	
5	IE5	Resist Contamination	International Power Converter	Contaminants	
6	IE6	Tolerate Moisture	International Power Converter	Moisture	
7	IE7	Set Configuration	International Power Converter	Configuration Setting	
8	IE8	Display Status and Information	International Power Converter	Device Status and Information	
9	IE9	Handle Device	International Power Converter	Handling Force	
10	IE10	Consume Electrical Power	Local Power Distribution System	Input Power	Converts Electrical Power
11	IE11	Convert Electrical Power	Electrically Powered Device	Output Power	Converts Electrical Power
12	IE12	Handle Device	Device and Converter User	Handling Force	
13	IE13	Set Configuration	Device and Converter User	Configuration Setting	
14	IE14	Display Status and Information	Device and Converter User	Device Status and Information	
15	IE15	Resist Contamination	Local Environment	Contaminants	
16	IE16	Tolerate Moisture	Local Environment	Moisture	
17	IE17	Consume Electrical Power	Local Environment	EMI	
18	IE18	Consume Electrical Power	Local Environment	Heat	

IE Name

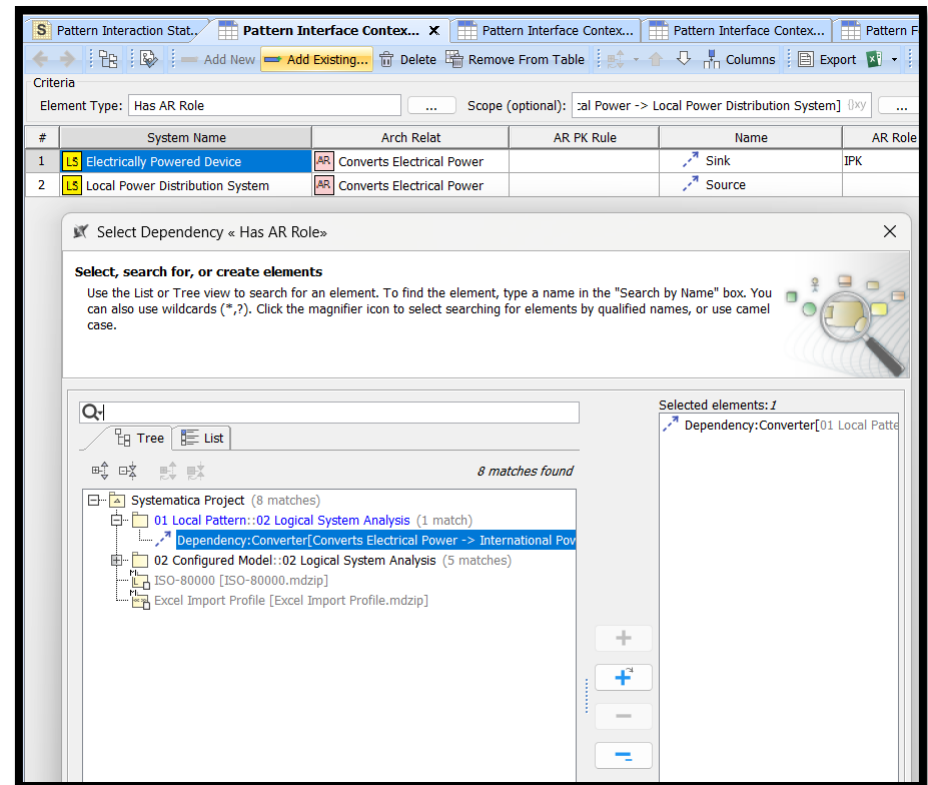
Interface Context: Ports, Table ICT2

#	Interaction Name	System Name	IO Name	Arch Relat
1	FE Consume Electrical Power	LS International Power Converter	IO Input Power	AR Converts Electrical Power
2	FE Convert Electrical Power	LS International Power Converter	IO Output Power	AR Converts Electrical Power
3	FE Consume Electrical Power	LS International Power Converter	IO Heat	
4	FE Consume Electrical Power	LS International Power Converter	IO EMI	
5	FE Resist Contamination	LS International Power Converter	IO Contaminants	
6	FE Tolerate Moisture	LS International Power Converter	IO Moisture	
7	FE Set Configuration	LS International Power Converter	IO Configuration Setting	
8	FE Display Status and Information	LS International Power Converter	IO Device Status and Information	
9	FE Handle Device	LS International Power Converter	IO Handling Force	
10	FE Consume Electrical Power	LS Local Power Distribution System	IO Input Power	AR Converts Electrical Power
11	FE Convert Electrical Power	LS Electrically Powered Device	IO Output Power	AR Converts Electrical Power
12	FE Handle Device	LS Device and Converter User	IO Handling Force	
13	FE Set Configuration	LS Device and Converter User	IO Configuration Setting	
14	FE Display Status and Information	LS Device and Converter User	IO Device Status and Information	
15	FE Resist Contamination	LS Local Environment	IO Contaminants	
16	FE Tolerate Moisture	LS Local Environment	IO Moisture	
17	FE Consume Electrical Power	LS Local Environment	IO EMI	
18	FE Consume Electrical Power	LS Local Environment	IO Heat	

Example Pattern Interface Context File ICT2

Interface Context: Architectural Relationships, Table ICT4

- Pattern Table ICT4 shows details for reified Architectural Relationships.
- Each row is a Has AR Role dependency. To add a row, select Add Existing, navigate to the Has AR Role and add it, or drag the Has AR Role from the browser onto the table.



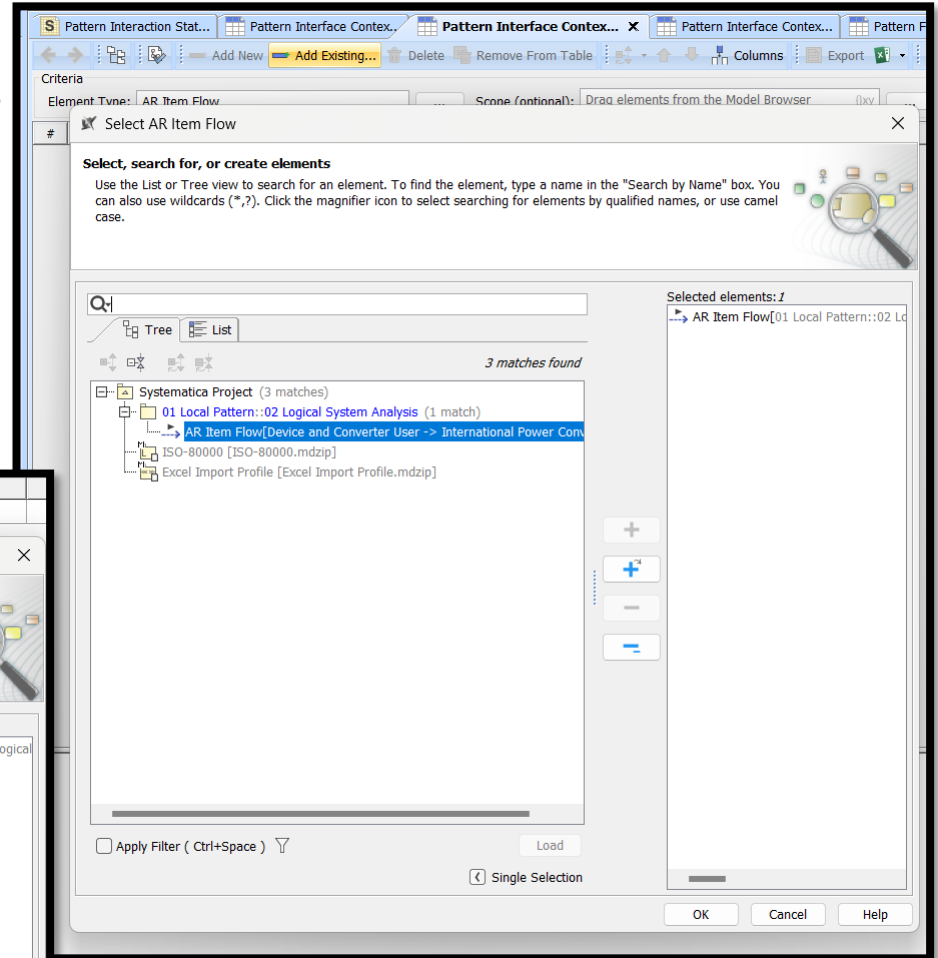
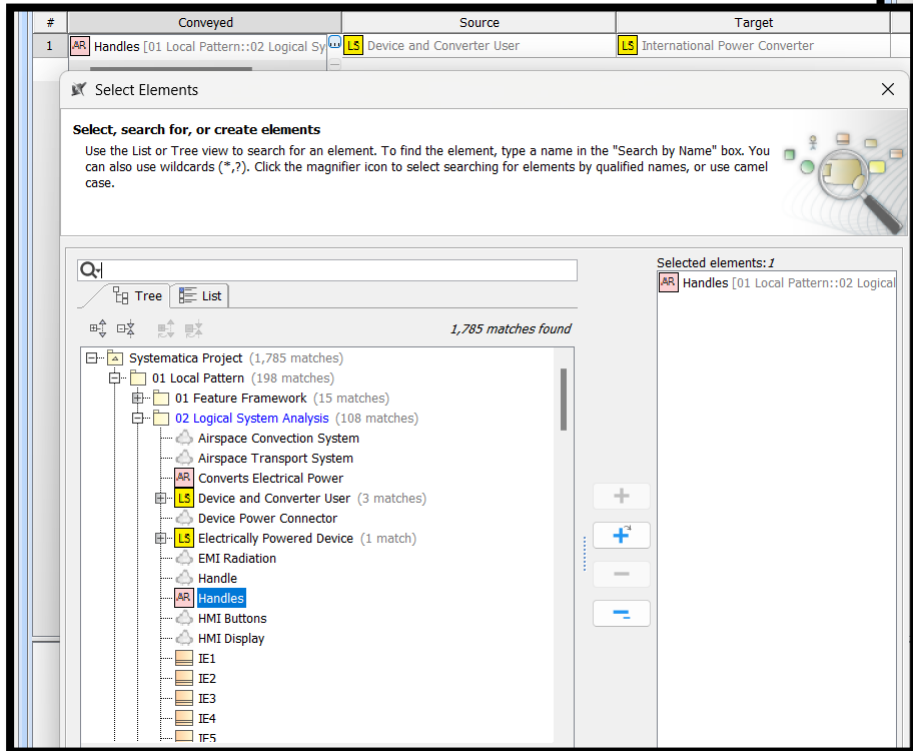
Interface Context: Architectural Relationships, Table ICT4

#	System Name	Arch Relat	AR PK Rule	AR Role Name	AR Role PK Rule	AR Internal or External	AR Complexity
1	LS Electrically Powered Device	AR Converts Electrical Power		Sink	IPK		Reified
2	LS Local Power Distribution System	AR Converts Electrical Power		Source			Reified
3	LS International Power Converter	AR Converts Electrical Power		Converter			Reified

Example Pattern Interface Context File ICT4

Interface Context: Architectural Relationships, Table ICT5

- Pattern Table ICT5 shows details for simple Architectural Relationships.
- Each row is an AR Item Flow. To add a row, select Add Existing, navigate to the AR Item Flow and add it. Or drag the AR Item Flow from the browser onto the table.
- To add the Conveyed property, click the ellipses, and select the AR.



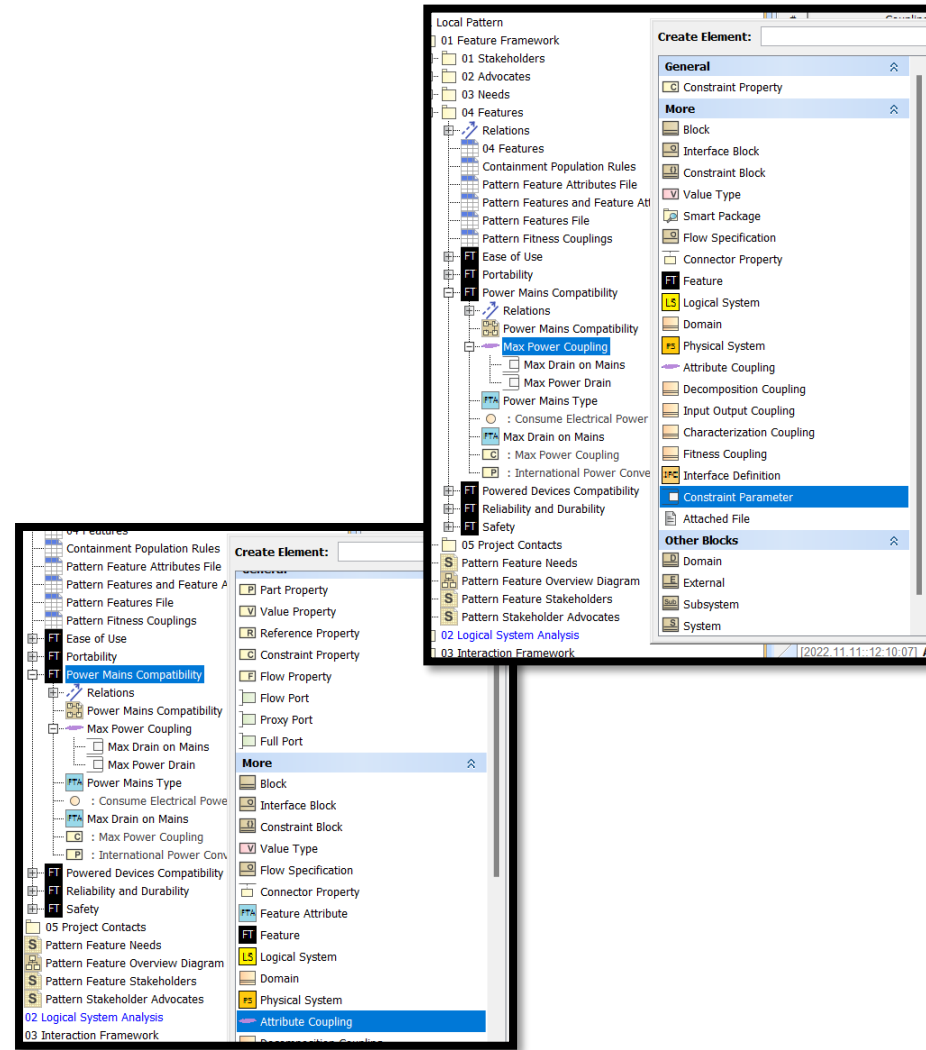
Interface Context: Architectural Relationships, Table ICT5

#	From Role	To Role	AR	AR PK Rule	AR Internal or External
1	LS Device and Converter User	LS International Power Converter	AR Handles		

Example Pattern Interface Context File ICT5

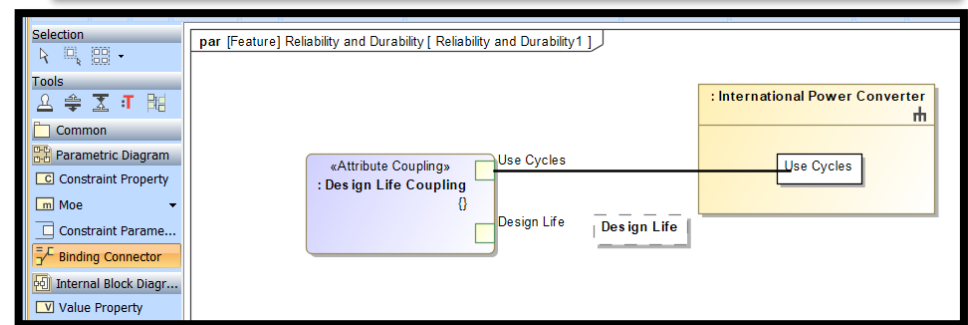
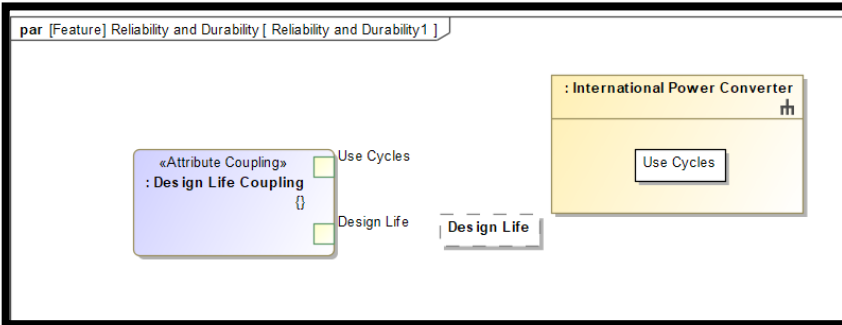
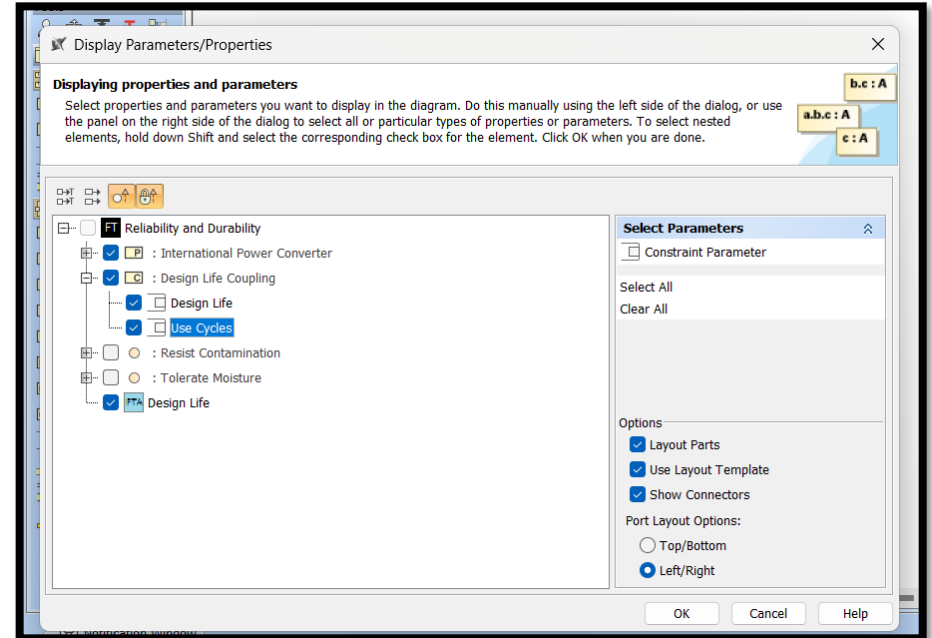
Attribute Couplings

- Attribute Couplings are created under the element that owns them in various packages including Features, Roles, Design Components, and Input-Outputs:
 - A given Attribute Coupling involves multiple Attributes, so which one belongs to the Coupling's owner?
 - Each Coupling drives (in a causality sense) only one Attribute—look there for the Coupling's ownership.
- Use ~Create Element and choose Attribute Coupling.
- Right-click on the new Attribute Coupling, use ~Create Element and choose Constraint Parameter. Give it the same name as the attribute it will be connected to.
- Add the two or more constraint parameters for the coupling(s).



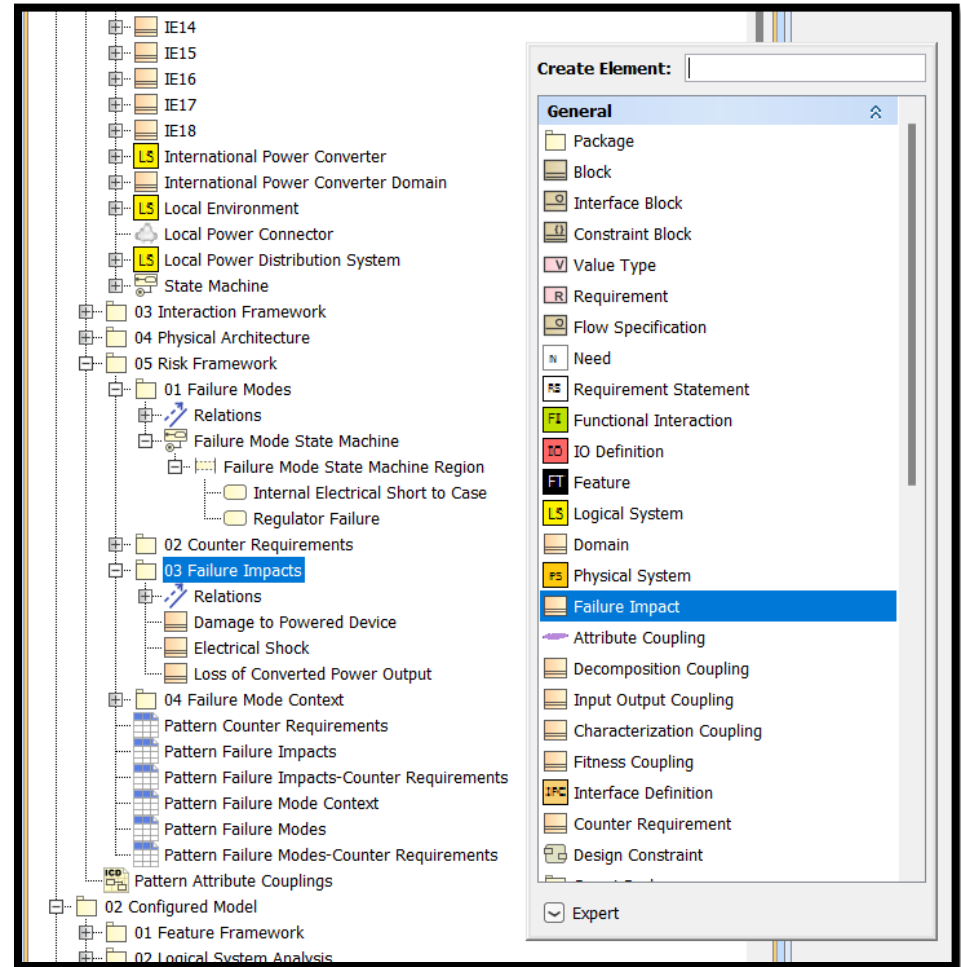
Attribute Couplings

- Right-click on the element that owns the new Attribute Coupling, use ~Create Diagram and choose SysML Parametric Diagram.
- Choose the associated display parameter details as shown in the box to the right.
- View the resulting populated diagram as shown below.
- Create binding connectors from the constraint parameters to the associated attributes as shown in the lower right.



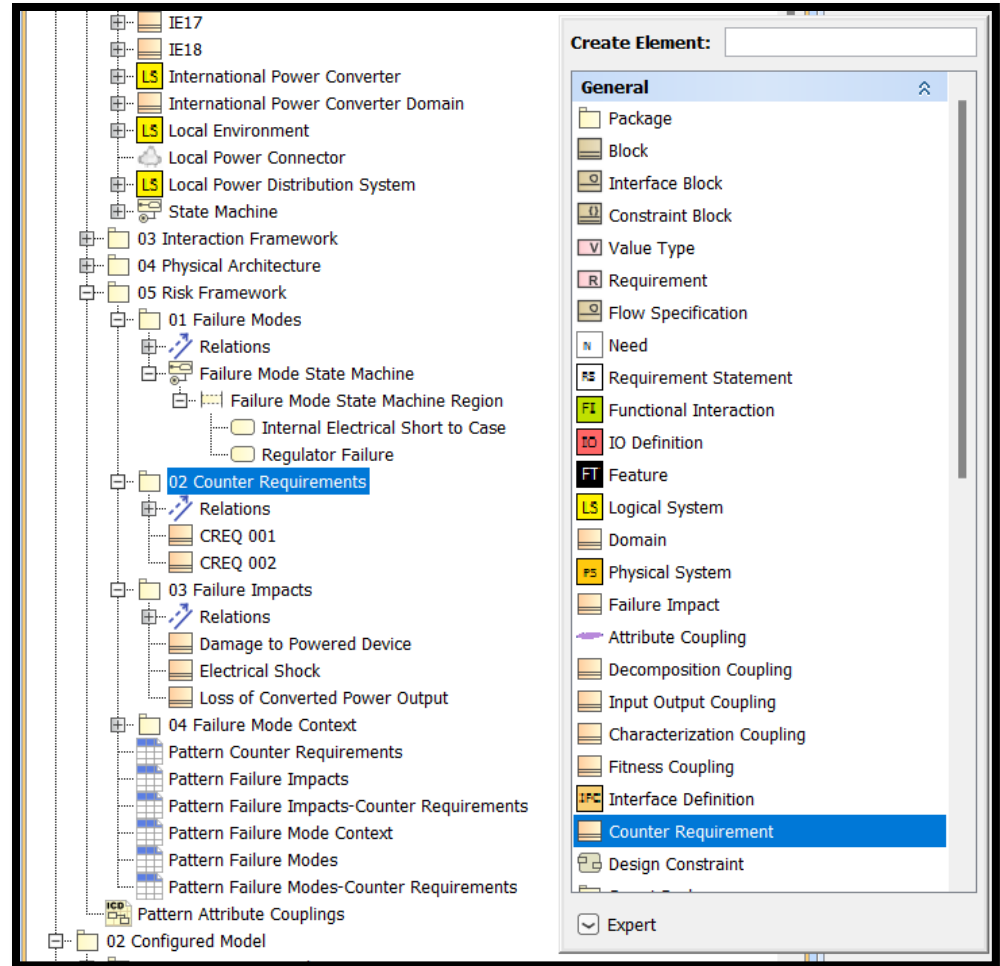
Failure Analysis: Creating Failure Impacts

- Pattern Failure Impacts are created in package 01 Local Pattern:::05 Risk Framework:::03 Failure Impacts
- Use ~Create Element and choose Failure Impact



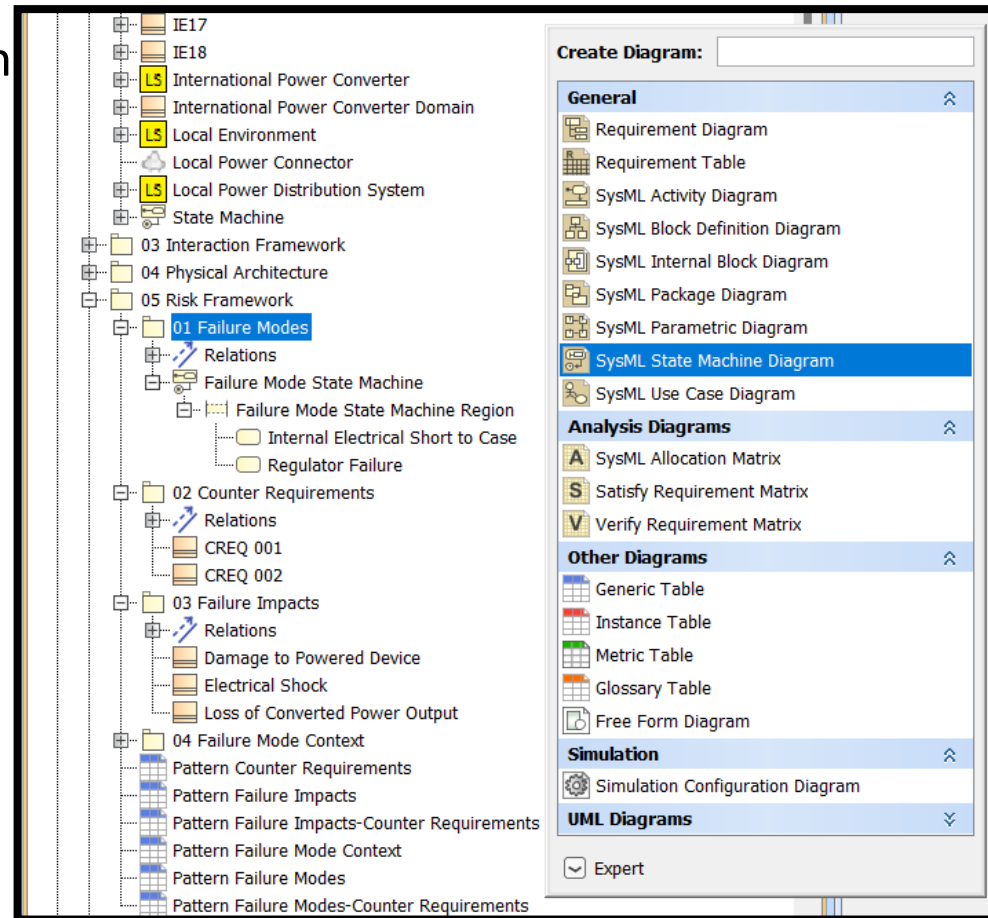
Failure Analysis: Creating Counter Requirements

- Pattern Failure Counter Requirements are created in package 01 Local Pattern::05 Risk Framework::02 Counter Requirements
- Use ~Create Element and choose Counter Requirement



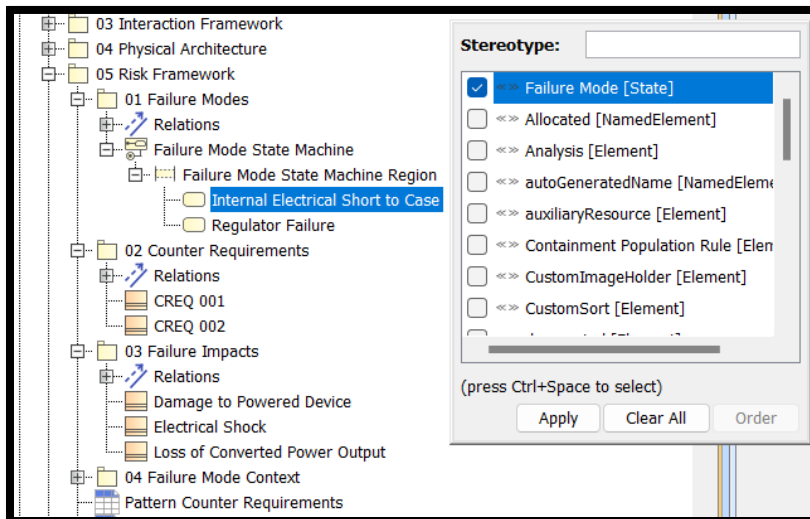
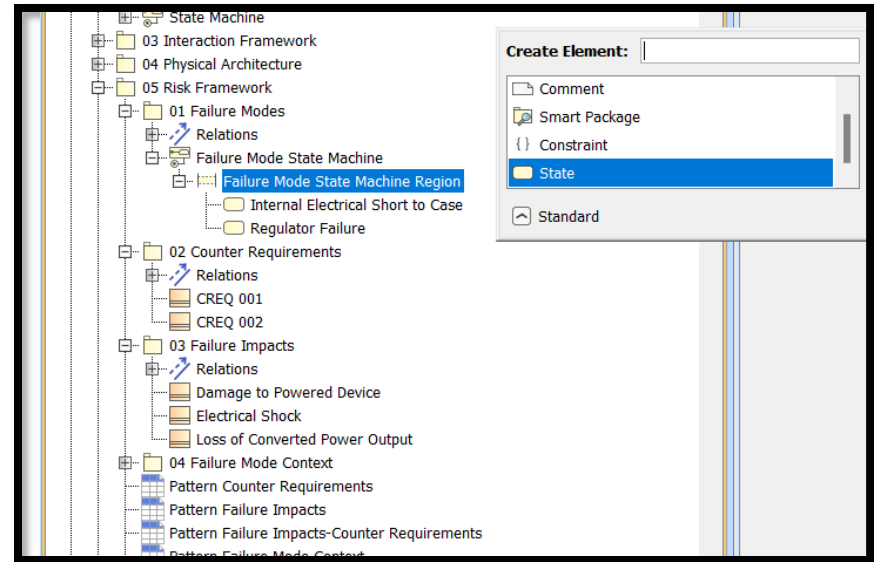
Failure Analysis: Creating Failure Mode State Machine and Failure Modes

- Pattern Failure Mode State Diagram and Failure Modes are created 01 Local Pattern::05 Risk Framework::01 Failure Modes
- Use ~Create Diagram and choose SysML State Machine Diagram
- Name the Diagram “Failure Mode State Machine”.
- Name the Region “Failure Mode State Machine Region”.



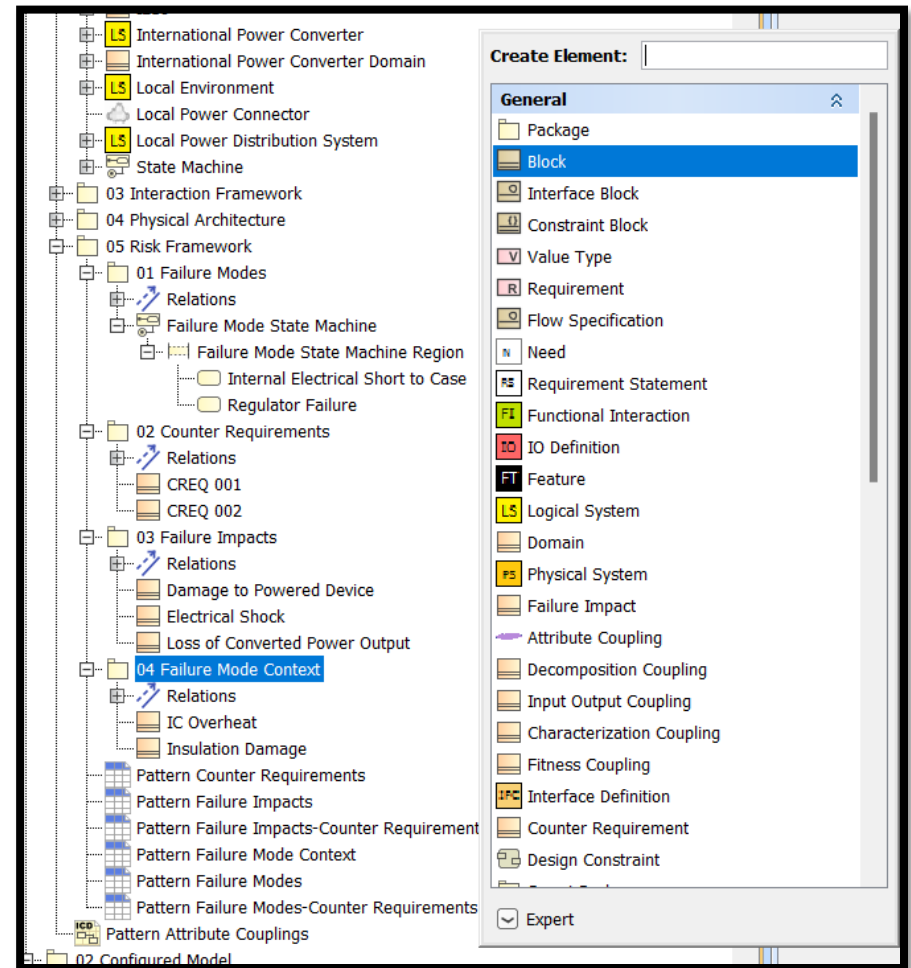
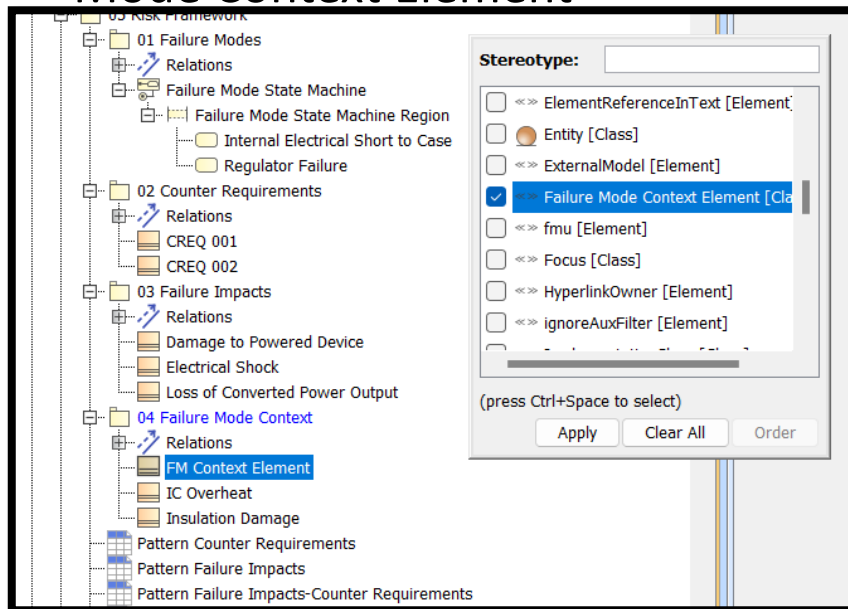
Failure Analysis: Creating Failure Modes

- Right-click the automatically created Region under the State Model and use ~Create Element and choose State. Make sure the menu is expanded to Expert.
- Right-click the newly created State, select Stereotype, and change it to Failure Mode as shown below.



Failure Analysis: Creating Failure Mode Context

- Pattern Failure Mode Context Elements are created in package 01 Local Pattern:::05 Risk Framework:::04 Failure Mode Context
- Use ~Create Element and choose Block
- Right-Click on the new element, and change stereotype to Failure Mode Context Element



4. Creating S* Pattern Relationships

This section of the guide details the steps for creating formally modeled S* Pattern Relationships (metarelationships) between S* Pattern Classes as mapped and implemented in Cameo Systems Modeler.

Metarelationships: Creating Stakeholder-Feature Relationships

- The simplest way to create relationships is using a matrix.
- Create a Dependency Matrix with Row Type, Row Scope, Column Type, Column Scope, and Dependency Criteria as shown.
- Right-click to create a new Stakeholder-Feature relationship.

The screenshot displays a dependency matrix with the following configuration:

- Criteria:** Row Element Type: Stakeholder, Column Element Type: Feature
- Row Scope:** 01 Stakeholders
- Column Scope:** 04 Features
- Dependency Criteria:** Benefits (FTR-STK)
- Direction:** Both
- Show Elements:** All

The matrix shows the following data points (Row: Stakeholder, Column: Feature):

Stakeholder	Automatic Braking System Feature	Commercial Vehicle Application Feature Group	Consumables Compatibility Feature	Cost of Operation Feature	Environmental Compatibility Feature	Military Vehicle Application Feature Group	Passenger Comfort Feature Group	Personal Vehicle Application Feature Group	Reliability & Availability Feature	Safety Feature Group	Traction Control Feature	Vehicle Aesthetics Feature Group	Vehicle Delivery Feature	Accountability Feature	Communications Feature Group	Configurability Feature	Maintainability Feature	Operability Feature	Cruise Control Feature	Navigation Feature	Remote-Autonomous Operation Feature	Remote Management Access Feature	Security Feature	Vehicle Performance Feature
Local Community	1																							
Passenger	8																							
Pedestrian	3																							
Vehicle Maintainer	2																							
Vehicle Operator	14																							
Vehicle Owner	17																							
Vehicle Supplier Shareholder	1																							

The context menu for the 'Vehicle Supplier Shareholder' row shows the following options:

- Create New (Row To Column)**
 - Dependency «Benefits (FTR-STK)»
- Create New (Column To Row)**
 - Dependency «Benefits (FTR-STK)»

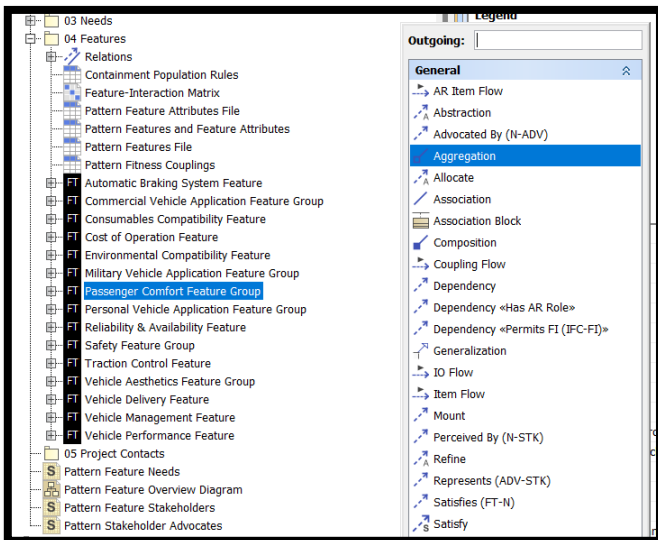
Metarelationships:

Creating Feature-Interaction Relationships

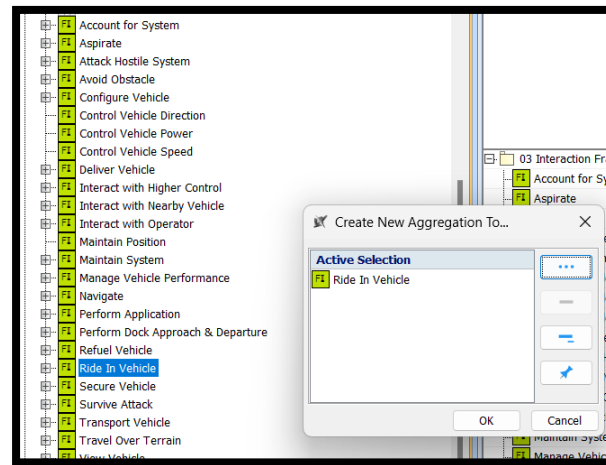
- Relationships between Features and Functional Interactions are SysML aggregations.
- Create an Association in which the Feature aggregates the Interaction.
- Add a “Uses Functional Interaction” stereotype to the new Association

Metarelationships: Creating Feature-Interaction Relationships

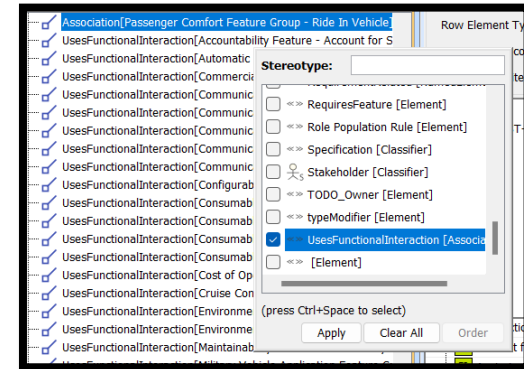
1. Right-Click the Feature, ~Create Relation, Select Outgoing, Choose Aggregation
2. Select the related Interaction(s)
3. Right-Click the new relationship, ~Stereotype, Select "Uses Functional Interaction"



1



2



3

Metarelationships: Creating Feature-Interaction Relationships

- The simplest way to create relationships is using a matrix.
- Create a Dependency Matrix with Row Type, Row Scope, Column Type, Column Scope, and Dependency Criteria as shown.
- Right-click to create a new Feature-Interaction relationship.

The screenshot shows the 'Feature-Interaction Matrix' window in Cameo Systems Modeler V19. The left pane displays a project tree with '04 Features' selected. The main window shows a matrix with '03 Interaction Framework' as the row scope and '04 Features' as the column scope. A context menu is open over a cell, showing 'Create New (Column To Row)' and 'UsesFunctionalInteraction'.

Row Element Type: Functional Interaction	Automatic braking	Commercial Vehicle	Consumables Com	Cost of Operation	Environmental Co	Military Vehicle Ap	Passenger Comfort	Personal Vehicle A	Reliability & Availi	Safety Feature Gr	Traction Control F	Vehicle Aesthetics	Vehicle Delivery F	Vehicle Management Feature	Accountability	Communicative	Configurability	Operability	Cruise Con	Navigator	Remobe-A	Remote Mana	Security Feat	Vehicle Performa
Row Scope: 03 Interaction Framework	1	1	2	1	2	5	1	1	1	1	1	2	1	3	1	1	2	1	2	1	2	1	3	
Dependency Criteria: UsesFunctionalInteraction																								

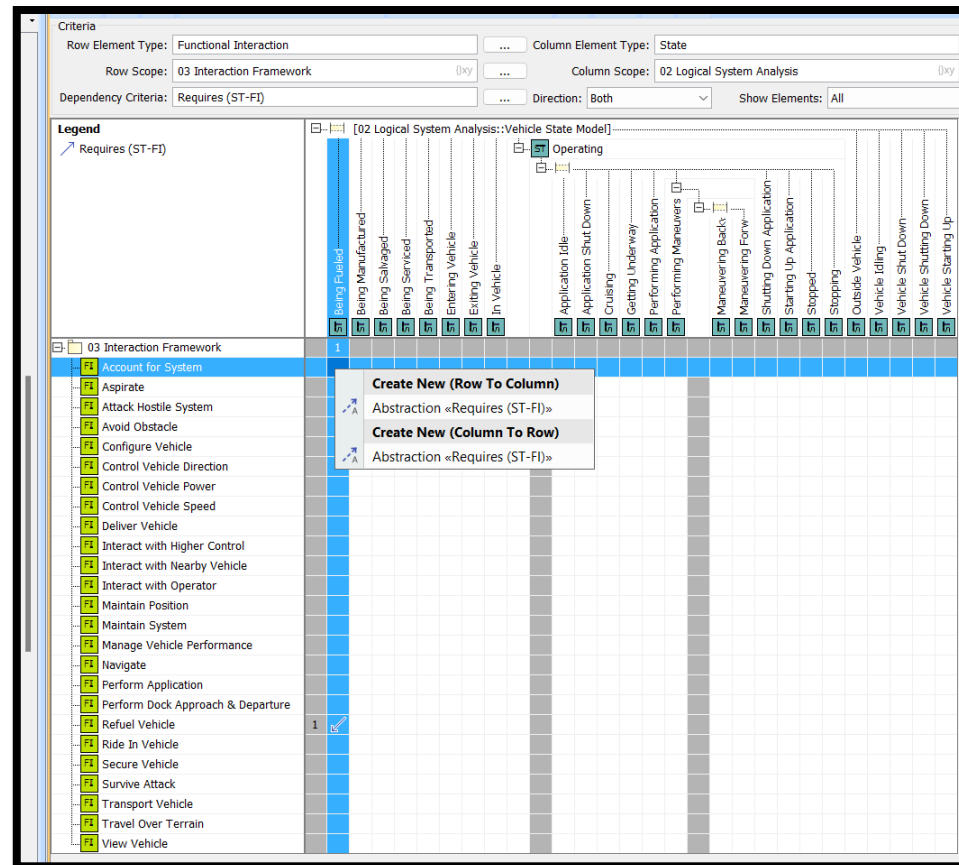
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

Other SysML relationship types can be entered and viewed via a matrix.

V2021x supports entering all relationship types via a matrix.

Metarelationships: Creating State-Interaction Relationships

- The simplest way to create relationships is using a matrix.
- Create a Dependency Matrix with Row Type, Row Scope, Column Type, Column Scope, and Dependency Criteria as shown.
- Right-click to create a new State-Interaction relationship.



*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

Other SysML relationship types can be entered and viewed via a matrix.

V2021x supports entering all relationship types via a matrix.

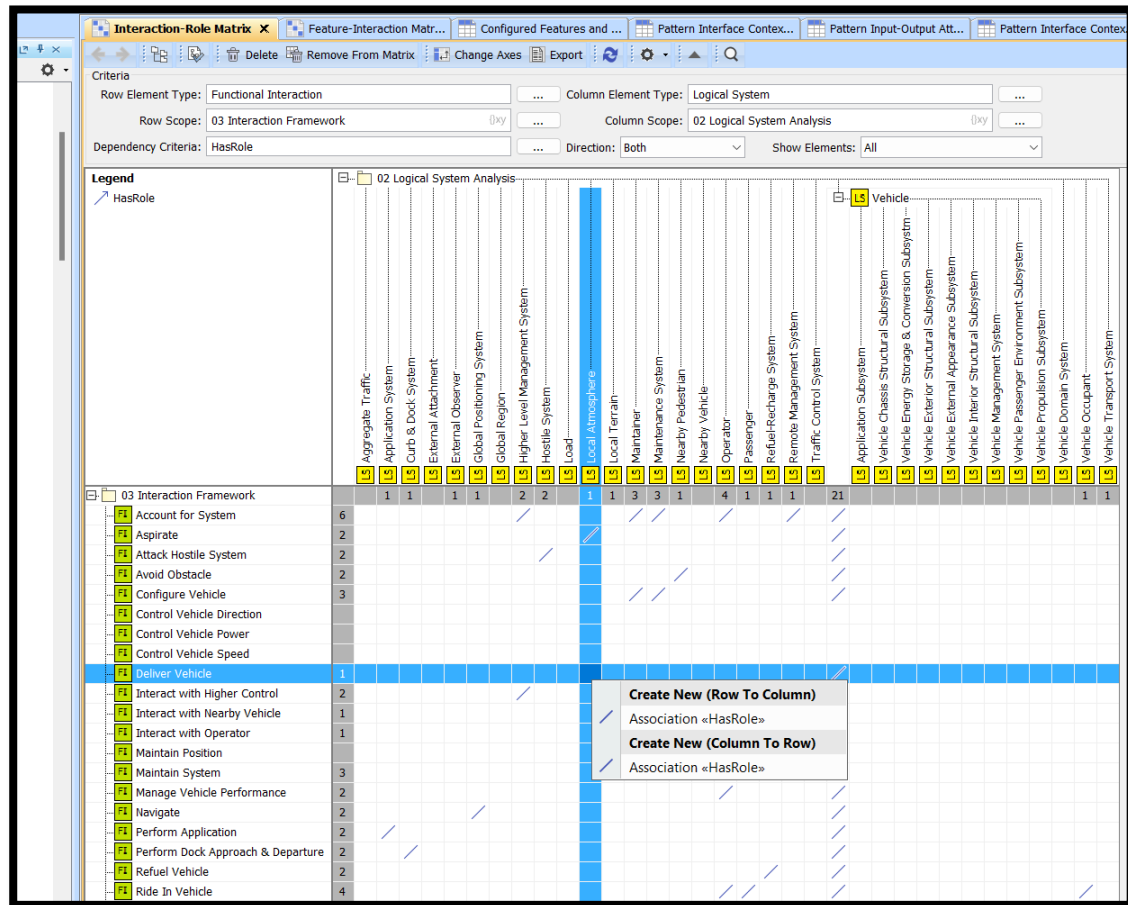
Metarelationships:

Creating Interaction-Role Relationships

- Relationships between Functional Interactions and Logical Systems are SysML aggregations.
- Create an Association in which the Interaction aggregates the Logical System.
- Add a “Has Role” stereotype to the new Association

Metarelationships: Creating Interaction-Role Relationships

- The simplest way to create relationships is using a matrix.
- Create a Dependency Matrix with Row Type, Row Scope, Column Type, Column Scope, and Dependency Criteria as shown.
- Right-click to create a new Interaction-Role relationship.



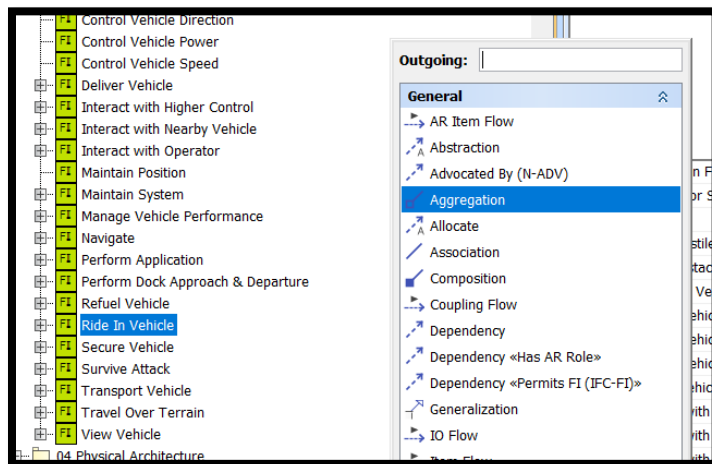
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

Other SysML relationship types can be entered and viewed via a matrix.

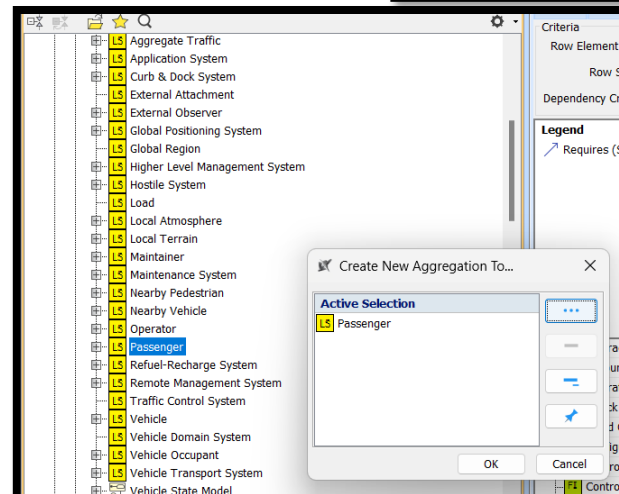
V2021x supports entering all relationship types via a matrix.

Metarelationships: Creating Interaction-Role Relationships

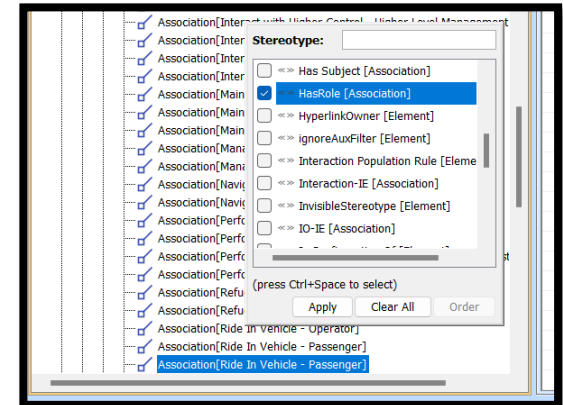
1. Right-Click the Interaction, ~Create Relation, Select Outgoing, Choose Aggregation
2. Select the related Role(s)
3. Right-Click the new relationship, ~Stereotype, Select "Has Role"



1



2

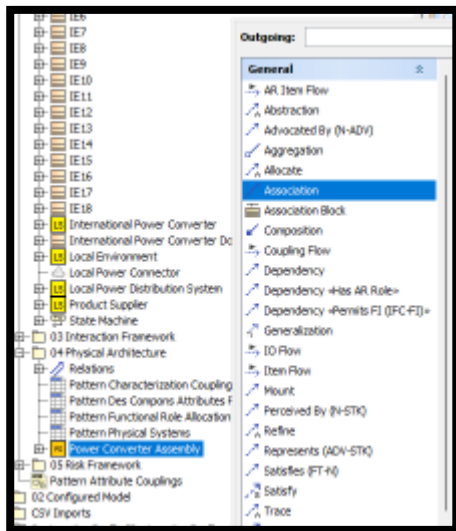


3

Metarelationships:

Creating Functional Role Allocations (Role-Design)

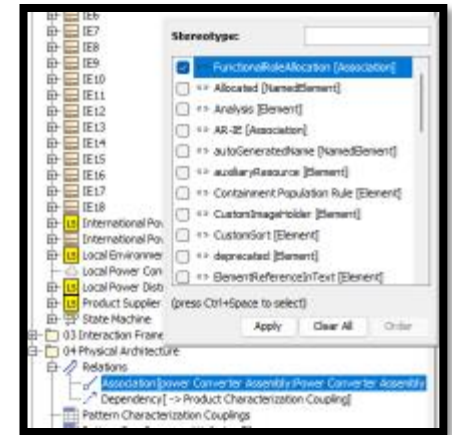
1. Relationships between Design Components (Physical Systems) and Logical Systems are SysML Associations.
2. Create an Association in which the Design Components (Physical System) aggregates the Logical System.
3. Add a “Functional Role Allocation” stereotype to the new Association



1

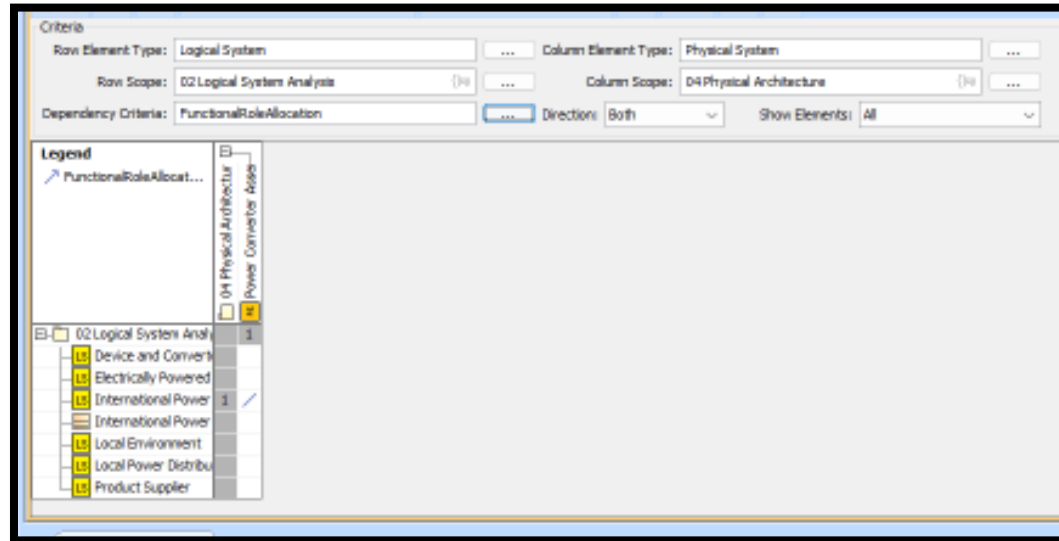


2



3

Metarelationships: Creating Functional Role Allocations (Role-Design)



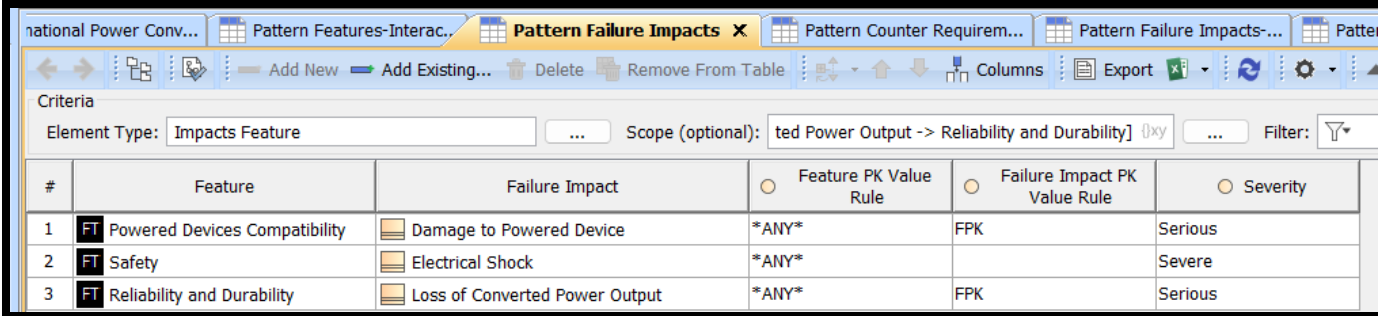
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

Other SysML relationship types can be entered and viewed via a matrix.

V2021x supports entering all relationship types via a matrix.

Metarelationships: Creating Failure Analysis Relationships—Impacts Feature

- Create a Dependency relationship with the Failure Impact as the Source and the Feature as the Target.
- Add a “Impacts Feature” stereotype to the new relationship

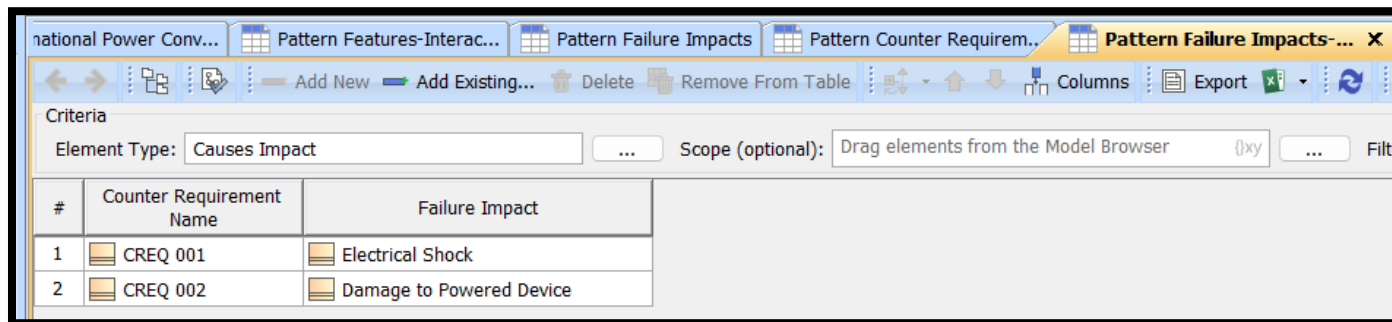


#	Feature	Failure Impact	Feature PK Value Rule	Failure Impact PK Value Rule	Severity
1	FT Powered Devices Compatibility	Damage to Powered Device	*ANY*	FPK	Serious
2	FT Safety	Electrical Shock	*ANY*		Severe
3	FT Reliability and Durability	Loss of Converted Power Output	*ANY*	FPK	Serious

Example Pattern Failure Impacts Table

Metarelationships: Creating Failure Analysis Relationships—Causes Impact

- Create a Dependency relationship with the Counter Requirement as the Source and the Failure Impact as the Target.
- Add a “Causes Impact” stereotype to the new relationship



#	Counter Requirement Name	Failure Impact
1	CREQ 001	Electrical Shock
2	CREQ 002	Damage to Powered Device

Example Pattern Failure Impacts-Counter Requirements Table

Metarelationships: Creating Failure Analysis Relationships— Replaces

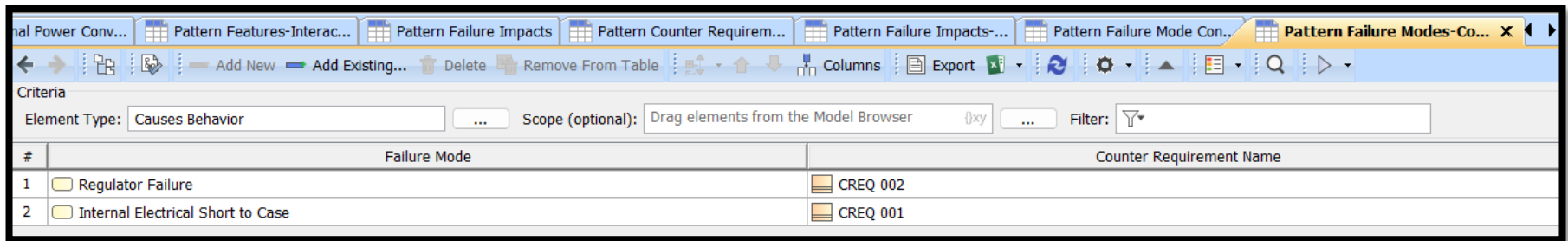
- Create a Dependency relationship with the Counter Requirement as the Source and the Requirement as the Target.
- Add a “Replaces” stereotype to the new relationship.

#	Requirement Name	Counter Requirement Name	Counter Requirement Statement	<input type="radio"/> RSPK Value Rule	<input type="radio"/> Counter Requirement PK Value Rule
1	REQ 005	CREQ 001	The system presents a shock hazard to users when operated according to its instructions.	*ANY*	RSPK
2	REQ 006	CREQ 002	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	*ANY*	RSPK

Example Pattern Counter Requirements Table

Metarelationships: Creating Failure Analysis Relationships—Causes Behavior

- Create a Dependency relationship with the Failure Mode as the Source and the Counter Requirement as the Target.
- Add a “Causes Behavior” stereotype to the new relationship.



The screenshot shows a software interface with a table titled "Pattern Failure Modes-Counter Requirements". The table has two columns: "Failure Mode" and "Counter Requirement Name". There are two rows of data:

#	Failure Mode	Counter Requirement Name
1	Regulator Failure	CREQ 002
2	Internal Electrical Short to Case	CREQ 001

Example Pattern Failure Modes-Counter Requirements Table

Metarelationships: Creating Failure Analysis Relationships—Abnormal State Of

- Create a Dependency relationship with the Failure Mode as the Source and the Design Component as the Target.
- Add a “Abnormal State Of” stereotype to the new relationship.

Criteria
Element Type: Abnormal State Of ... Scope (optional): Drag elements from the Model Browser (jxy) ... Filter:

#	<input type="radio"/> DCPK Value Rule	<input type="radio"/> Probability	<input type="radio"/> FMPK Value Rule	Design Component	Failure Mode
1	*ANY*	0.0002	CRPK	Power Converter Assembly	<input type="checkbox"/> Regulator Failure
2	*ANY*	0.0003	DCPK	Power Converter Assembly	<input type="checkbox"/> Internal Electrical Short to Case

Example Pattern Failure Modes Table

Metarelationships:

Creating Failure Analysis Relationships—Provides Failure Mode Context

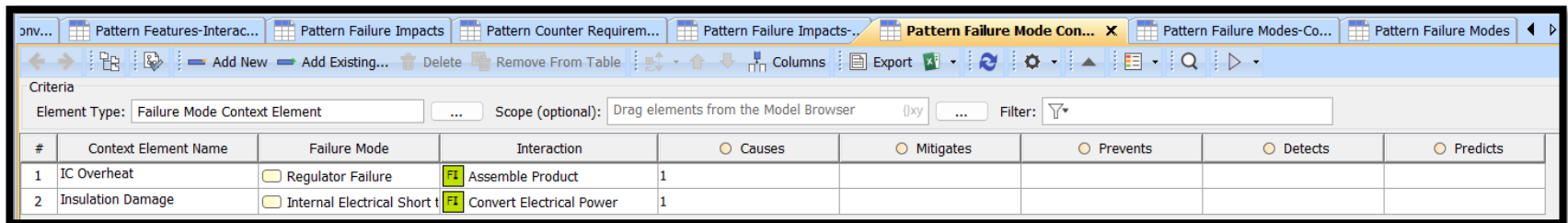
- Create a Dependency relationship with the Failure Mode Context Element as the Source and the Failure Mode as the Target.
- Add a “Provides Failure Mode Context” stereotype to the new relationship.

#	Context Element Name	Failure Mode	Interaction	<input type="radio"/> Causes	<input type="radio"/> Mitigates	<input type="radio"/> Prevents	<input type="radio"/> Detects	<input type="radio"/> Predicts
1	IC Overheat	<input type="checkbox"/> Regulator Failure	<input checked="" type="checkbox"/> Assemble Product	1				
2	Insulation Damage	<input type="checkbox"/> Internal Electrical Short t	<input checked="" type="checkbox"/> Convert Electrical Power	1				

Example Pattern Failure Mode Context Table

Metarelationships: Creating Failure Analysis Relationships—Provides Failure Context

- Create a Dependency relationship with the Failure Mode Context Element as the Source and the Interaction as the Target.
- Add a “Provides Failure Context” stereotype to the new relationship.



The screenshot shows a software interface with a table titled "Pattern Failure Mode Context". The table has columns for "#", "Context Element Name", "Failure Mode", "Interaction", and five relationship types: "Causes", "Mitigates", "Prevents", "Detects", and "Predicts". Two rows are visible in the table, each with a value of "1" in the "Causes" column.

#	Context Element Name	Failure Mode	Interaction	Causes	Mitigates	Prevents	Detects	Predicts
1	IC Overheat	Regulator Failure	Assemble Product	1				
2	Insulation Damage	Internal Electrical Short t	Convert Electrical Power	1				

Example Pattern Failure Mode Context Table

5. Creating S* Pattern Configuration Rules

This section of the guide details the steps and rules that govern when and how certain pattern metaclasses are to be automatically populated by the Configuration Wizard during Pattern Configuration based on the population of another metaclass(es), using the metarelationships and details described in this section.

Population Rules: Editing Interaction Population Rules

- The “Pattern Feature Interactions” table under the Interaction Framework package allows the user to edit the Interaction Population Rules
- The values in the FPK Value and IPK Rule columns govern when and how Interactions are populated based on which Features have been populated.
- The FPK Value may be blank, *ANY*, or <constant> and helps decide if an Interaction should be populated
- The IPK Rule may be blank, FPK, /<constant>/, or FPK+/<constant>/ and describes how to generate the IPK Value of the Interaction to be populated.
- An IPK Rule of *ANY* will tell the Configuration Agent not to populate an Interaction but to populate a relationship between the Feature and Interaction if they are both populated from other population rules.
- A detailed list of Population Rules is in the Metamodel document reference.

#	Feature	FPK Value	Interaction	IPK Rule
1	Ease of Purchase		Purchase Device	
2	Ease of Use		Display Status and Information	
3	Ease of Use		Handle Device	
4	Environmentally Friendly		Recycle Device	
5	Portability		Handle Device	
6	Power Mains Compatibility		Consume Electrical Power	
7	Powered Devices Compatibility	*ANY*	Convert Electrical Power	FPK
8	Powered Devices Compatibility	*ANY*	Set Configuration	FPK
9	Reliability and Durability		Resist Contamination	
10	Reliability and Durability		Tolerate Moisture	
11	Reliability and Durability		Provide Warranty Data	
12	Safety	*ANY*	Handle Device	

Population Rules: Editing State Population Rules

- The “Pattern Interactions-States” table under the State Machine allows the user to edit the State Population Rules
- The values in the IPK, RPK, and State PK Value Rule columns govern when and how States are populated based on which Interactions and Roles have been populated
- A detailed list of Population Rules is in the Metamodel document reference.

#	State	Interaction	○ IPK	Logical System	○ RPK	○ State PK Value Rule	○ State Type
1	ST Operating	PT Convert Electrical Power	*ANY*	LS International Power Converter	*ANY*	IPK	Simple
2	ST Being Purchased	PT Purchase Device		LS International Power Converter	*ANY*		Simple
3	ST Being Recycled	PT Recycle Device		LS International Power Converter	*ANY*		Simple
4	ST In Service	PT Consume Electrical Power		LS International Power Converter	*ANY*		Simple
5	ST In Service	PT Display Status and Information		LS International Power Converter	*ANY*		Simple
6	ST In Service	PT Handle Device		LS International Power Converter	*ANY*		Simple
7	ST In Service	PT Resist Contamination		LS International Power Converter	*ANY*		Simple
8	ST In Service	PT Set Configuration	*ANY*	LS International Power Converter	*ANY*		Simple
9	ST In Service	PT Tolerate Moisture		LS International Power Converter	*ANY*		Simple
10	ST Being a Warranty Record	PT Provide Warranty Data		LS International Power Converter	*ANY*		Simple
11	ST Off	PT Convert Electrical Power	*ANY*	LS International Power Converter	*ANY*	IPK	Empty
12	ST Initial IPC	PT Purchase Device		LS International Power Converter	*ANY*		Initial
13	ST Terminal IPC	PT Recycle Device		LS International Power Converter	*ANY*		Terminal

Population Rules: Editing State Population Rules, Continued

- The “Pattern States-Transitions, Events” table under the State Machine allows the user to edit the Transition, Event Population Rules
- The values in the From State PK Matching Rule, To State PK Matching Rule, Interaction, Transition PK Value Rule, and Event PK Value Rule columns govern when and how Transitions and Events are populated based on which Interactions and States have been populated.
- A detailed list of Population Rules is in the Metamodel document reference.

#	From State	To State	Event	Interaction	○ Transition Type	○ Join Transition Name	○ Transition PK Value Rule	○ Event PK Value Rule	○ From State PK Matching Rule	○ To State PK Matching Rule
1	ST Off	ST Operating	Turn On Request	FI Set Configuration	Simple			IPK	EPK	EPK
2	ST Off	ST Being Recycled	Start Retirement	FI Recycle Device	Join	Wait for Retire			*ANY*	*ANY*
3	ST Operating	ST Off	Turn Off Request	FI Set Configuration	Simple			IPK	EPK	EPK
4	ST Being Purchased	ST Being a Warranty Record	Device Purchased	FI Purchase Device	Fork				*ANY*	*ANY*
5	ST Being Purchased	ST Off	Device Purchased	FI Purchase Device	Fork				*ANY*	*ANY*
6	ST Being a Warranty Record	ST Being Recycled	Complete Retirement	FI Recycle Device	Join	Wait for Retire			*ANY*	*ANY*
7	ST Initial IPC	ST Being Purchased	Request Purchase	FI Purchase Device	Simple				*ANY*	*ANY*
8	ST Being Recycled	ST Terminal IPC	Recycling Completed	FI Recycle Device	Simple				*ANY*	*ANY*

Population Rules: Editing Role Population Rules

- The “Pattern Interaction Roles” table under the Interaction Framework package allows the user to edit the Role Population Rules
- The values in the IPK Value and RPK Rule columns govern when and how Logical Systems are populated based on which Interactions have been populated.
- The IPK Value may be blank, *ANY*, or <constant> and helps decide if a Logical System should be populated.
- The RPK Rule may be blank, IPK, /<constant>/, or IPK+/<constant>/ and describes how to generate the RPK Value of the Logical System to be populated.
- A detailed list of Population Rules is in the Metamodel document reference.













#	△ Interaction	○ IPK Value	Logical System	○ RPK Rule
1	Consume Electrical Power	*ANY*	International Power Converter	
2	Consume Electrical Power	*ANY*	Local Power Distribution System	
3	Convert Electrical Power	*ANY*	Electrically Powered Device	IPK
4	Convert Electrical Power	*ANY*	International Power Converter	
5	Convert Electrical Power	*ANY*	Local Environment	
6	Display Status and Information	*ANY*	International Power Converter	
7	Display Status and Information	*ANY*	Device and Converter User	
8	Handle Device	*ANY*	International Power Converter	
9	Handle Device	*ANY*	Device and Converter User	
10	Provide Warranty Data	*ANY*	Product Supplier	
11	Provide Warranty Data	*ANY*	Device and Converter User	
12	Purchase Device	*ANY*	Device and Converter User	
13	Purchase Device	*ANY*	International Power Converter	
14	Recycle Device	*ANY*	Product Supplier	
15	Recycle Device	*ANY*	International Power Converter	
16	Recycle Device	*ANY*	Local Environment	
17	Resist Contamination	*ANY*	International Power Converter	
18	Resist Contamination	*ANY*	Local Environment	
19	Set Configuration	*ANY*	International Power Converter	
20	Set Configuration	*ANY*	Device and Converter User	
21	Tolerate Moisture	*ANY*	International Power Converter	
22	Tolerate Moisture	*ANY*	Local Environment	

Population Rules:

Editing Requirement Population Rules

- The “Pattern Requirements” table under the Interaction Framework package allows the user to edit the Requirement Population Rules
- The values in the IPK Value, RPK Value, and RSPK Rule columns govern when and how Requirement Statements are populated based on which Interactions and Logical Systems have been populated.
- The IPK Value may be blank, *ANY*, or <constant> and helps decide if a Requirement Statement should be populated.
- The RPK Value may be blank, *ANY*, or <constant> and helps decide if a Requirement Statement should be populated.
- The RSPK Rule may be blank, IPK, RPK, /<constant>/, or IPK+/<constant>/, or RPK+/<constant>/ and describes how to generate the RSPK Value of the Requirement Statement to be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

Population Rules: Editing Requirement Population Rules

#	Functional Interaction	IPK Value	Functional Role	RPK Value	Req ID	RSPK Rule	Text
1	 Consume Electrical Power		 International Power Converter		REQ 001		The system shall consume not more than [Max Drain on Mains] of Input Power from the Local Power Distribution System.
2	 Convert Electrical Power	*ANY*	 International Power Converter	*ANY*	REQ 002	IPK	The system shall provide up to [Max Power Drain] of Output Power to each Electrically Powered Device.
3	 Convert Electrical Power	*ANY*	 Electrically Powered Device	*ANY*	REQ 003	IPK	The system shall not consume more than [Max Power Drain] of Output Power.
4	 Consume Electrical Power		 Local Power Distribution System		REQ 004		The system shall provide up to [Max Drain on Mains] of Input Power.
5	 Handle Device		 International Power Converter		REQ 005		The system shall not present a shock hazard to users when operated according to its instructions.
6	 Convert Electrical Power	*ANY*	 International Power Converter	*ANY*	REQ 006	IPK	The system shall generate Output Power to attached Electrically Powered Devices which is compatible with the [Output Voltage-Power Profile] and [Output Frequency Profile].

Population Rules: Editing Physical System (Design Component) Population Rules

- The “Pattern Functional Role Allocations” table under the Physical Architecture package allows the user to edit the Physical System (Design Component) Population Rules
- The values in the Configuration Rule and IPPK Value columns govern when and how Physical Systems (Design Components) are populated based on which Logical Systems (Roles) have been populated.
- The Configuration Rule may be blank, *ANY*, or <constant> and helps decide if a Physical System (Design Component) should be populated.
- The IPPK Rule may be blank, RPK, /<constant>/, or RPK+/<constant>/ and describes how to generate the IPPK Value of the Physical System (Design Component) to be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

#	Design Component	○ IPPK Value	Role	○ Configuration Rule
1	Power Converter Assembly		International Power Converter	

Population Rules: Table ICT1 Population Rules

- The “Interface Context Table ICT1” under the Local Pattern package allows the user to edit the Interface Context Population Rules
- The values in the Interface PK Rule, IO PK Rule, SOA PK Rule and Port PK Rule columns govern when and how those elements are populated based on which Interactions and Roles have been populated.
- A detailed list of Population Rules is in the Metamodel document reference.

#	System Name	△ Interface Name	Interface PK Rule	Input Output	IO PK Rule	Direction	Port	Port PK Rule	SOA Name	SOA PK Rule	SOA Internal or External
1	International Power Converter	Configuration and Information Interface		Configuration Setting		in	CI.1	IFPK	HMI Buttons		
2	International Power Converter	Configuration and Information Interface		Device Status and Information		out	CO.2	IFPK	HMI Display		
3	Local Environment	Contaminant Source Interface		Contaminants		out	ENV.1	IFPK	Airspace Transport System		
4	Electrically Powered Device	Device Power Interface		Output Power	IPK	in	DPI.1	IFPK	Device Power Connector	IPK	
5	Local Environment	EMI Receiver Interface		EMI		in	ENV.3	IFPK	EMI Radiation		
6	Local Environment	Environment Thermal Sink Interface		Heat		in	ENV.4	IFPK	Airspace Convection System		
7	International Power Converter	Environmental Interface		Heat		in	EN.1	IFPK	Airspace Convection System		
8	International Power Converter	Environmental Interface		Contaminants		in	EN.3	IFPK	Airspace Transport System		
9	International Power Converter	Environmental Interface		Moisture		in	EN.4	IFPK	Airspace Transport System		
10	International Power Converter	Environmental Interface		EMI		out	EN.2	IFPK	EMI Radiation		
11	Device and Converter User	Finger Interface		Configuration Setting		out	UCO.1	IFPK	HMI Buttons		
12	Device and Converter User	Hand Interface		Handling Force		inout	UHO.1	IFPK	Handle		
13	International Power Converter	Handling Interface		Handling Force		inout	HI.1	IFPK	Handle		
14	Local Environment	Moisture Source Interface		Moisture		out	ENV.2	IFPK	Airspace Transport System		
15	International Power Converter	Power Input Interface		Input Power		in	PI.1		Local Power Connector		
16	Local Power Distribution System	Power Mains Interface		Input Power		out	LPO.1	IFPK	Local Power Connector		
17	International Power Converter	Power Output Interface	IPK	Output Power	IPK	out	PO.1	IFPK	Device Power Connector	IPK	
18	Device and Converter User	Vision Interface		Device Status and Information		in	UC1.1	IFPK	HMI Display		

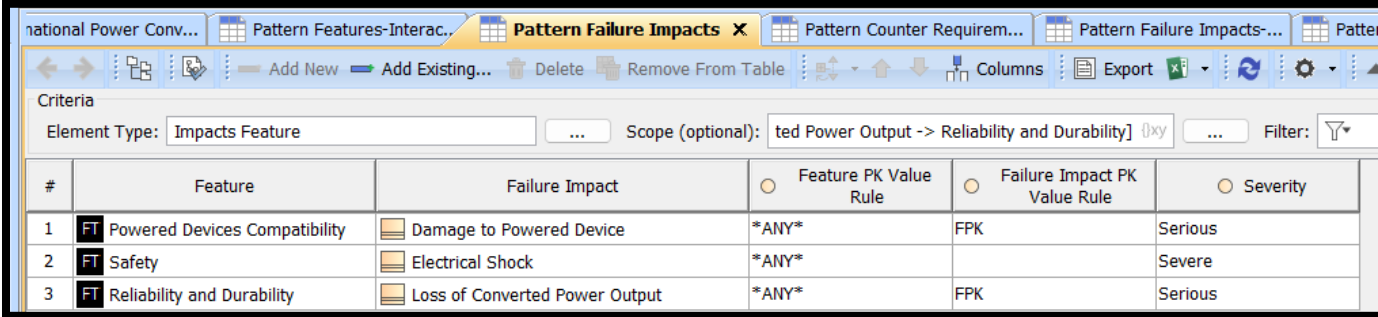
Population Rules: Editing Attribute Coupling Population Rules

- The “Pattern Attribute Couplings” table under the Local Pattern package allows the user to edit the Attribute Coupling Population Rules
- The values in the PK column govern when and how Attribute Couplings are populated based on which related driven Attributes have been populated.
- The value in the PK column may be $APK < CPK$, $APK = CPK$, $CPK < APK$, *ANY* or (empty) and helps decide if an Attribute Coupling should be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

#	Coupling Name	Coupling Direction	Attributed Class Name	Metaclass of Attributed Class	Port A	Coupling PK Value Rule
1	Product Characterization Coupling	in	 Power Converter Assembly	 Physical System [Class]	<input type="checkbox"/> Part Number	
2	Max Power Coupling	in	 International Power Converter	 Logical System [Class]	<input type="checkbox"/> Max Power Drain	
3	Max Power Coupling	out	 Power Mains Compatibility	 Feature [Class]	<input type="checkbox"/> Max Drain on Mains	

Population Rules: Editing Failure Impact Population Rules

- The “Pattern Failure Impacts” table under the Local Pattern package allows the user to edit the Failure Impact Population Rules
- The values in the PK column govern when and how Failure Impacts are populated based on which related Features have been populated.
- The value in the Failure Impact PK Value Rule column may be FPK, *ANY* or (empty) and helps decide if a Failure Impact should be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

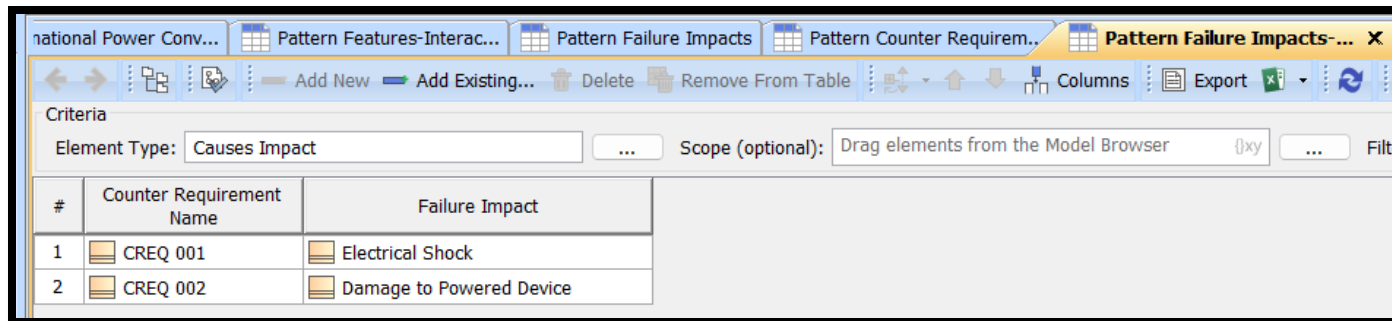


#	Feature	Failure Impact	Feature PK Value Rule	Failure Impact PK Value Rule	Severity
1	FT Powered Devices Compatibility	Damage to Powered Device	*ANY*	FPK	Serious
2	FT Safety	Electrical Shock	*ANY*		Severe
3	FT Reliability and Durability	Loss of Converted Power Output	*ANY*	FPK	Serious

Example Pattern Failure Impacts Table

Population Rules: Editing Causes Impact Relationship Population Rules

- The “Pattern Failure Impacts-Counter Requirements” table under the Local Pattern package allows the user to edit the Causes Impact Population Rules
- The table governs when and how Causes Impact relationships (associating Counter Requirements with the Failure Impacts they cause) are populated based on which related Failure Impacts and Counter Requirements have been populated.
- A detailed list of Population Rules is in the Metamodel document reference.



The screenshot shows a software window titled "Pattern Failure Impacts-Counter Requirements" with a toolbar containing "Add New", "Add Existing...", "Delete", "Remove From Table", "Columns", "Export", and "Filter". Below the toolbar, the "Criteria" section shows "Element Type: Causes Impact" and "Scope (optional): Drag elements from the Model Browser". The main table has the following data:

#	Counter Requirement Name	Failure Impact
1	CREQ 001	Electrical Shock
2	CREQ 002	Damage to Powered Device

Example Pattern Failure Impacts-Counter Requirements Table

Population Rules: Editing Counter Requirements Population Rules

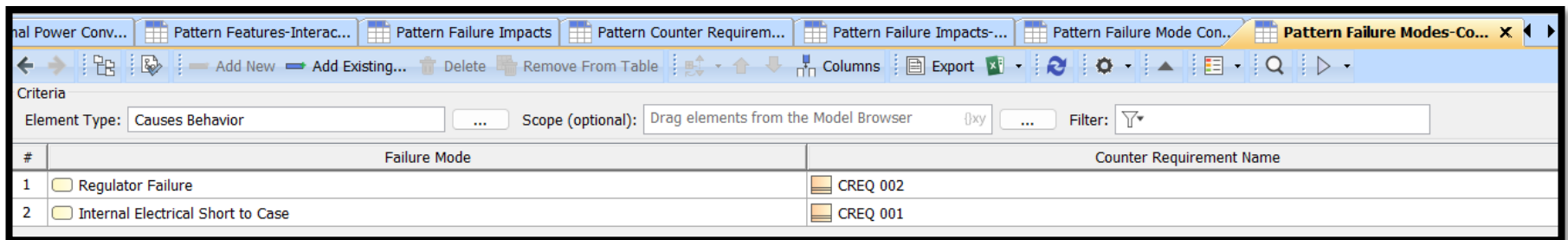
- The “Pattern Counter Requirements” table under the Local Pattern package allows the user to edit the Counter Requirement Population Rules
- The values in the PK column govern when and how Counter Requirements are populated based on which related Requirements have been populated.
- The value in the Counter Requirement PK Value Rule column may be RSPK, *ANY*, or (empty) and helps decide if a Counter Requirement should be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

#	Requirement Name	Counter Requirement Name	Counter Requirement Statement	<input type="radio"/> RSPK Value Rule	<input type="radio"/> Counter Requirement PK Value Rule
1	REQ 005	CREQ 001	The system presents a shock hazard to users when operated according to its instructions.	*ANY*	RSPK
2	REQ 006	CREQ 002	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	*ANY*	RSPK

Example Pattern Counter Requirements Table

Population Rules: Editing Causes Behavior Relationship Population Rules

- The “Pattern Failure Modes-Counter Requirements” table under the Local Pattern package allows the user to edit the Causes Behavior Relationship Population Rules
- The table governs when and how Causes Behavior relationships (associating Failure Modes with the Counter Requirements they cause) are populated based on which related Failure Modes and Counter Requirements have been populated.
- A detailed list of Population Rules is in the Metamodel document reference.



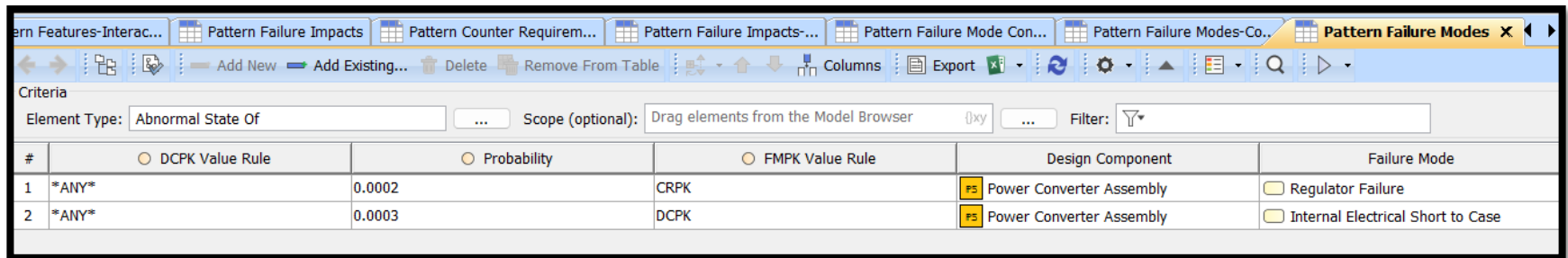
The screenshot shows a software interface with a table titled "Pattern Failure Modes-Counter Requirements". The table has two columns: "Failure Mode" and "Counter Requirement Name". The table contains two rows of data:

#	Failure Mode	Counter Requirement Name
1	Regulator Failure	CREQ 002
2	Internal Electrical Short to Case	CREQ 001

Example Pattern Failure Modes-Counter Requirements Table

Population Rules: Editing Failure Mode Population Rules

- The “Pattern Failure Modes” table under the Local Pattern package allows the user to edit the Failure Mode Population Rules
- The values in the PK column govern when and how Failure Modes are populated based on which Design Components have been populated.
- The value in the FMPK Value Rule column may be CRPK, DCPK, or *ANY*, or (empty) and helps decide if a Failure Mode should be populated.
- A detailed list of Population Rules is in the Metamodel document reference.

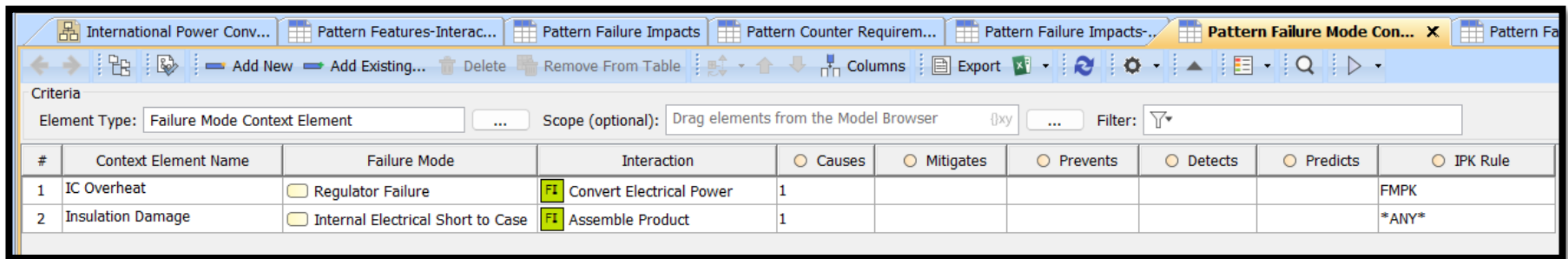


#	DCPK Value Rule	Probability	FMPK Value Rule	Design Component	Failure Mode
1	*ANY*	0.0002	CRPK	Power Converter Assembly	Regulator Failure
2	*ANY*	0.0003	DCPK	Power Converter Assembly	Internal Electrical Short to Case

Example Pattern Failure Modes Table

Population Rules: Editing Failure Mode Context Element Population Rules

- The “Pattern Failure Mode Context” table under the Local Pattern package allows the user to edit the Failure Mode Context Element Population Rules
- The table governs when and how Failure Mode Context Elements are populated based on which related Failure Modes and Interactions have been populated.
- Failure Mode Context Elements relate Interactions to the Failure Modes that they cause, mitigate, prevent, detect, or predict.
- The value in the IPK Rule column may be FMPK, IMPK, CRPK, *ANY*, or (empty) and helps decide if a Failure Mode Context Element should be populated.
- A detailed list of Population Rules is in the Metamodel document reference.



Criteria

Element Type: Failure Mode Context Element Scope (optional): Drag elements from the Model Browser Filter:

#	Context Element Name	Failure Mode	Interaction	<input type="radio"/> Causes	<input type="radio"/> Mitigates	<input type="radio"/> Prevents	<input type="radio"/> Detects	<input type="radio"/> Predicts	<input type="radio"/> IPK Rule
1	IC Overheat	<input type="checkbox"/> Regulator Failure	<input checked="" type="checkbox"/> Convert Electrical Power	1					FMPK
2	Insulation Damage	<input type="checkbox"/> Internal Electrical Short to Case	<input checked="" type="checkbox"/> Assemble Product	1					*ANY*

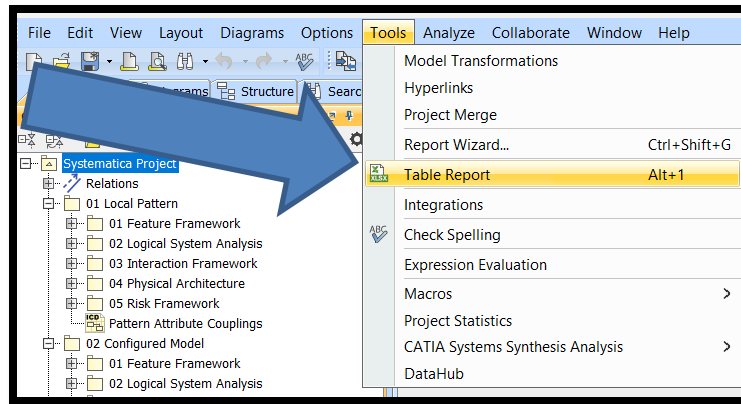
Example Pattern Failure Mode Context Table

6. Exporting and Importing Pattern and Model Data During S* Pattern Configuration

This section of the guide details the steps for (1) Exporting Pattern data from Cameo Systems Modeler into single or multiple output files for consumption by the Configuration Wizard and (2) Importing Configured Model data from the Pattern Configuration Wizard into Cameo Systems Modeler from multiple CSV files. Refer to the diagram on Slide 7 and the Configuration Wizard Guide.

Exporting Pattern Tables Using the Report Wizard

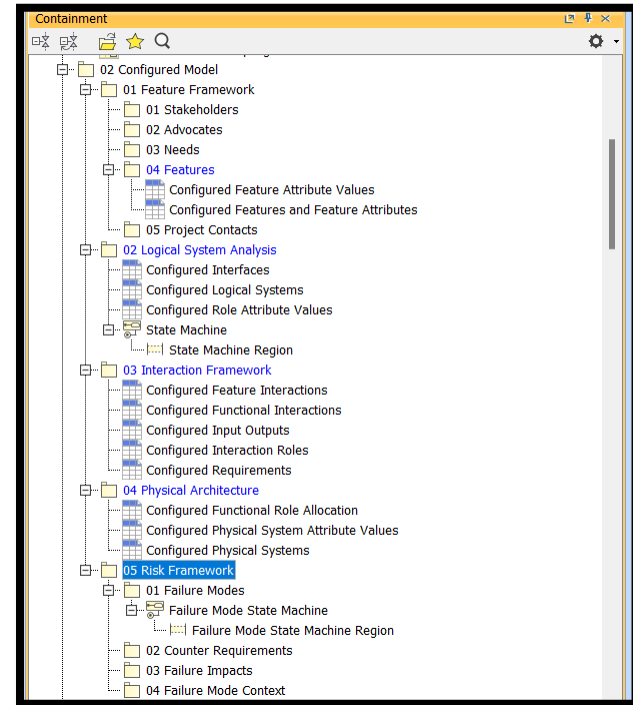
After defining the Export Tables Report as shown in Section 1, it can be invoked by selecting it from the Tools Menu.



If the Report does not work correctly, refer to Section 1 to define it again.

Emptying the Configured Model Packages Before the Import

- Prior to importing a configured model into the repository, any previous model content should be removed from there, leaving the basic framework elements in place. Once it is emptied of previous model data, the empty 02 Configured Model package of the Containment Browser should look like the example shown here. If it contains elements or relations other than what is shown here, those should be deleted, retaining the package structure and tables shown.



Importing the Configured Model Data

Choose the Import CSV option. If not available, see the Plug-In Manager to install it.

Select the Execute Map Group Option and choose the Map Group depending on your import needs.

- The Configured Model HLR DLR is the first one to select and imports a subset of the entire model including Features, Interactions, Roles, Requirements, and Design Components.
- Configured States includes importing States, Forks/Joins, Events, and Transitions.
- Configured IFC Context includes importing Interfaces, Input-Outputs, Architectural Relationships, Systems of Access, and Ports
- Configured Couplings includes Couplings, Constraint Parameters, and Connectors
- Configured Failure Analysis includes Counter Requirements, Failure Modes, and Failure Impacts
- ***The Configuration Wizard control panel allows for similar configuration grouping choices. (See guide)***

Importing the Configured Model Data

The screenshot displays a software application interface. On the left, a 'Containment' tree shows a hierarchical structure of project elements, including '01 Feature Framework', '02 Logical System Analysis', and '03 Interaction Framework'. On the right, a 'Notification Window' displays a log of import operations. A blue arrow points from the notification window back to the project tree. A text box at the bottom right explains that the notification window shows CSV import log details only after the entire import is completed.

Notification Window Log:

- [2023.01.18:09:51:25] Import Wizard Started
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 10 records processed from total 10 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 4 records processed from total 4 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 6 records processed from total 6 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 14 records processed from total 14 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 16 records processed from total 16 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 8 records processed from total 8 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 1 records processed from total 1 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 32 records processed from total 32 records.
- [2023.01.18:09:51:30] Import Started
- [2023.01.18:09:51:30] Import Complete - 12 records processed from total 12 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 32 records processed from total 32 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 32 records processed from total 32 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 12 records processed from total 12 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 1 records processed from total 1 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 1 records processed from total 1 records.
- [2023.01.18:09:51:31] Import Started
- [2023.01.18:09:51:31] Import Complete - 1 records processed from total 1 records.

See the Notifications Window for the CSV Import log details. The details are only displayed after the entire import is completed.