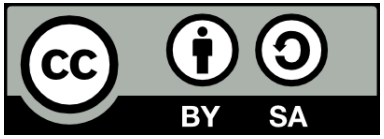
ICTT System Sciences®
Understand your systems.

V1.4.6

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

By: S* Patterns Community

© 2025, System Sciences, LLC.



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

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 1. Systematica Profile 2. Systematica Project Template 3. Systematica Images 4. Systematica Diagram Styles 5. Pattern Export Tables Definition 6. CSV Import Mapping Definition	12-27
3. Creating S* Pattern Classes 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	28-76

Contents

Section	Pages
4. Creating S* Pattern Relationships	77-94
5. Creating S* Pattern Configuration Rules	95-110
6. Exporting and Importing Pattern and Model Data During S* Pattern Configuration	111-115
7. Pre-defined Cameo Model CSV Import Mapping Definitions	116-147

Change History

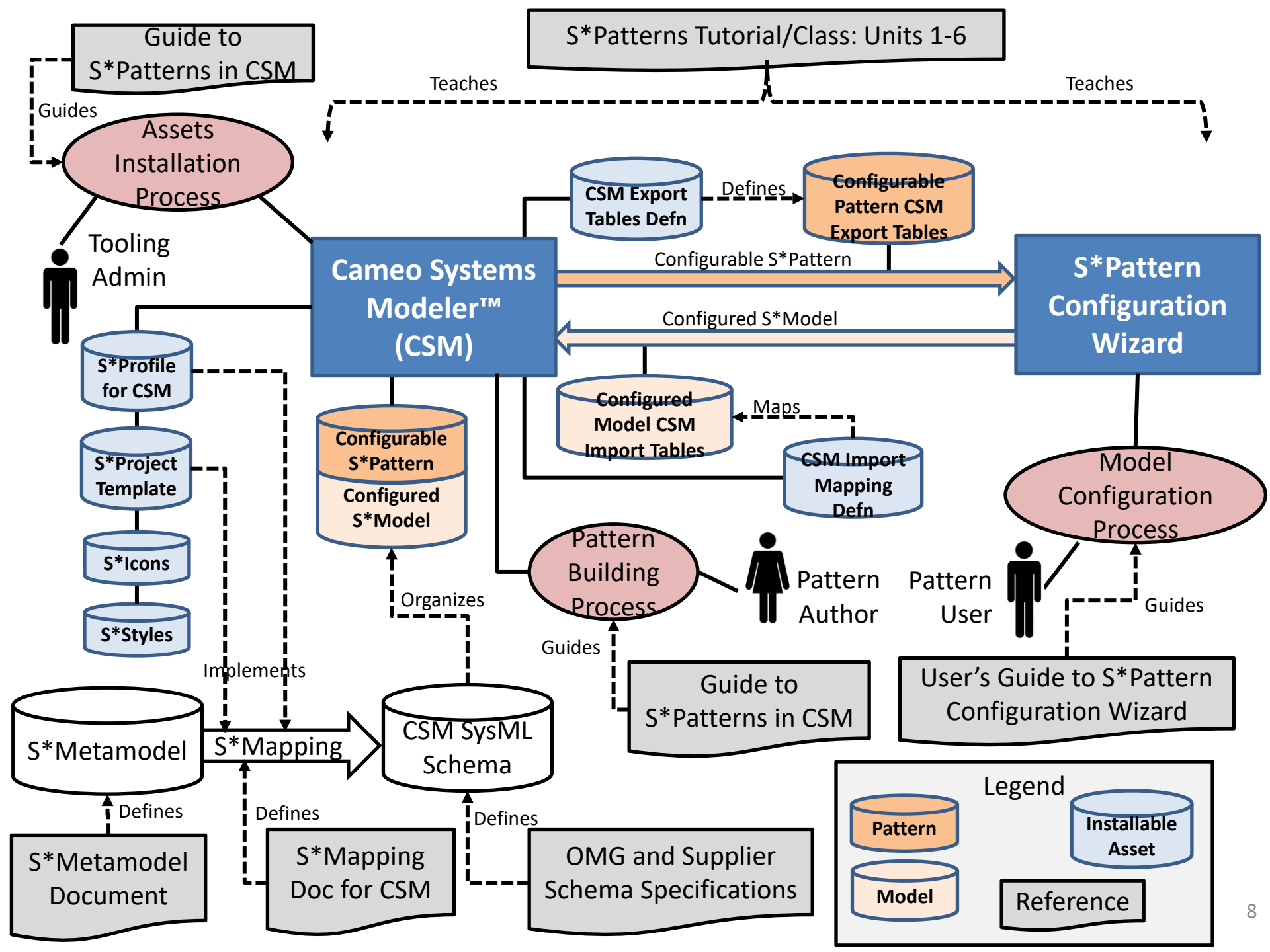
Version #	Date	Who	Description
1.3.28	10.19.2023	Stephen Lewis	Inserted updated diagram on slide 7
1.3.29	10.20.2023	Stephen Lewis	Added Change History Slide, Slides 32, 98-99
1.3.30	10.23.2023	Stephen Lewis	Table of Contents and Section descriptions, Slides 35, 71
1.3.31	10.24.2023	Stephen Lewis	Matrix 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
1.3.40	11.7-8.2023	Stephen Lewis	Updated screenshots for tables with new column headings
1.3.41	11.9.2023	Stephen Lewis	Figure Page Number Reference

Change History

Version #	Date	Who	Description
1.3.42	3.6.2024	Stephen Lewis	CSV Import Literal String Replacement Slide 26
1.3.43	3.20.2024	Stephen Lewis	Export Tables report details, companion asset versions
1.3.44	4.17.2024	Stephen Lewis	CSV Import Path Variable, Slides 25-26
1.4.1	2.26.2025	Stephen Lewis	Updated CSV Import Path Details, Slides 25-26. Added CSV Import Mapping Definitions for Cameo 2024xR2, Slides 115-145
1.4.2	2.27.2025	Stephen Lewis	Updates to details on Slides 25-26, added new slide 27
1.4.3	2.28.2025	Stephen Lewis	Import Mapping Definition slide layouts 116-146
1.4.4	3.3.2025	Stephen Lewis	Specific section for Import Mapping Defn's, textual edits
1.4.5	3.5.2025	Stephen Lewis	Slides 14-7, 19, 20, 22-4, 26, title
1.4.6	3.6-7.2025	Stephen Lewis	Companion Asset Installation Section

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 Systematica (S*) Profile and Systematica (S*) Project Template, Images and Styles, 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.


Guide to S*Patterns in CSM
S*Patterns Tutorial/Class: Units 1-6
Guides
Teaches
Teaches
Assets Installation Process
Tooling Admin
CSM Export Tables Defn
Defines
Configurable Pattern CSM Export Tables
Configurable S*Pattern
Cameo Systems Modeler™ (CSM)
Configured S*Model
S*Pattern Configuration Wizard
S*Profile for CSM
S*Project Template
S*Icons
S*Styles
Configurable S*Pattern
Configured S*Model
Configured Model CSM Import Tables
Maps
CSM Import Mapping Defn
Model Configuration Process
Pattern Building Process
Pattern Author
Pattern User
Guides
Organizes
Guides
Implements
Guide to S*Patterns in CSM
User's Guide to S*Pattern Configuration Wizard
S*Metamodel
S*Mapping
CSM SysML Schema
Defines
Defines
Defines
S*Metamodel Document
S*Mapping Doc for CSM
OMG and Supplier Schema Specifications
Legend
Pattern
Installable Asset
Model
Reference

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 w/ periodic updates for S* MTM or CSM releases.

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, 06/11/2024.
3. “S*Metamodel Mapping for MagicDraw/Cameo Systems Modeler Version 19”, Version 1.11.12, 2/28/2024.
4. “Guide to the S*Pattern Configuration Wizard”, V1.5.13, 8/8/2024.

*Refer to Figure 1, Page 8

Companion Assets

1. S* Project Template Version Feb2025, *Systematica.mdzip*.
2. S* Profile Version Feb2025, *Systematica Profile.mdzip*.
3. S* Images, *Systematica.zip*.
4. S* Diagram Styles, *Systematica.stl*.
5. Pattern Export Tables Definition
6. CSV Import Mapping Definition
7. Example International Power Converter S* Pattern or client-specific S* Pattern

Refer to Figure 1, Page 8

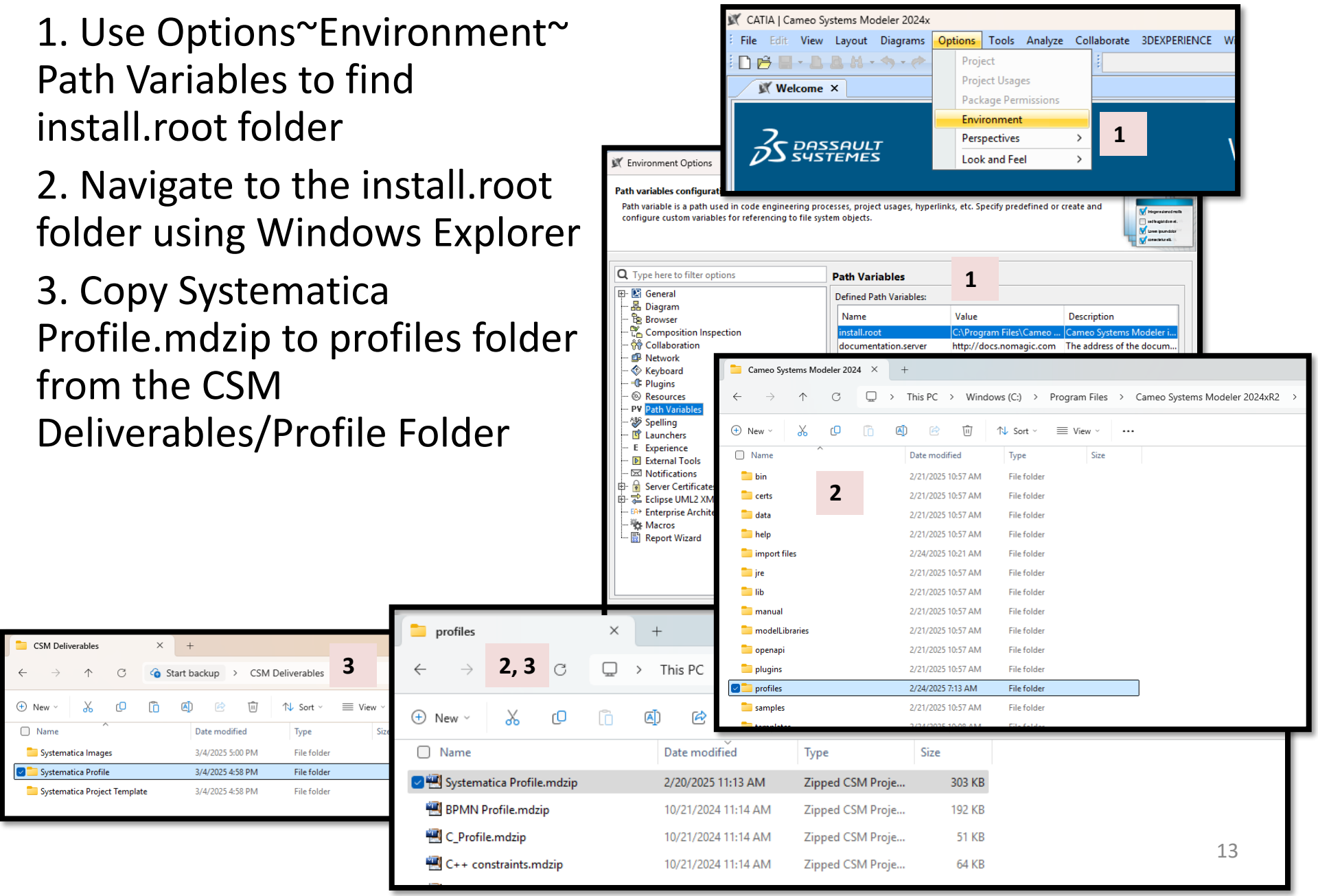
2. Installation of Companion Assets

- This section of the guide details the steps for installation of the S* Profile and S* Project Template.
- To prepare CSM for use with this guidance, load the CSM tool with the above listed Companion Assets using the steps shown in the following slides.
- For Slides 13-15, open the Cameo Systems Modeler application, but do not open a new or existing project. Then use the menu options and dialogs described on those slides.
- For Slides 16-28, in addition to opening the Cameo Systems Modeler application, open the example S* pattern, client-specific S* pattern, or a new Systematica Project. Then use the menu options and dialogs described on those slides.

- *This section assumes Cameo Systems Modeler has already been installed.*

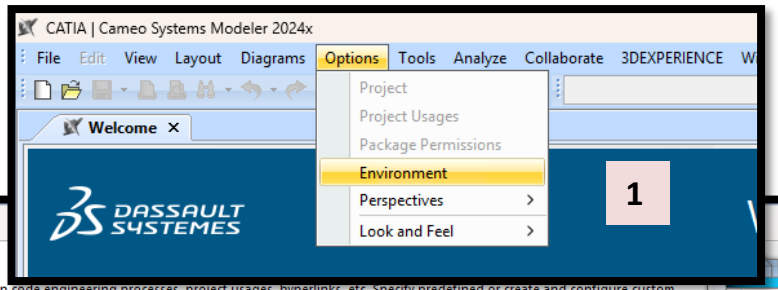
Installing Systematica Profile

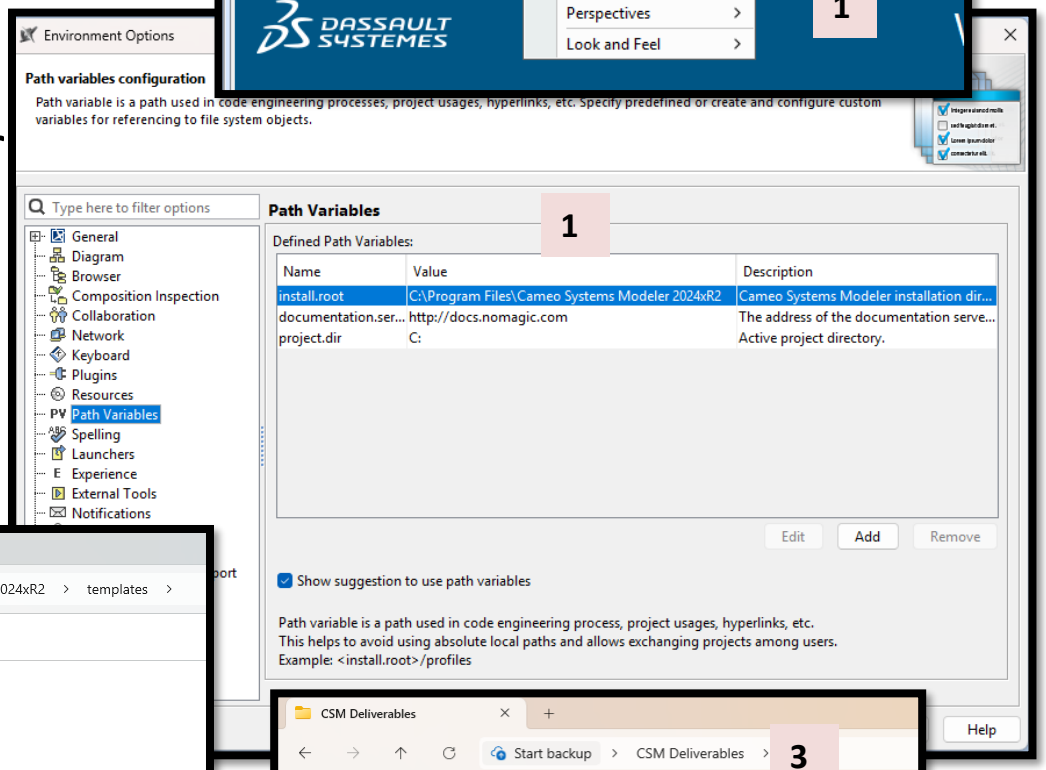
1. Use Options~Environment~ Path Variables to find install.root folder
2. Navigate to the install.root folder using Windows Explorer
3. Copy Systematica Profile.mdzip to profiles folder from the CSM Deliverables/Profile Folder

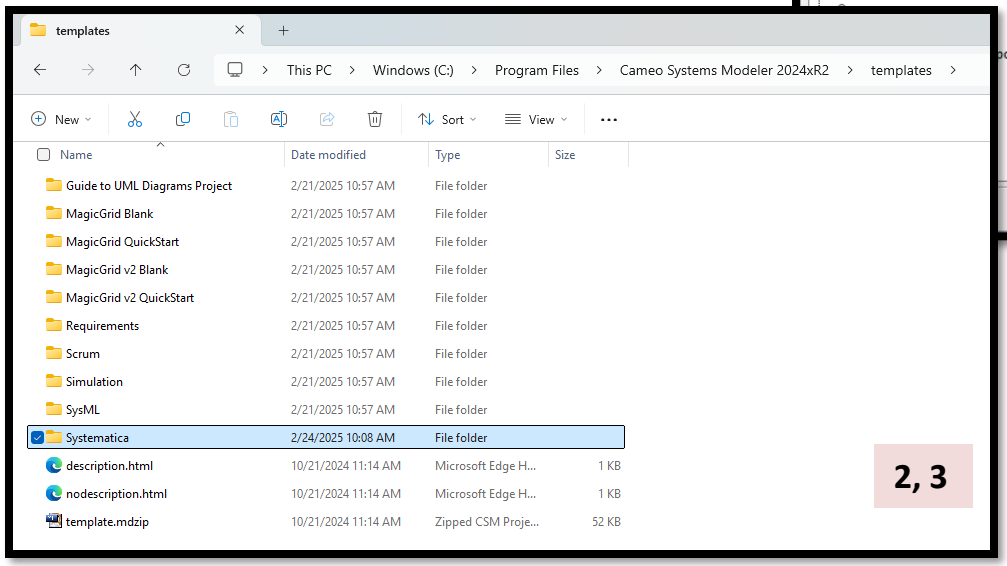


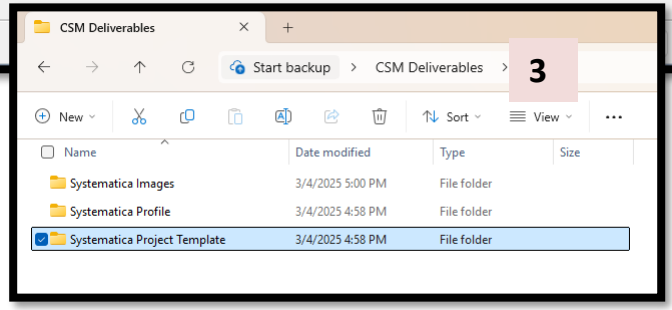
Installing Systematica Project Template

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 CSM Deliverables/Systematica Project Template Folder



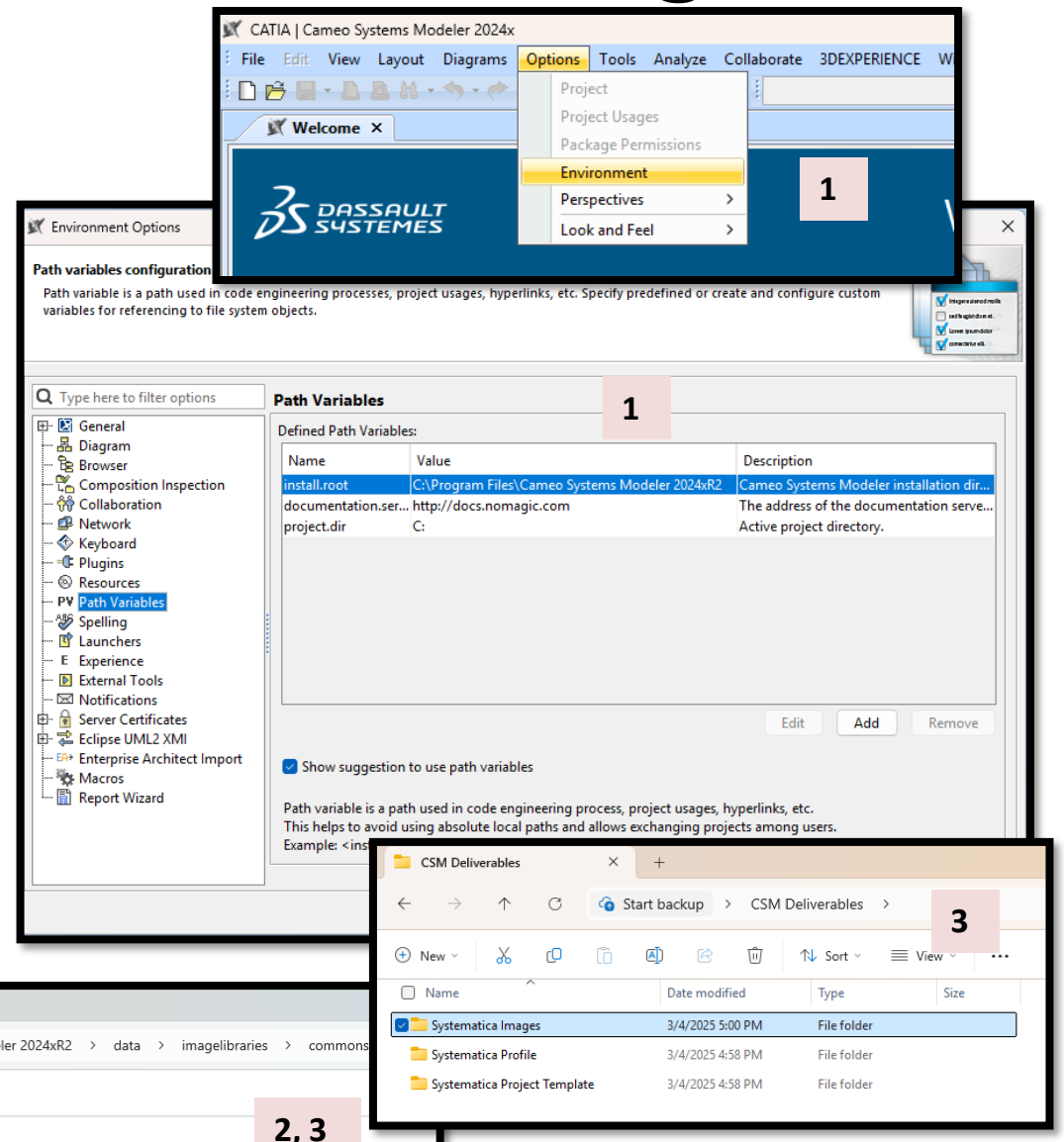






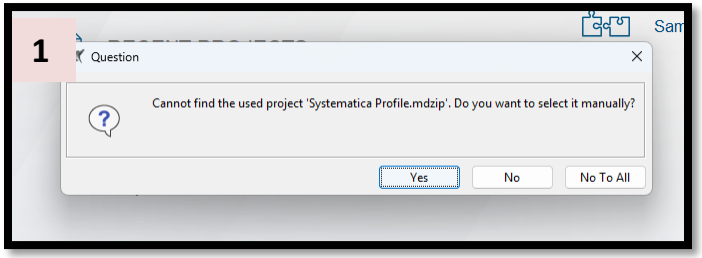
Installing Systematica Images

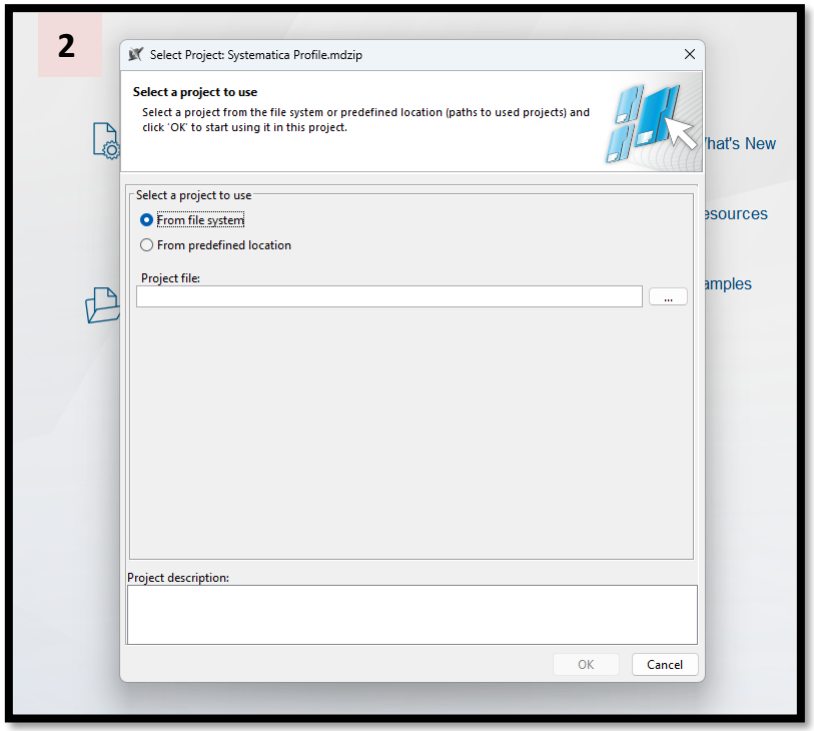
1. Use Options~Environment~ Path Variables to find install.root folder
2. Navigate to the install.root folder using Windows Explorer
3. Copy Systematica zip file to data\imagelibraries\common symbols folder from the CSM Deliverables/Systematica Images Folder

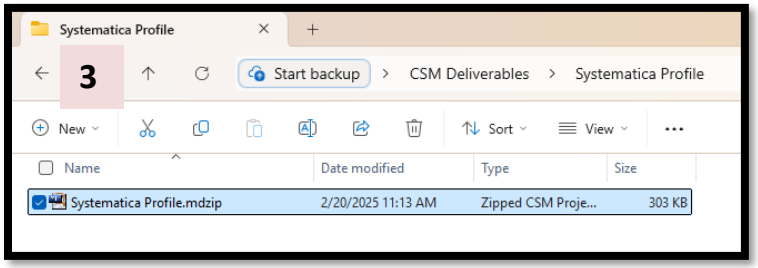


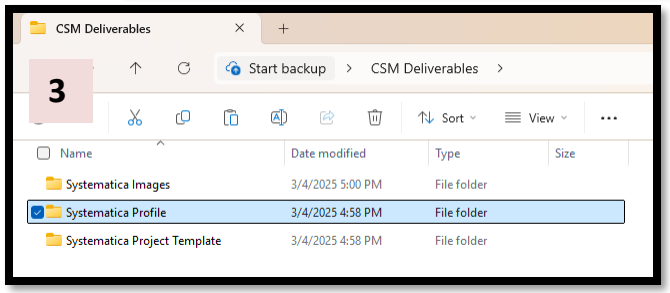
Installing Systematica Profile

1. If you receive the question dialog shown, select “Yes”.
2. Choose the “From file system” option
3. Navigate to the Systematica Profile.mdzip file in the CSM Deliverables/Systematica Profile folder.





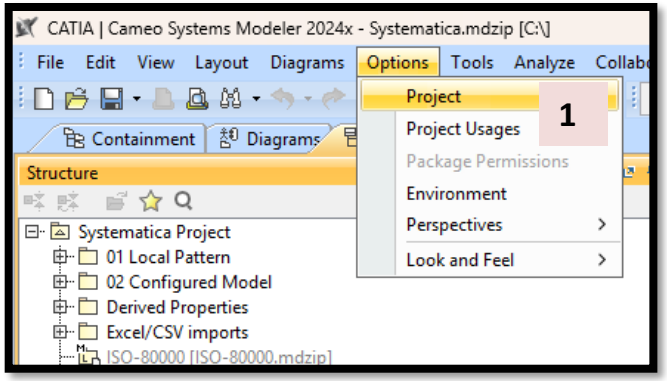




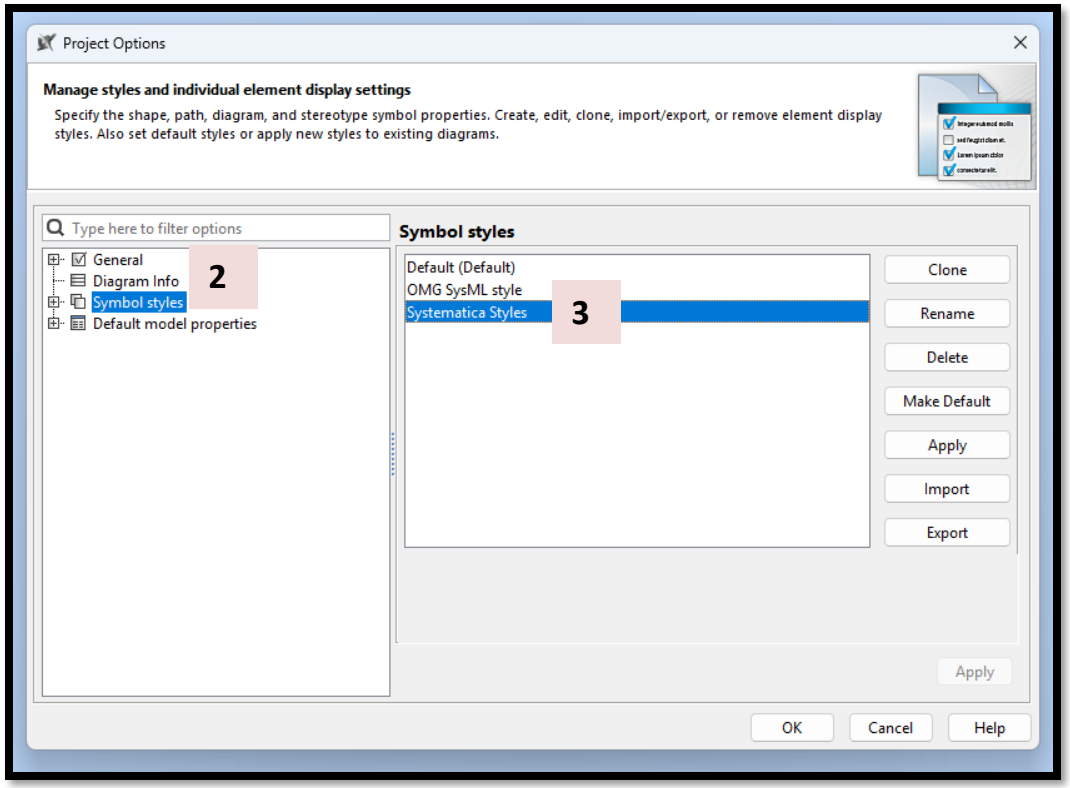
- Alternatively, you can copy the profile to a different accessible location.

Installing Systematica Diagram Styles

- The S* Pattern should already have the Systematica Diagram Styles installed. One can confirm this and apply the styles, if necessary, as described below.



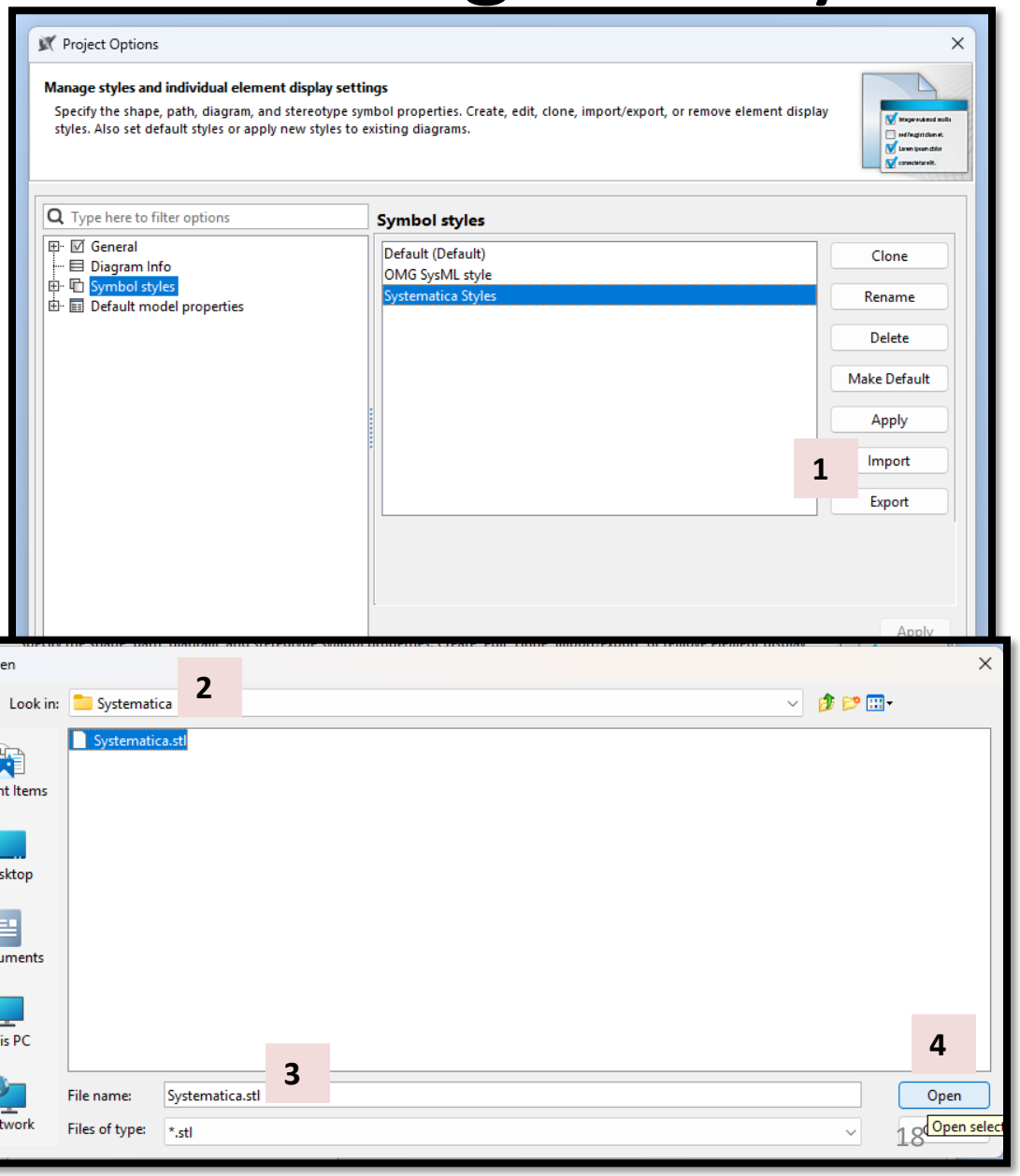
1. Select the menu option ~Options~Project
2. Click on ‘Symbol styles’ in tree in left pane.
3. Confirm the “Systematica Styles” are installed.



- If not shown, use the next slide to install the styles.

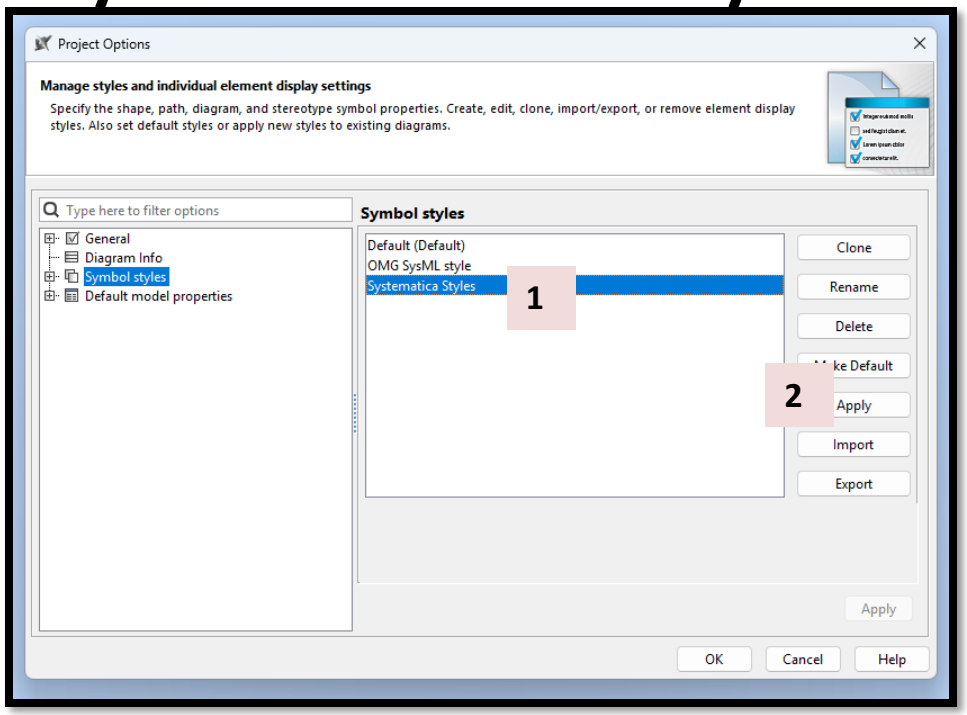
Importing Systematica Diagram Styles

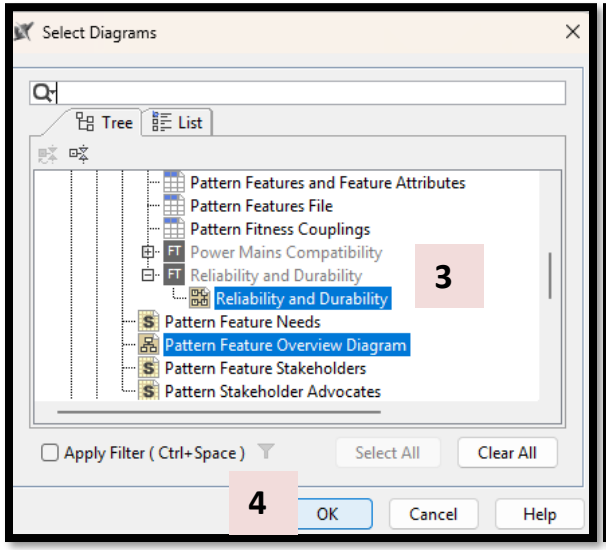
1. Click the ‘Import’ button
2. Navigate to the CSM Deliverables/Systematica Project Template/Systematica Folder
3. Select the Systematica.stl file
4. Click the ‘Open’ button

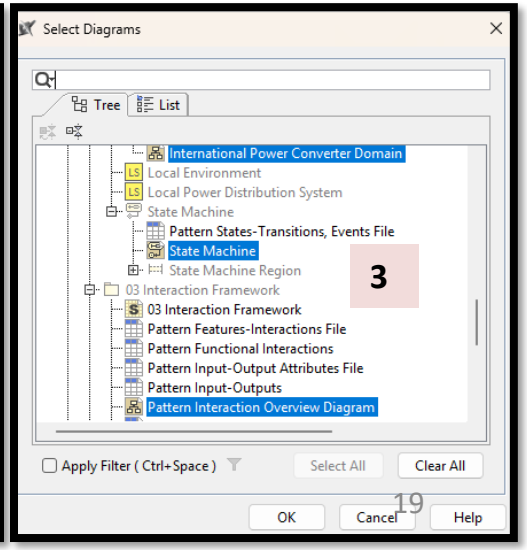


Applying Diagram Systematica Styles

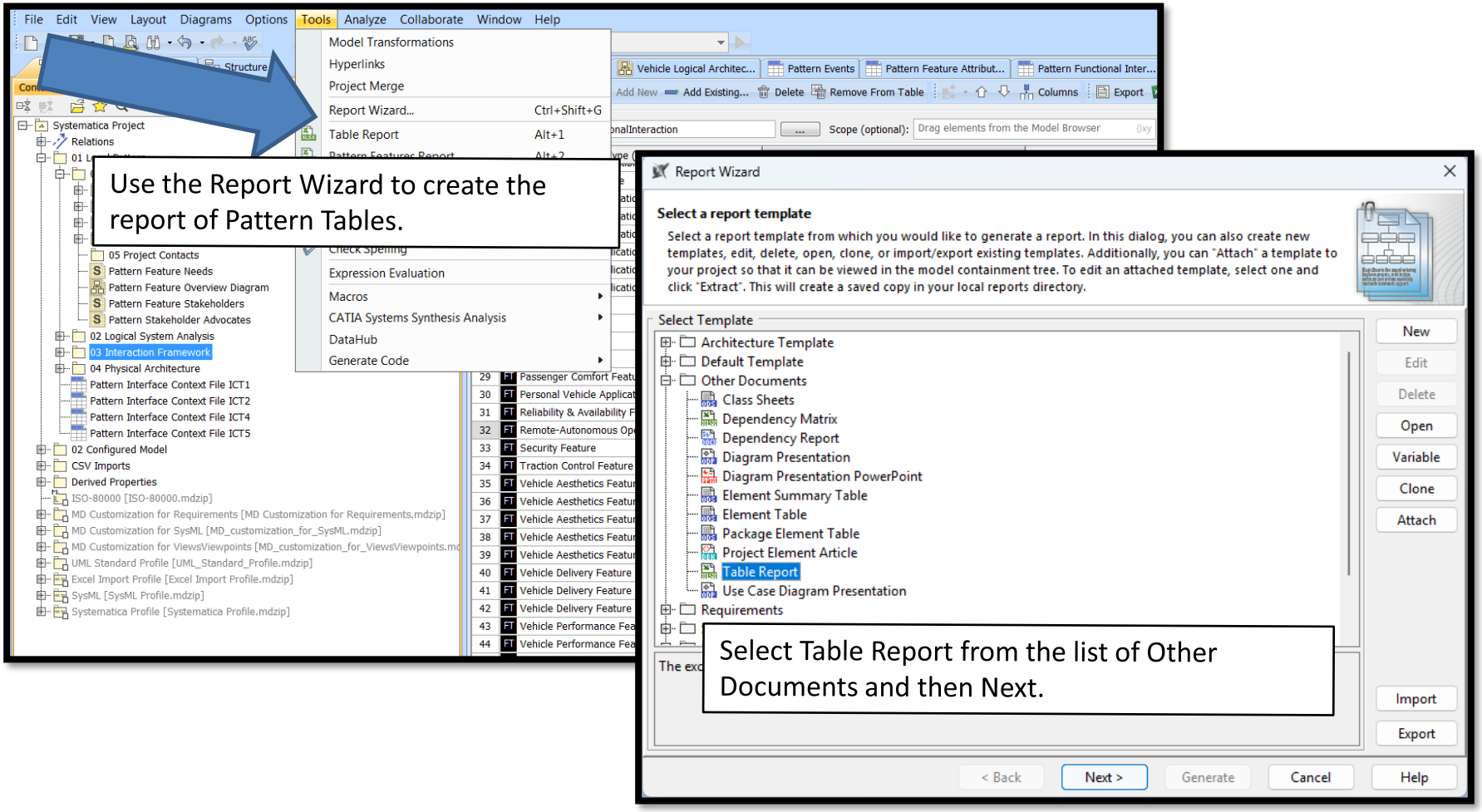
1. Select the ‘Systematica Styles’ node in tree in the left pane.
2. Click the ‘Apply’ button
3. Select diagrams, if present. Some have general names like Pattern Feature Overview Diagram, Interaction Overview Diagram, and State Machine, while others have specific names like Reliability and Availability (Parametric Attribute Coupling Diagram) and International Power Converter Domain (Domain Diagram).
4. Click the ‘OK’ button



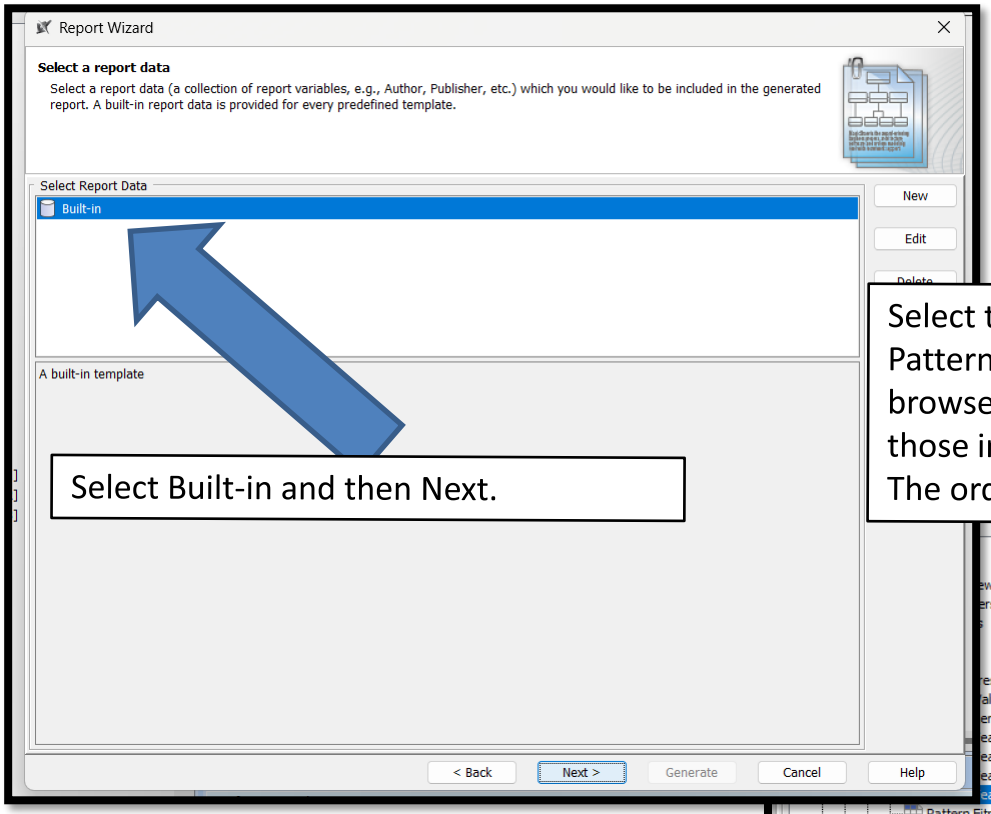




Defining Pattern Export Tables Using the Report Wizard

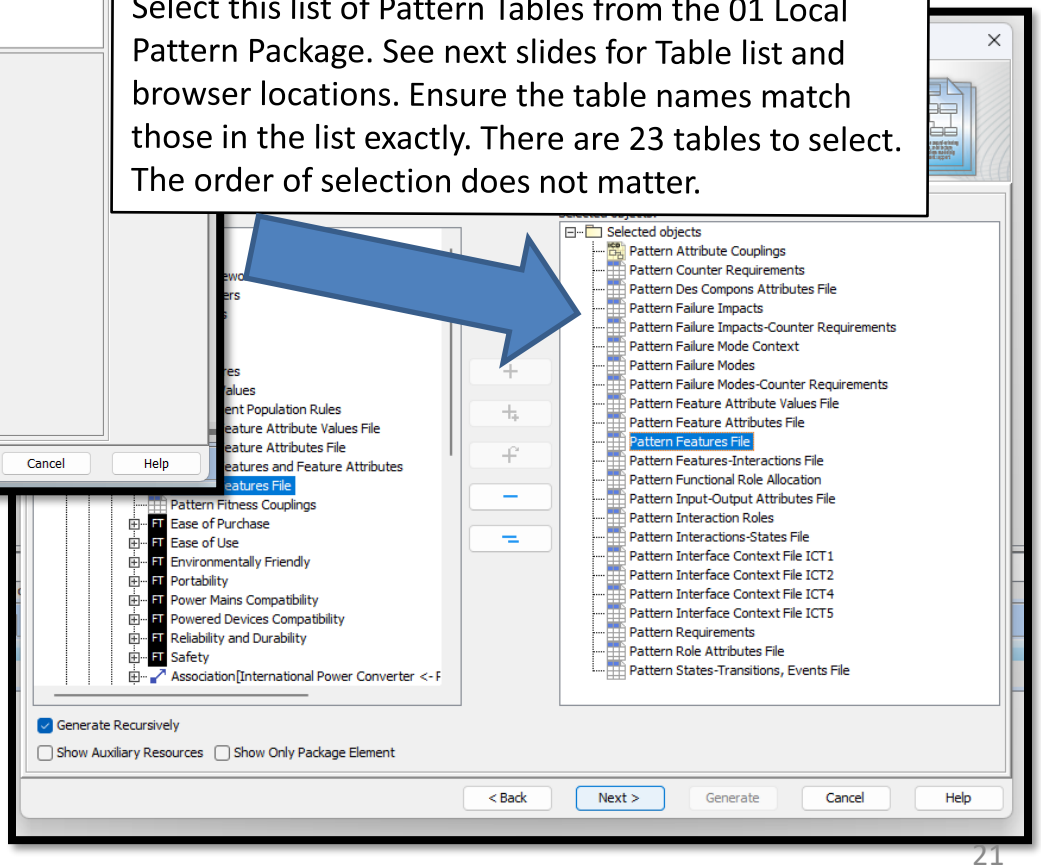


Defining Pattern Export Tables Using the Report Wizard



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.

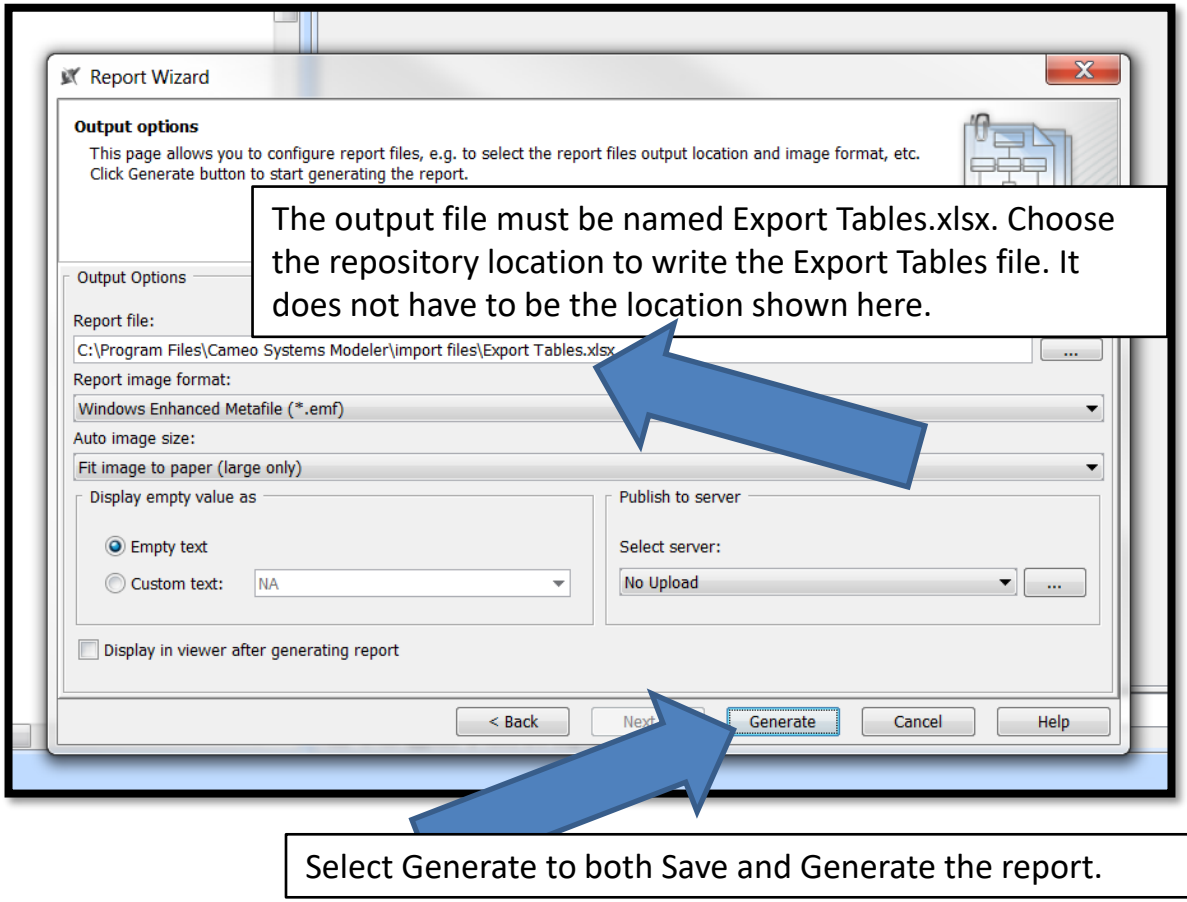
Defining Pattern Export Tables Using the Report Wizard

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

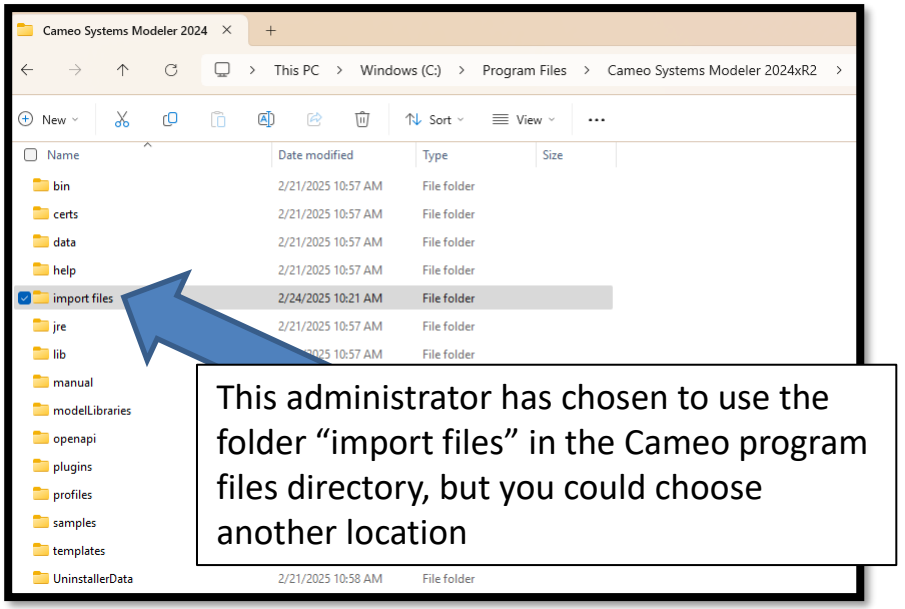
Table Name	Table Browser Location
Pattern Functional Role Allocation	01 Local Pattern>04 Physical Architecture
Pattern Des Compons Attributes File	01 Local Pattern>04 Physical Architecture
Pattern Failure Impacts	01 Local Pattern>05 Risk Framework
Pattern Failure Modes	01 Local Pattern>05 Risk Framework
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



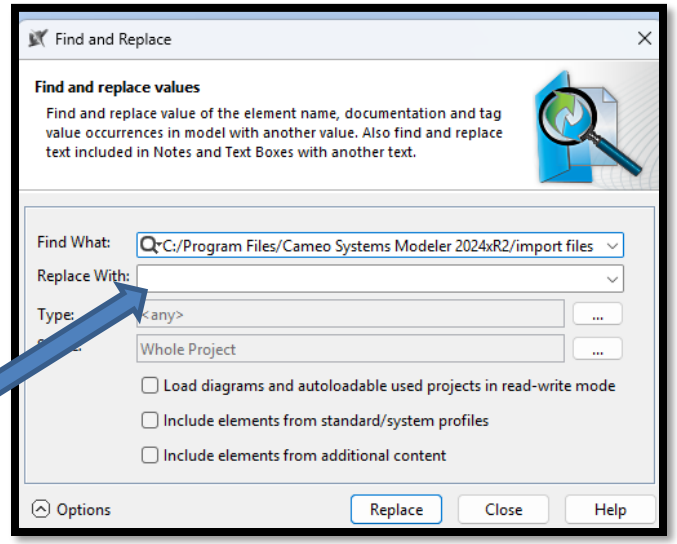
CSV Import Mapping Definition for Configured Model Data

- Refer to Figure 1. The following explains how to set up a folder location, in the file system referenced by CSM, into which configured model data will be deposited by the Configuration Wizard for import into CSM. (Guidance for the Configuration Wizard end of that setup is found in the Configuration Wizard Guide.) Guidance for the CSM import end of that setup follows below.
- 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.



CSV Import Mapping Definition for Configured Model Data

- There are approximately 60 pre-defined import map definitions provided in the Companion Assets, describing the detailed mapping of configured models into CSM. Each of those map definitions includes a file reference (a file name with a directory path to it in the folder structure) to a specific importable configured model file that will be generated by the Configuration Wizard, for import into CSM. All of those pre-defined file references should point to files that will be in the same folder, which you can choose to change for your file folder structure, as described on the previous page.
- The CSV Import file path can be edited in Cameo to a different path, as needed.
- Select **Find** and **Replace** from the **Edit** Menu
- The preset path is C:/Program Files/Cameo Systems Modeler 2024xR2/import files
- Enter the above path in “Find What:”
- Enter the alternate path in “Replace With:”



CSV Import Mapping Definition for Configured Model Data

Display shown of import definition file paths, found by a FIND operation by the CSM editor, looking for “C:/Program Files/Cameo Systems Modeler 2024xR2/import files”, prior to replacing that path with an alternate path. Until CSM import definitions support use of the CSM configurable Environment Path Variable, the alternate path for all those import files can be inserted in one step by using the CSM text editor to change all the import paths in a single operation. A FIND operation will show all the current appearances of the initial default path as shown in the adjacent screen shot. A REPLACE operation can be used to replace that prefix in each different path name with an alternate path name prefix of your choosing as shown on the previous slide.

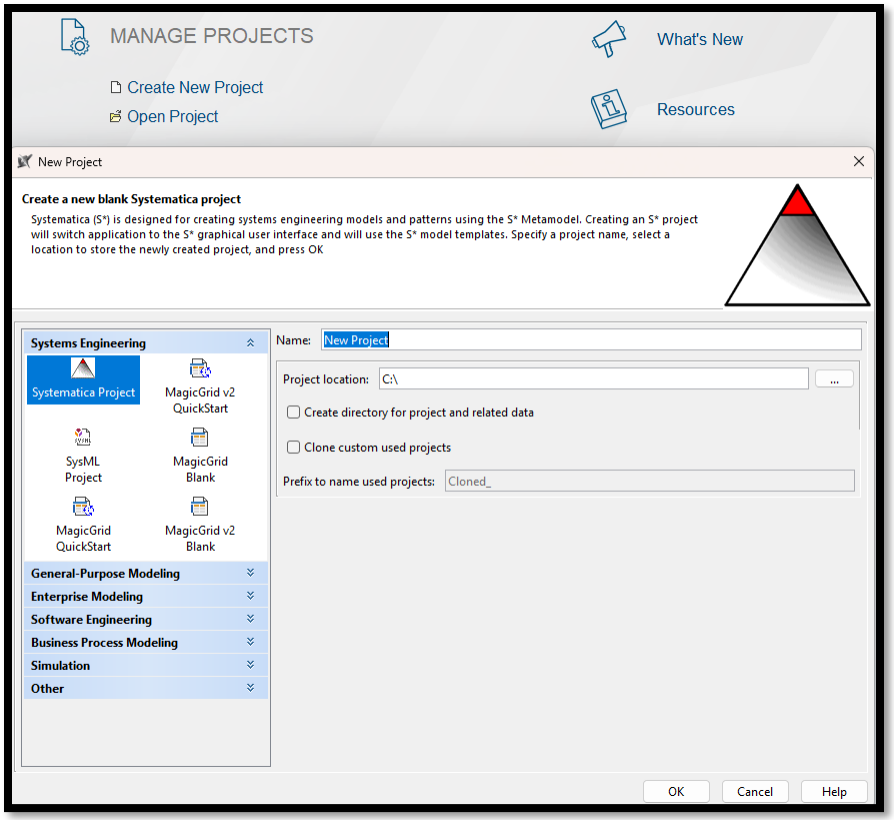


3. Creating S* Pattern Classes

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

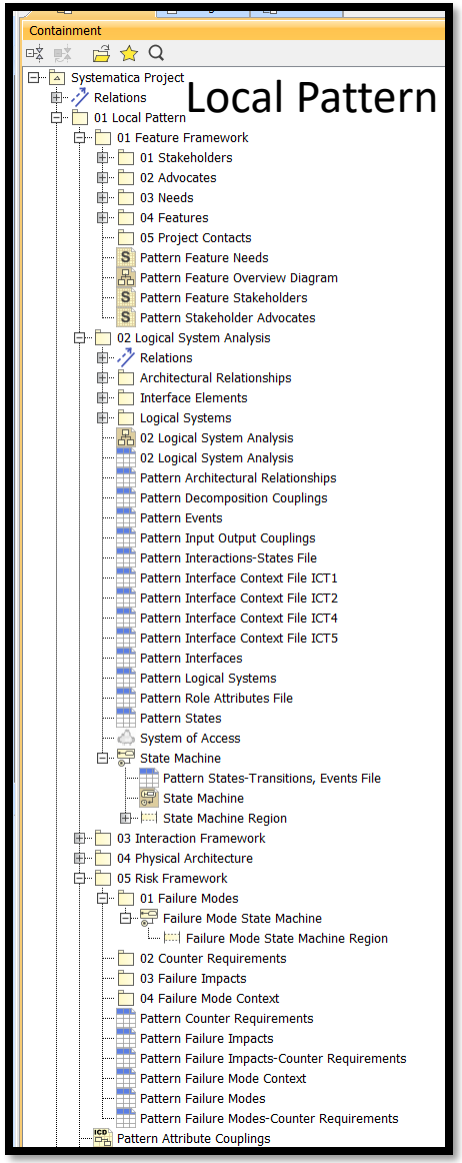
Creating a New S* Pattern Project

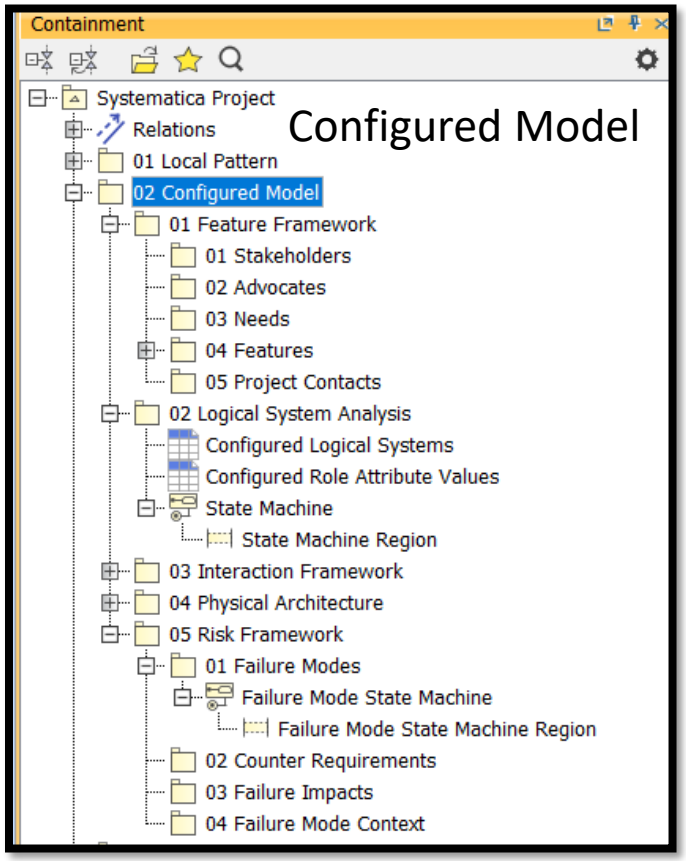
- Start CSM and Select Create New Project.
- Select the Systematica Project.
- Name the Project and Select a File Location.



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.



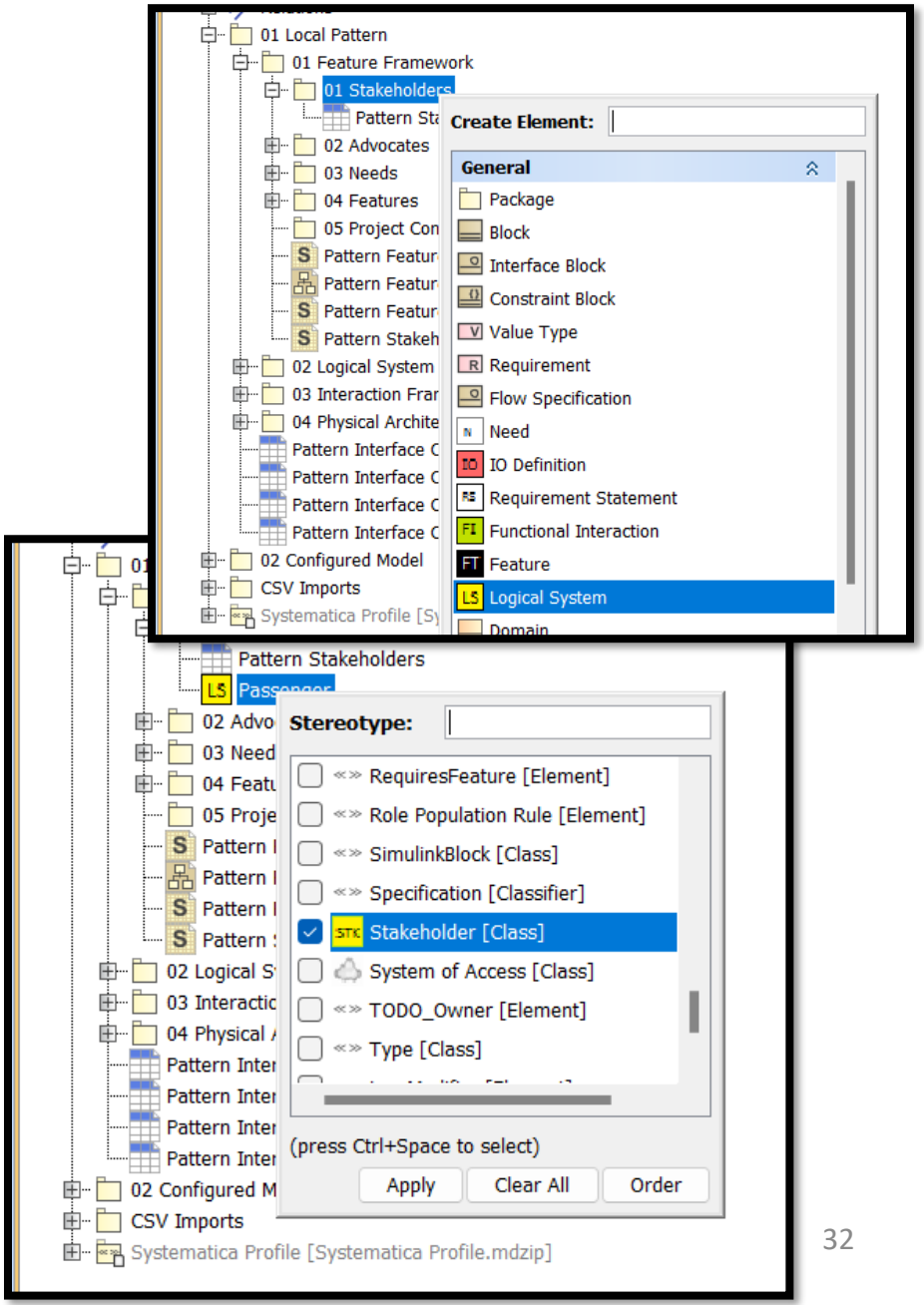


Pattern data entry methods and details

- 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.

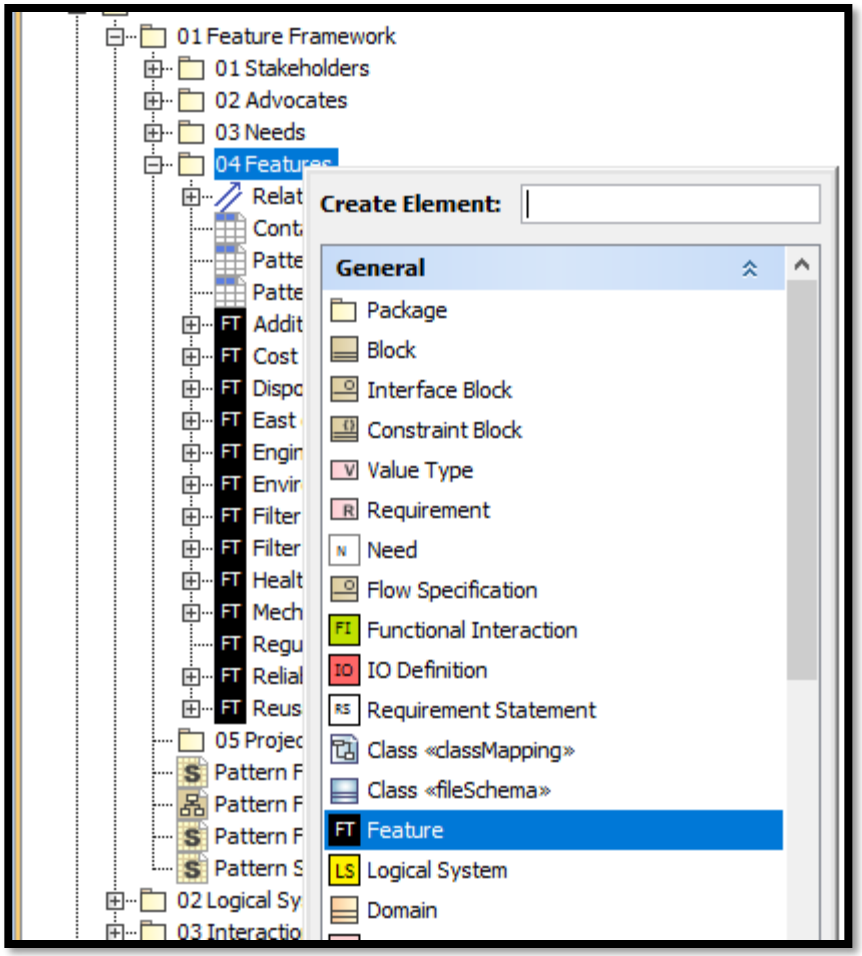
Feature Framework: Creating Stakeholders

- 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.



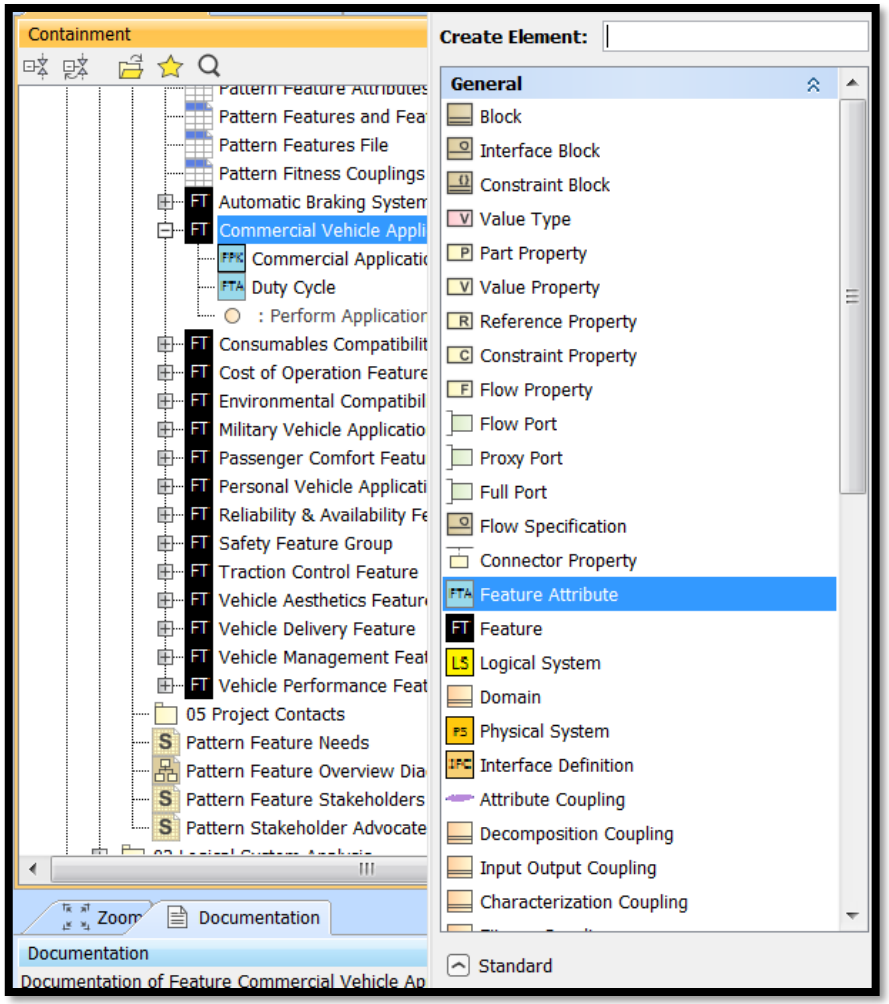
Feature Framework: Creating Features

- 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



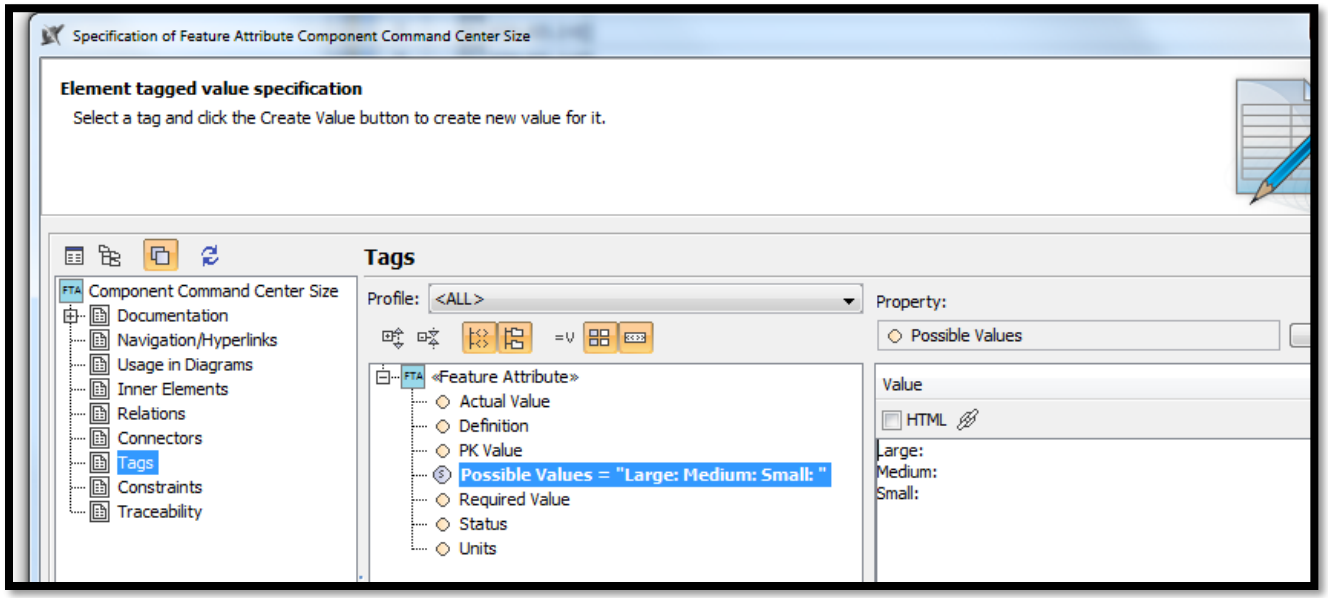
Feature Framework: Creating Feature Attributes

- 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.



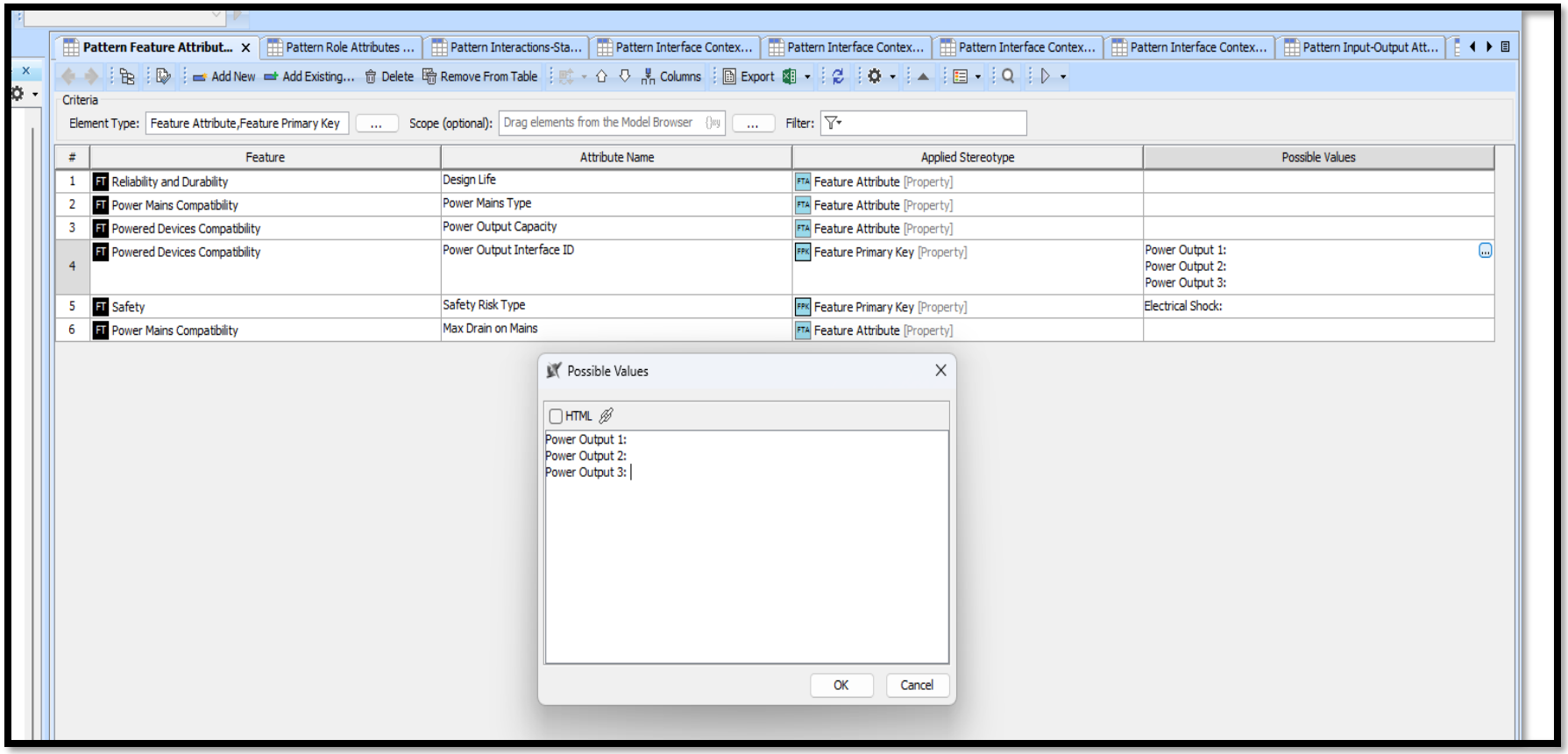
Feature Framework: Feature Attribute Possible Values

- 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



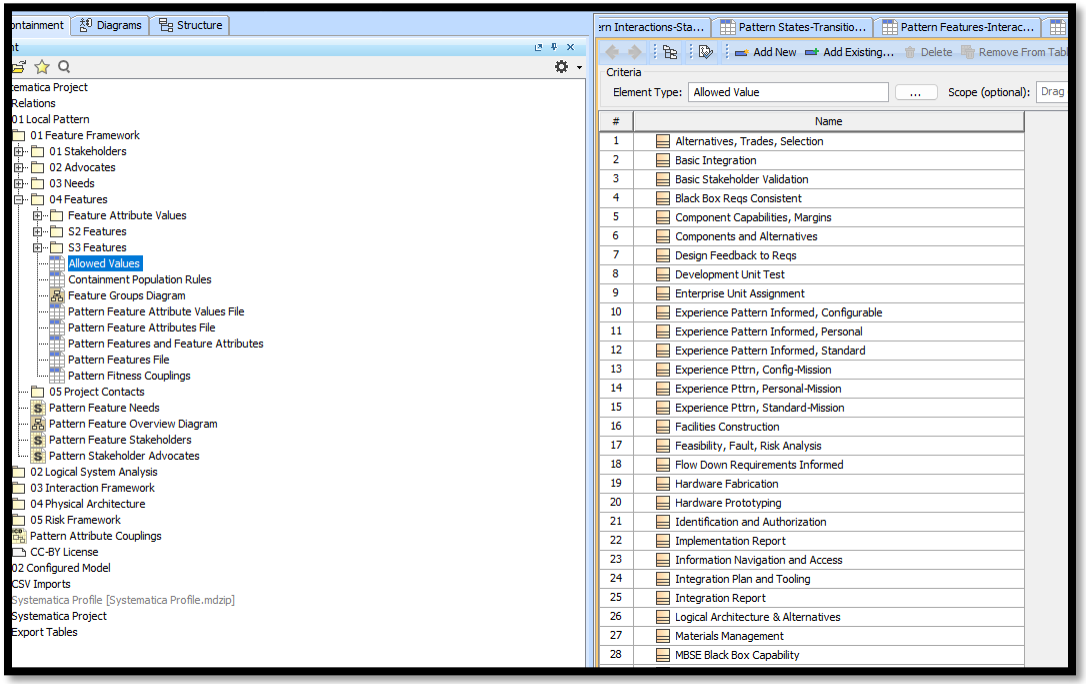
Feature Framework: Feature Attribute Possible Values

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



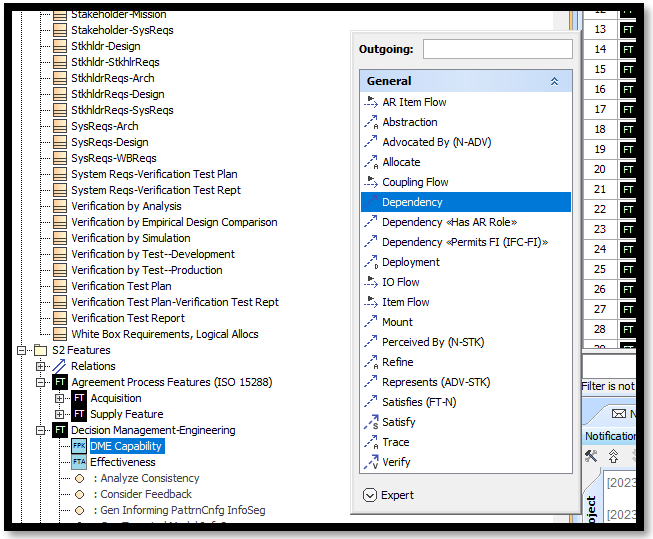
Feature Framework: Feature Attribute Allowed Values

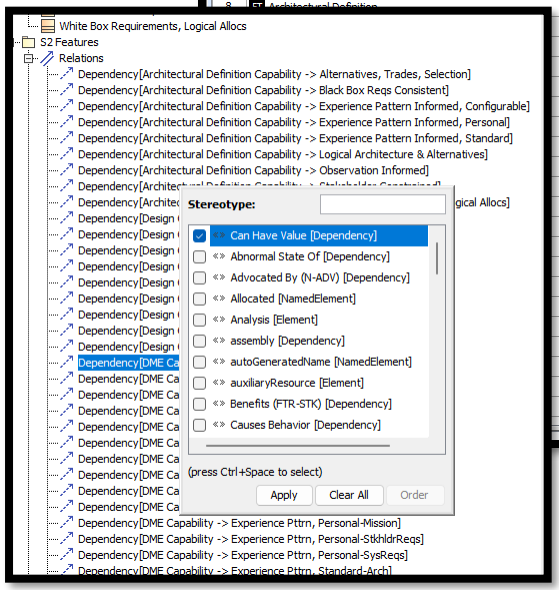
- An alternative method for creating Feature Attributes and connecting them to Features for configuration includes the use of allowed values.
- Enter the complete list of Feature Attribute Values in the Allowed Values Table.



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.

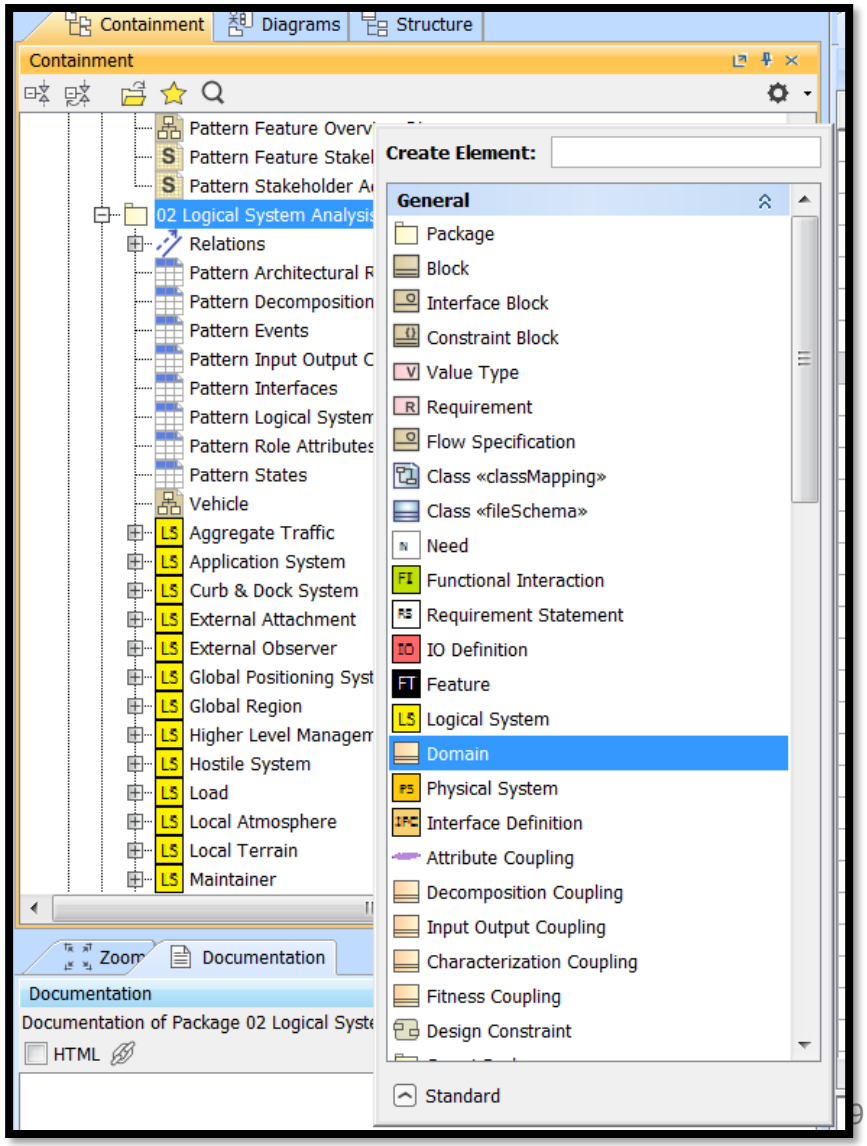




#	Feature	Feature Primary Key	Allowed Value
1	Architectural Definition	Architectural Definition Capability	Alternatives, Trades, Selection
2	Architectural Definition	Architectural Definition Capability	Black Box Reqs Consistent
3	Architectural Definition	Architectural Definition Capability	Experience 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
9	Architectural Definition	Architectural Definition Capability	White Box Requirements, Logical Allocs
10	Mission Analysis Capability	Mission Analysis Capability	Experience Pattern Informed, Configurable
11	Mission Analysis Capability	Mission Analysis Capability	Experience Pattern Informed, Personal
12	Mission Analysis Capability	Mission Analysis Capability	Experience Pattern Informed, Standard
13	Mission Analysis Capability	Mission Analysis Capability	Observation Informed
14	Mission Analysis Capability	Mission Analysis Capability	Simulation Informed
15	Mission Analysis Capability	Mission Analysis Capability	Stakeholder Informed
16	DME Capability	DME Capability	Experience Pttrn, Config-Mission
17	DME Capability	DME Capability	Experience Pttrn, Personal-Mission
18	DME Capability	DME Capability	Experience Pttrn, Standard-Mission
19	DME Capability	DME Capability	Observation-Mission
20	DME Capability	DME Capability	Simulation-Mission
21	DME Capability	DME Capability	Stakeholder-Mission
22	DME Capability	DME Capability	ComparDesign-Design
23	DME Capability	DME Capability	Constraints-Arch
24	DME Capability	DME Capability	DesAnalysis-SysReqs
25	DME Capability	DME Capability	Experience Pttrn, Config-Arch
26	DME Capability	DME Capability	Experience Pttrn, Config-Design
27	DME Capability	DME Capability	Experience Pttrn, Config-StkhldrReqs
28	DME Capability	DME Capability	Experience Pttrn, Config-SysReqs

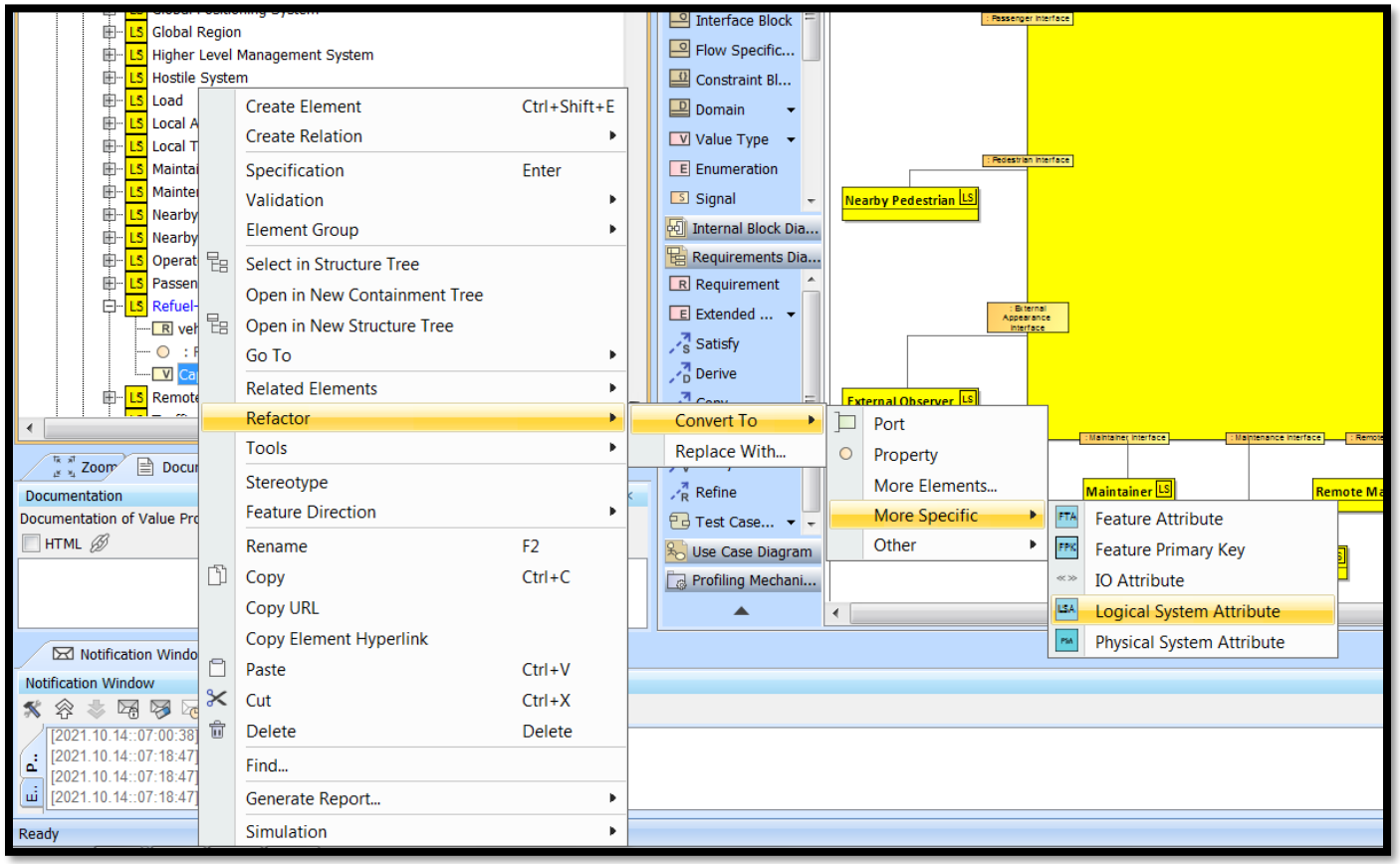
Domain Analysis:
Creating Domains and Logical Systems

- 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



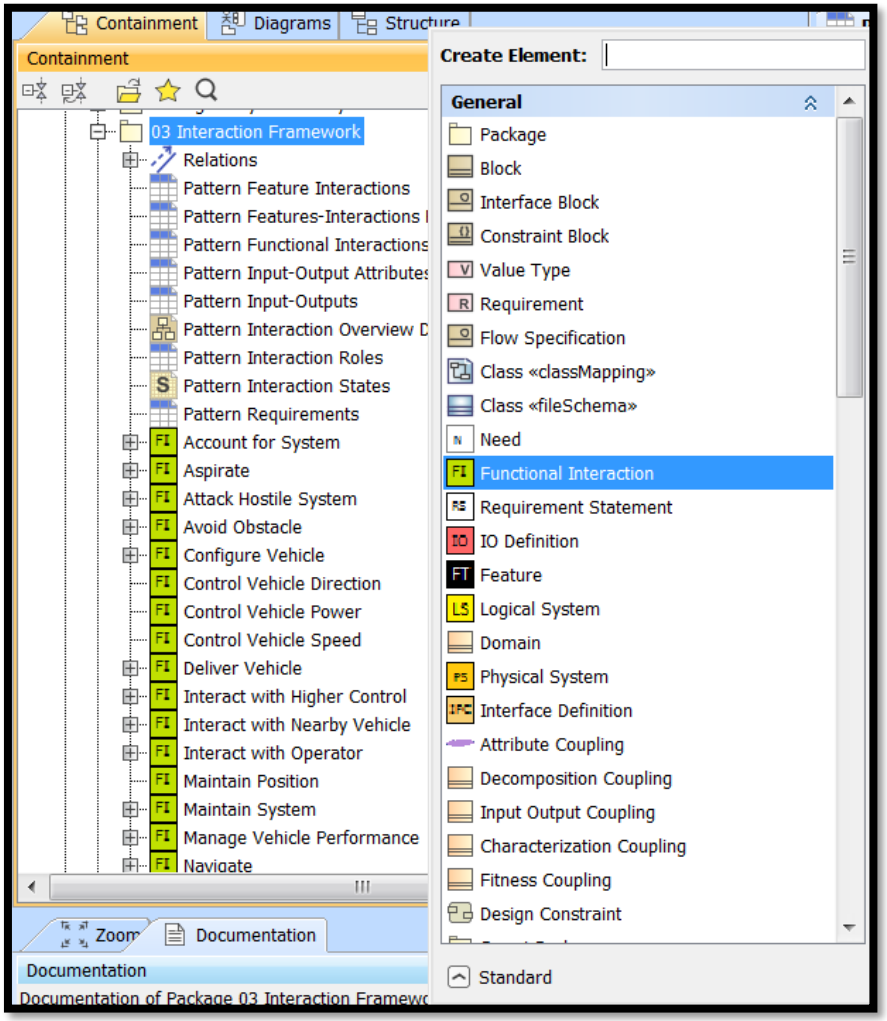
Domain Analysis: Creating Logical System Attributes

- 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



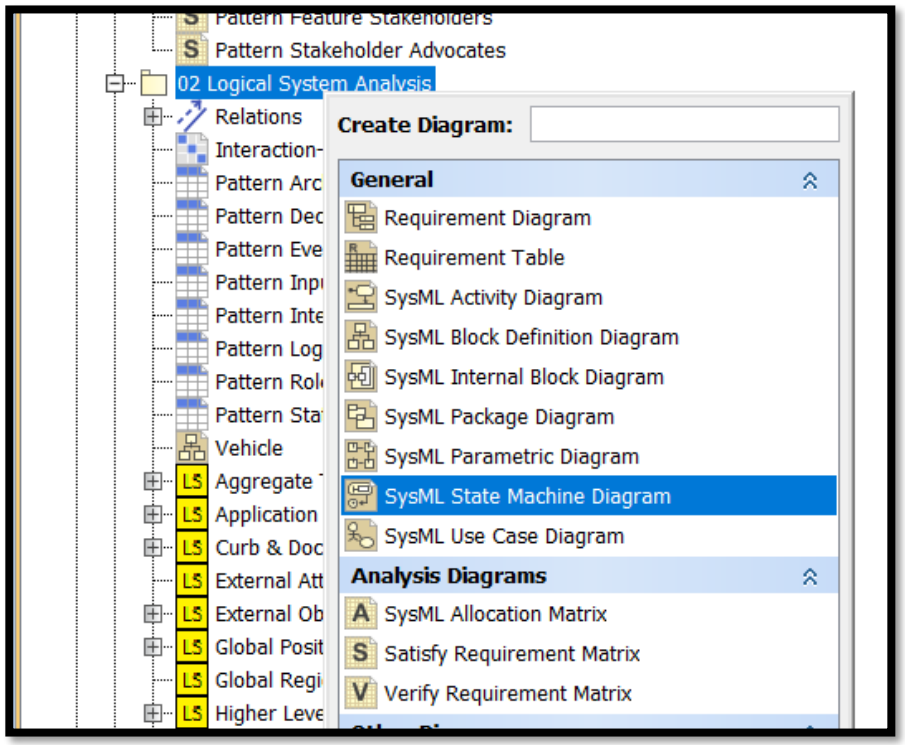
Interaction Framework: Creating Functional Interactions

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



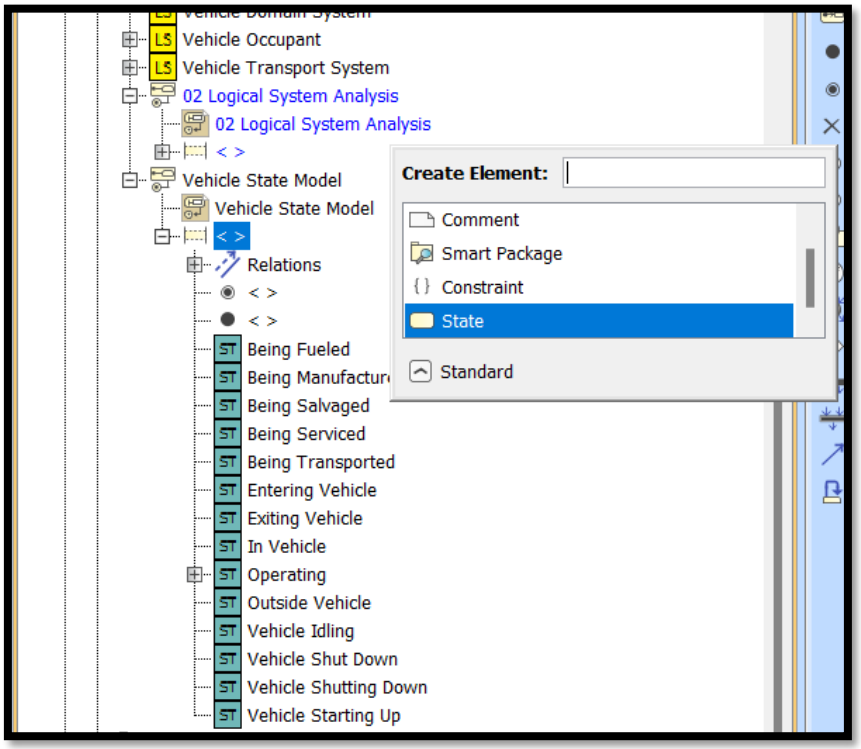
State Machine: Creating State Machine and States

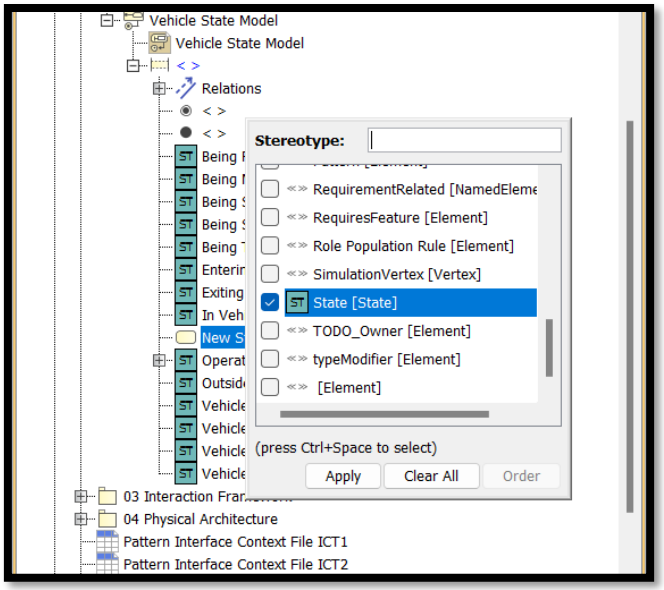
- Pattern State Diagram and States are created in package 01 Local Pattern::02 Logical System Analysis
- Use ~Create Diagram and choose SysML State Machine Diagram



State Machine: Creating State Machine and States

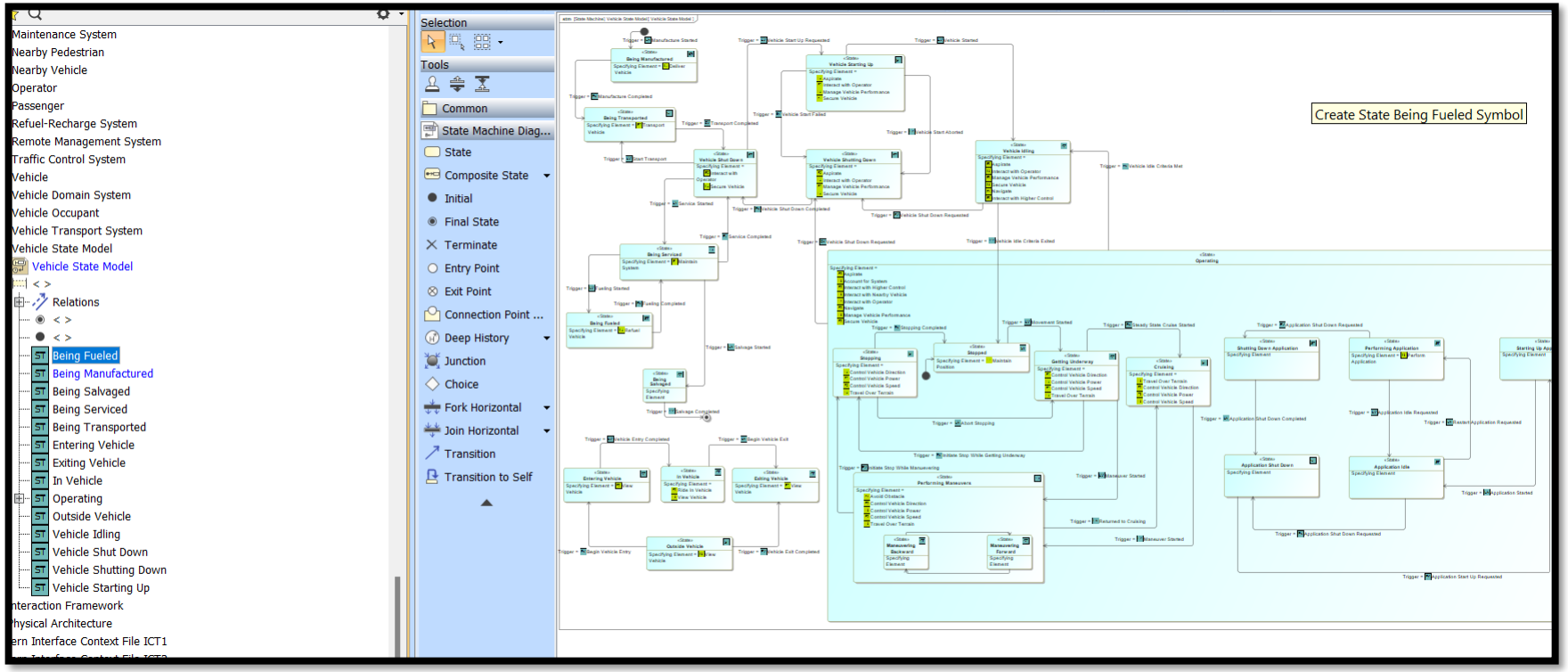
- Right-click the automatically created Region under the State Model and use ~Create Element and choose State. Make sure the menu is expanded to Expert.
- Right-click the newly created State, select Stereotype, and change it to State as shown below.





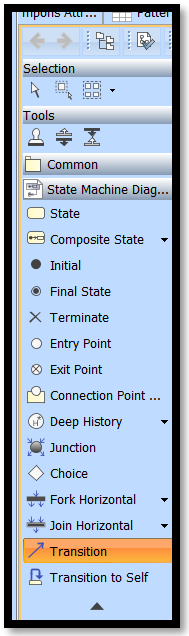
State Machine: Creating the Diagram

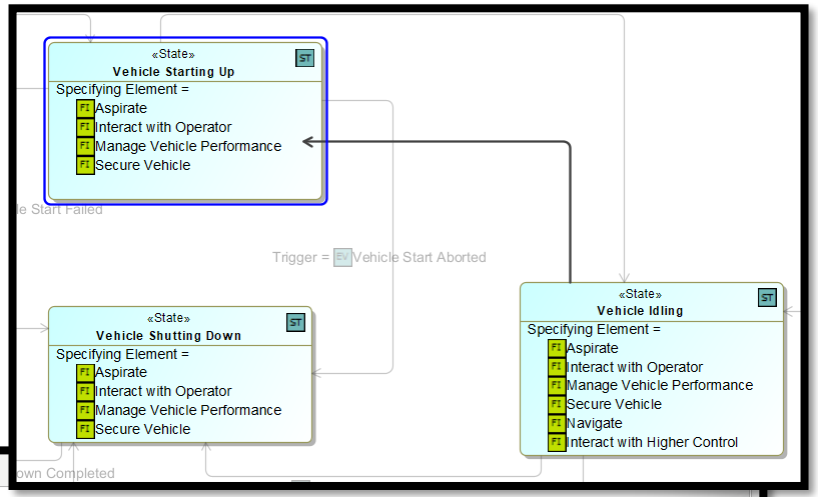
- Open the State Machine Diagram created in the earlier step.
- Populate the State Machine Diagram with States by dragging them from the containment browser onto the diagram.

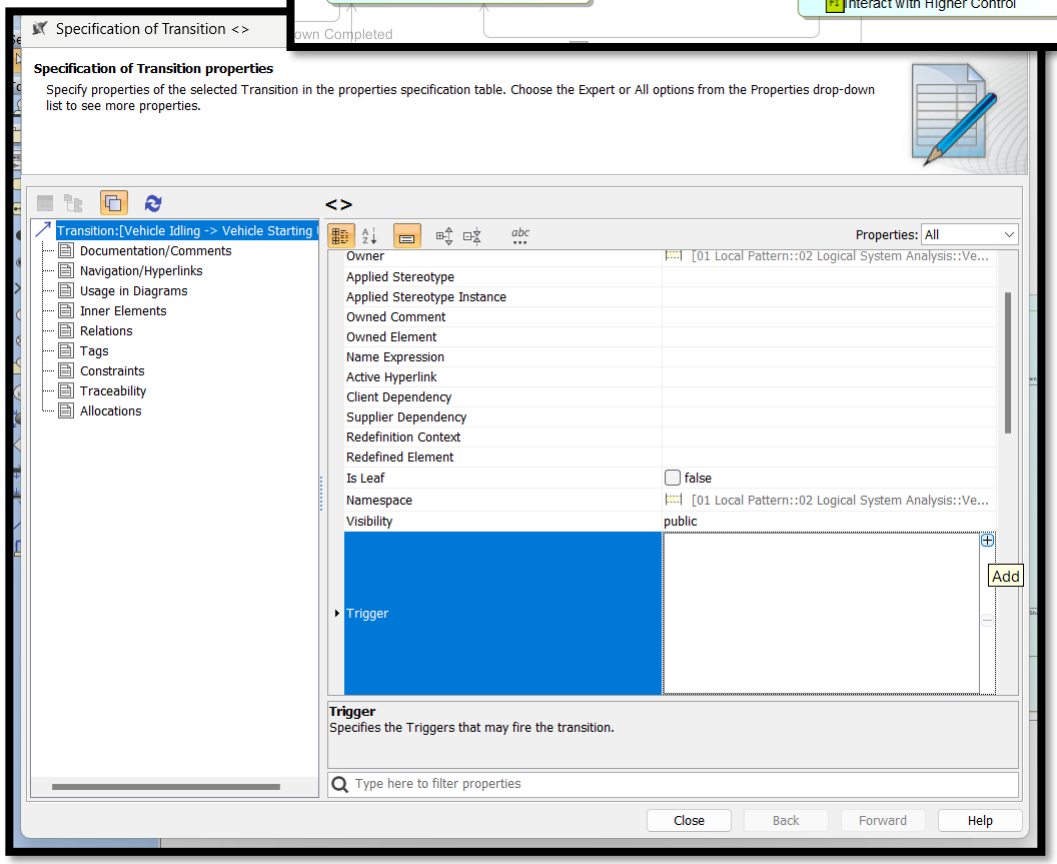


State Machine: Creating State-State Transitions, Events

- 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.

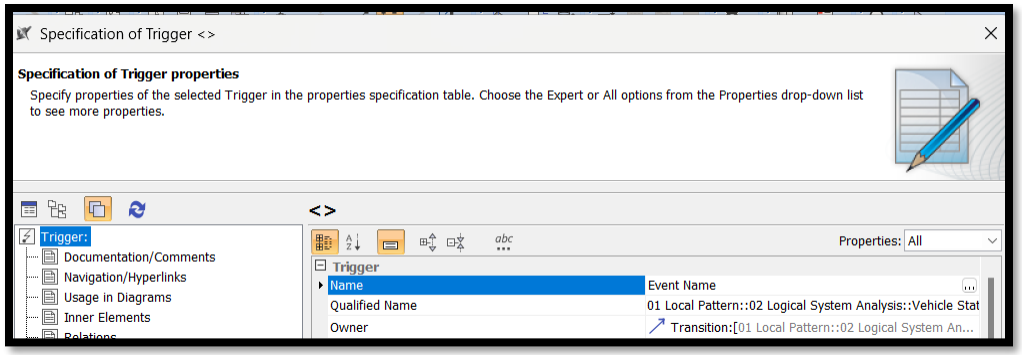


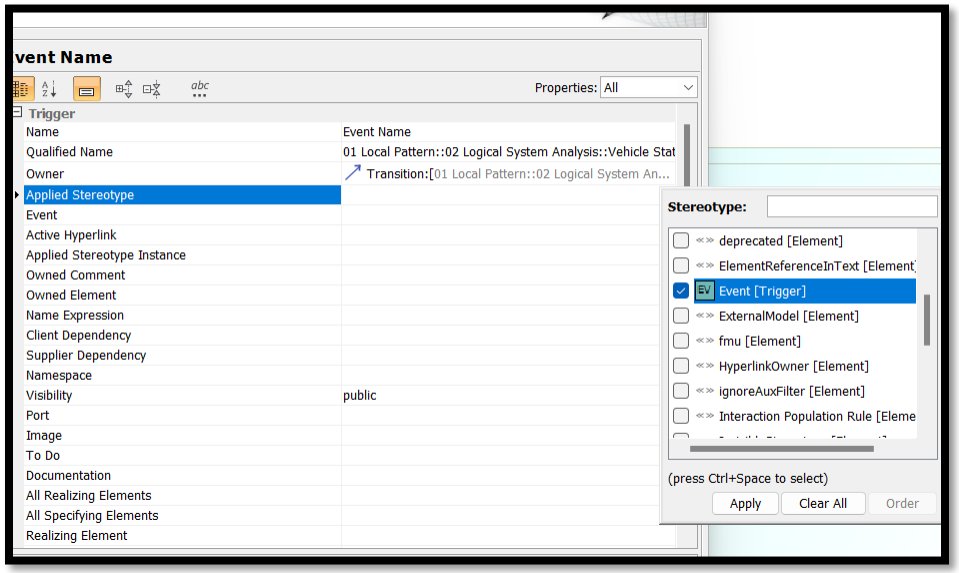




State Machine: Creating State-State Transitions, Events

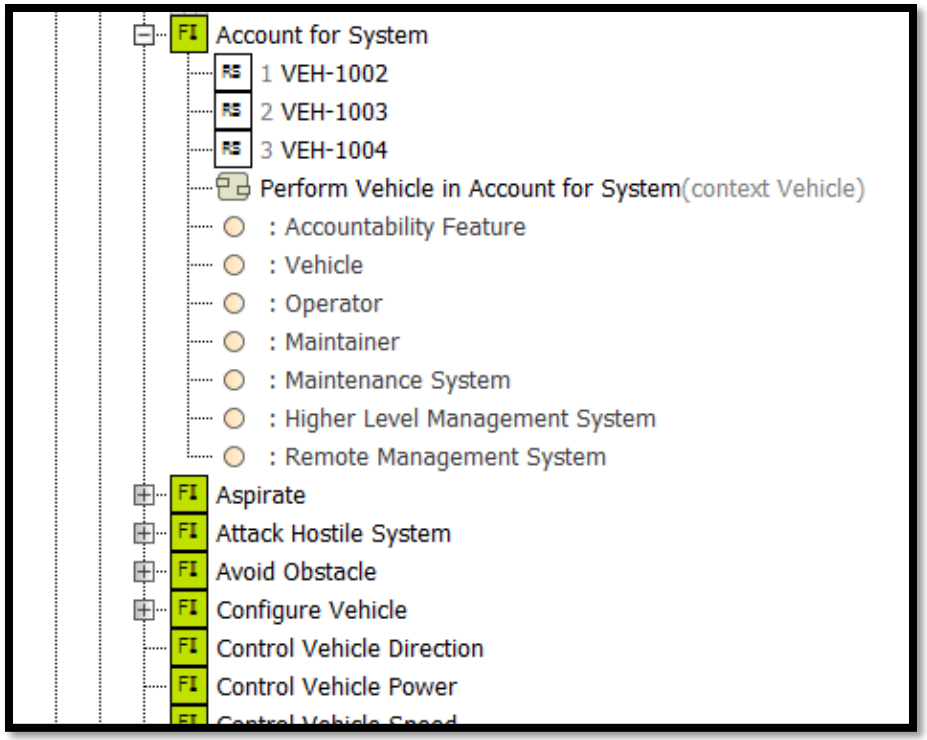
- In the new Trigger Specification window that opens enter the name of the event in the Name property.
- Select the Applied Stereotype property and choose the Event stereotype option.





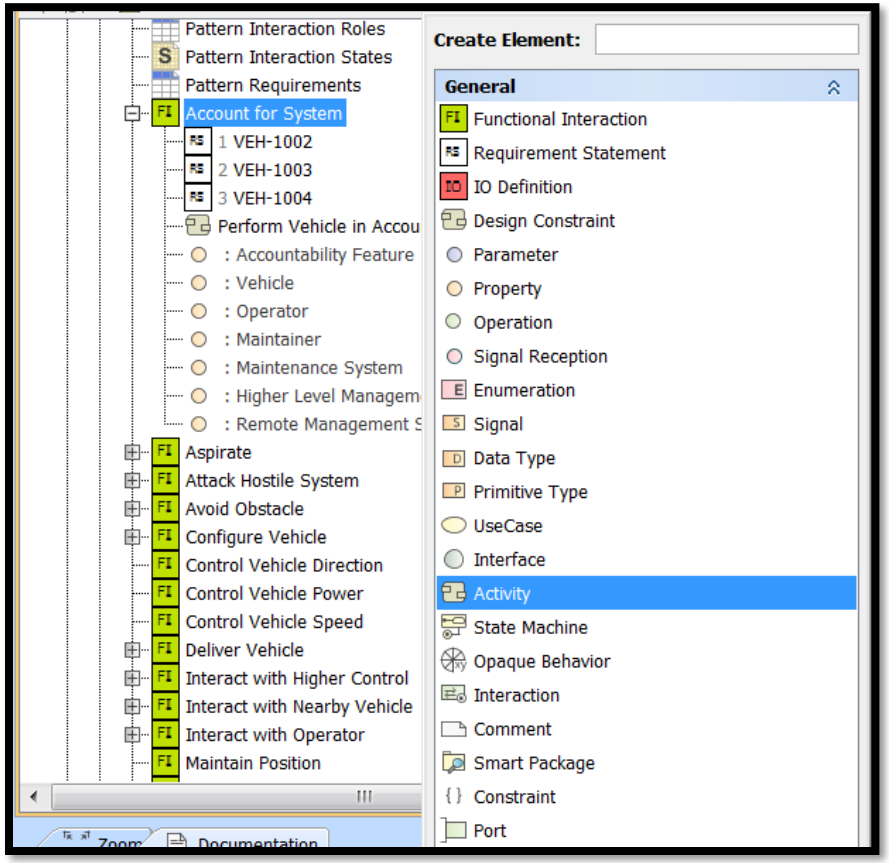
Detail Requirements: Overview

- 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.



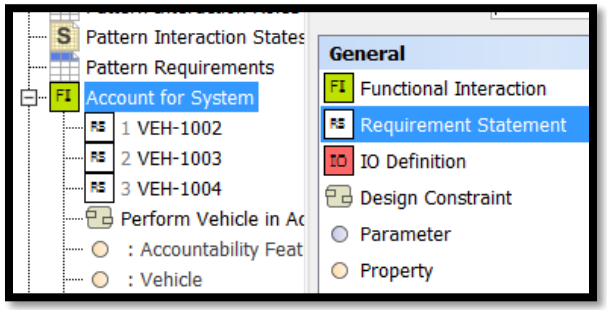
Detail Requirements:
Creating Requirement Transfer Functions

- 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

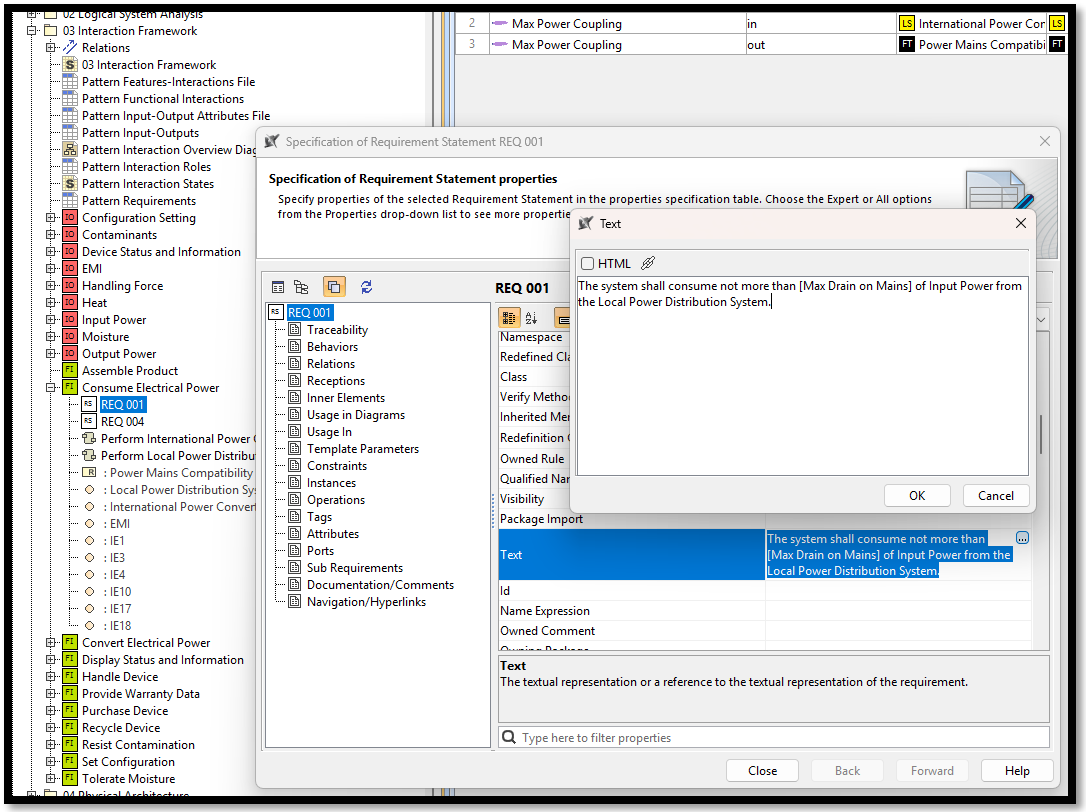
- Use ~Create Element and choose Requirement Statement
- Name the new Requirement Statement with its Requirement ID.
- Create a Satisfy dependency to the new Requirement Statement from the Requirement Transfer Function it specifies.
- At this point, the Requirement row will appear in the Pattern Requirements 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	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].

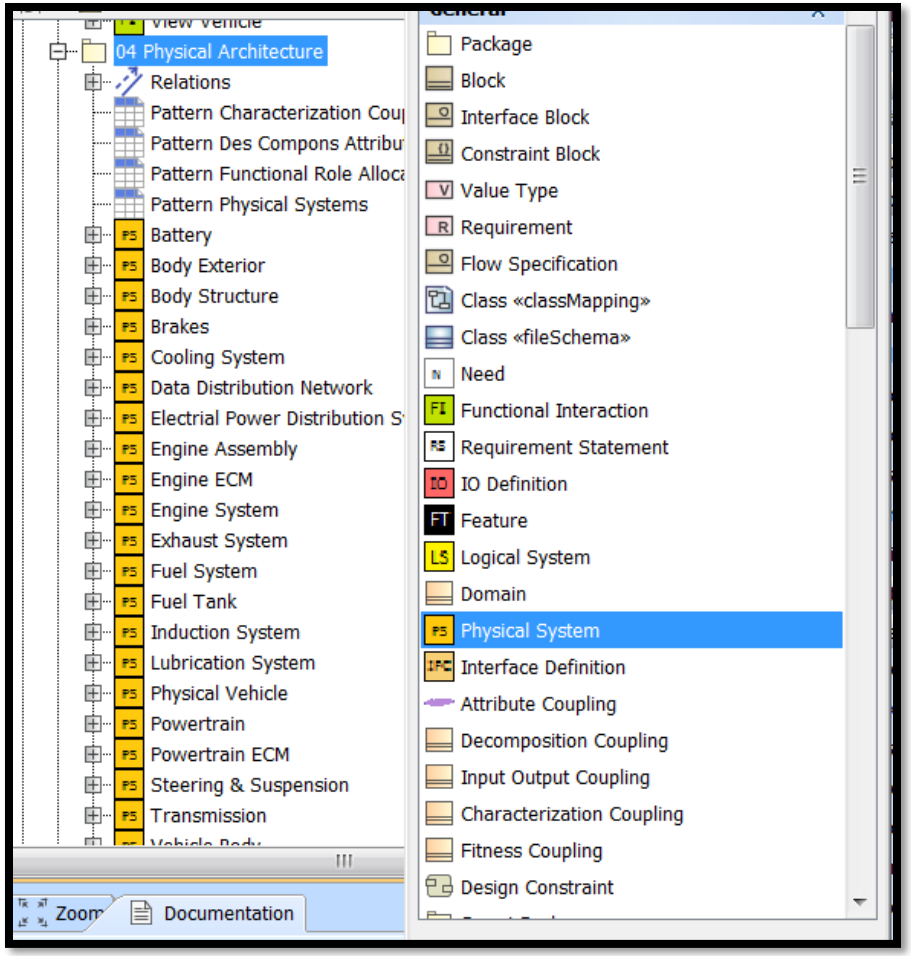
Detail Requirements: Creating Requirement Statements

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



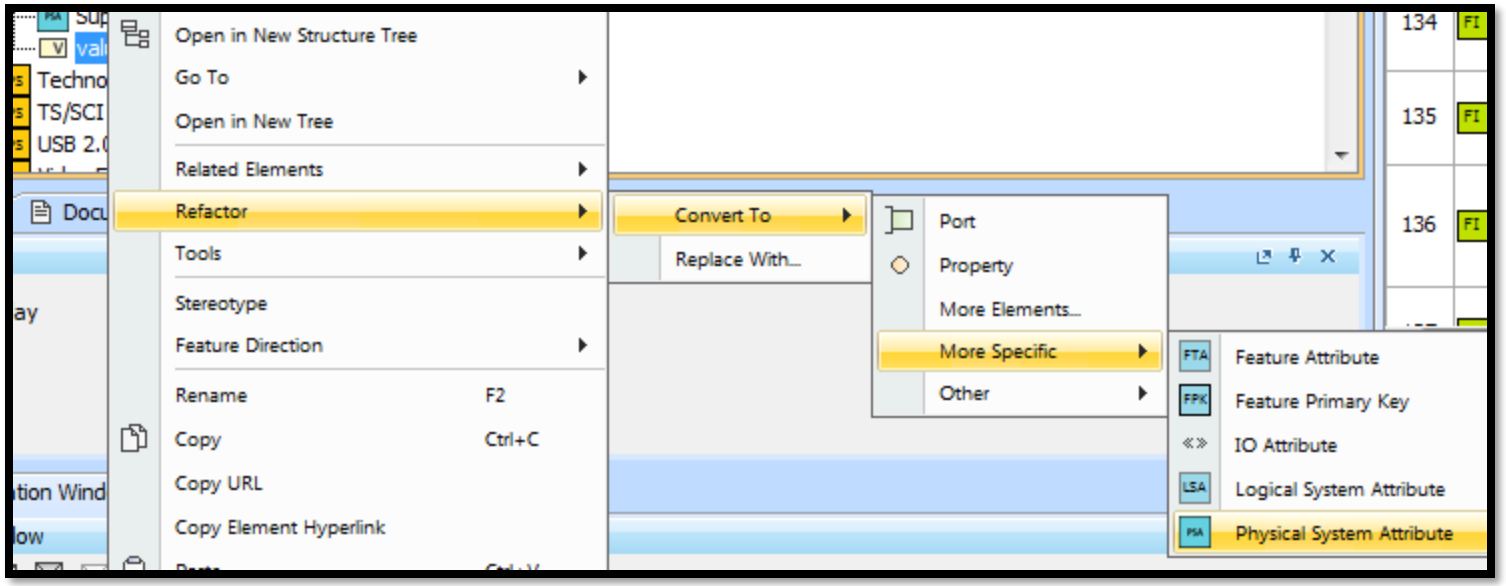
High Level Design:
Creating Design Components (Physical Systems)

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



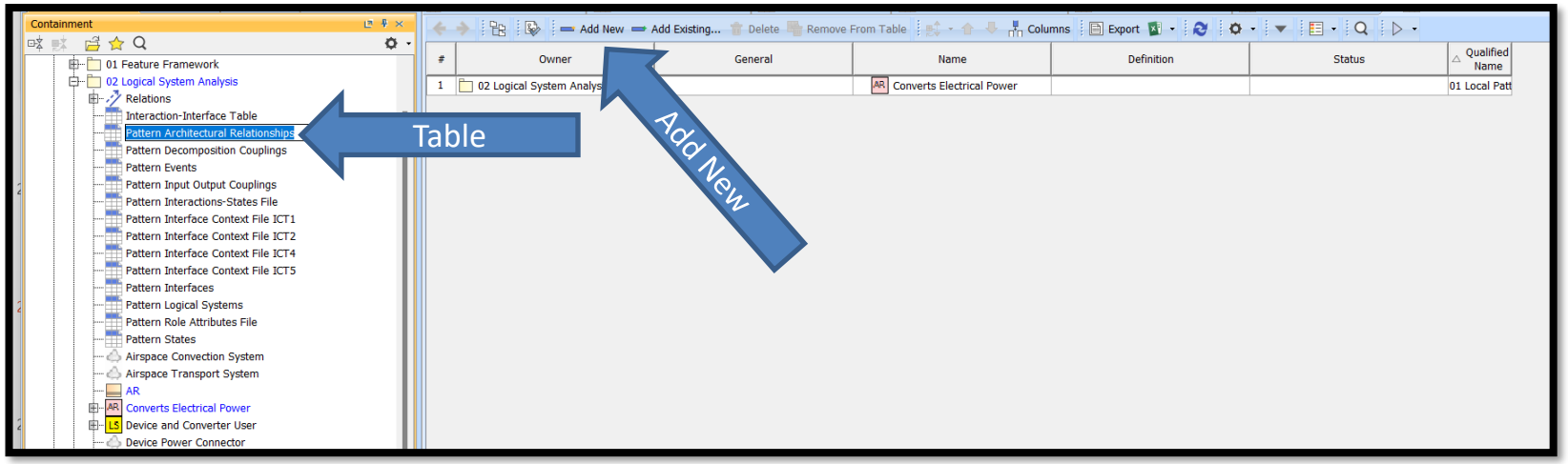
High Level Design: Creating Design Component (Physical System) Attributes

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



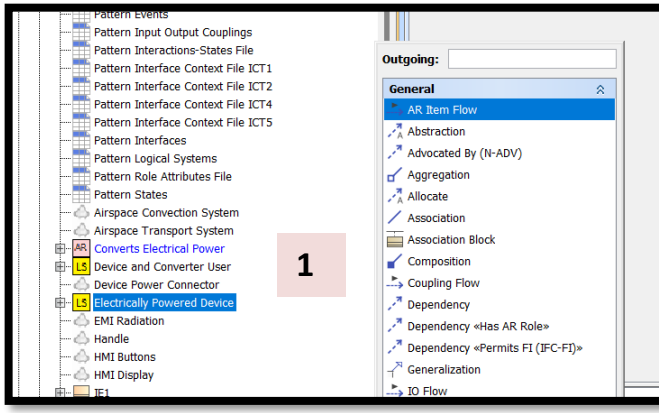
Interface Context: Architectural Relationships

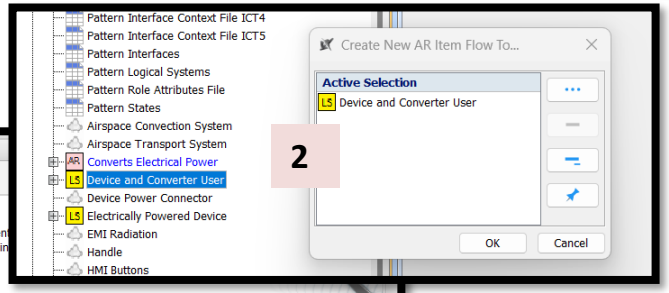
- Architectural Relationships are created in package 01 Local Pattern::02 Logical System Analysis.
- Select and Open the Pattern Architectural Relationships Table and then Select Add New.
- The newly added Architectural Relationship will show up in the table and the Containment Browser

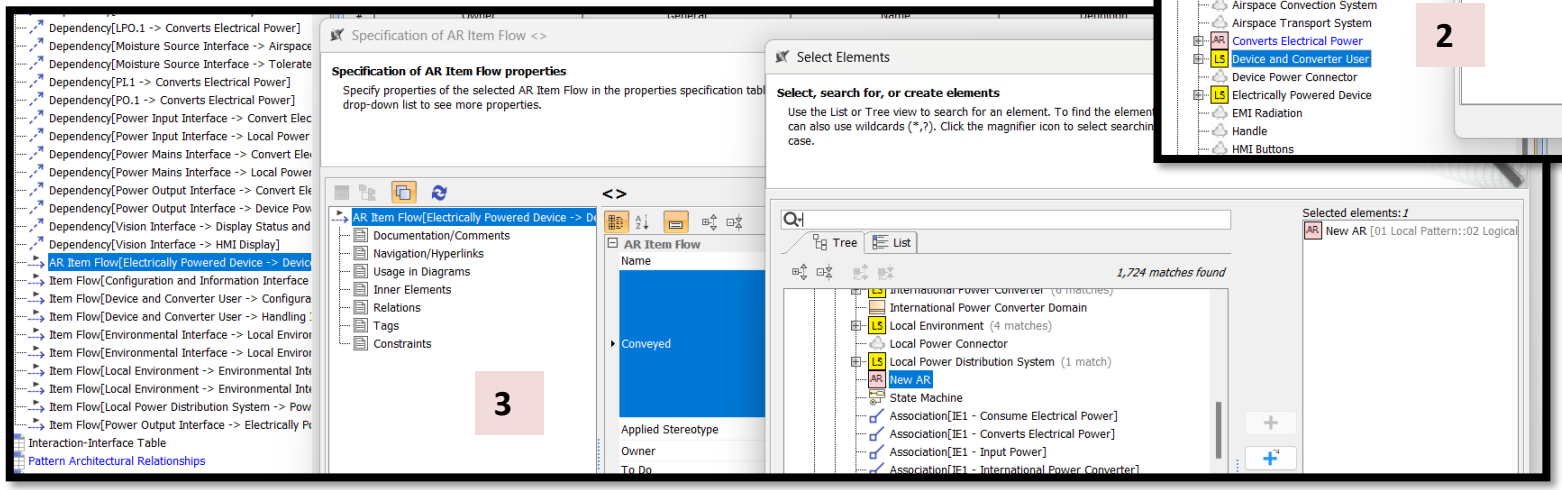


Interface Context: Architectural Relationships

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.

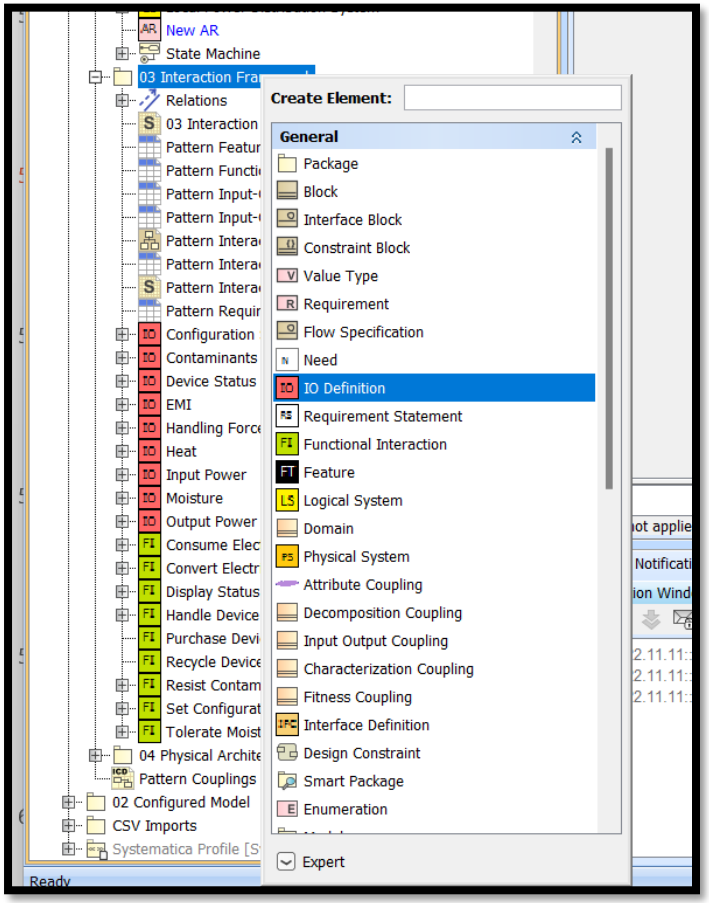






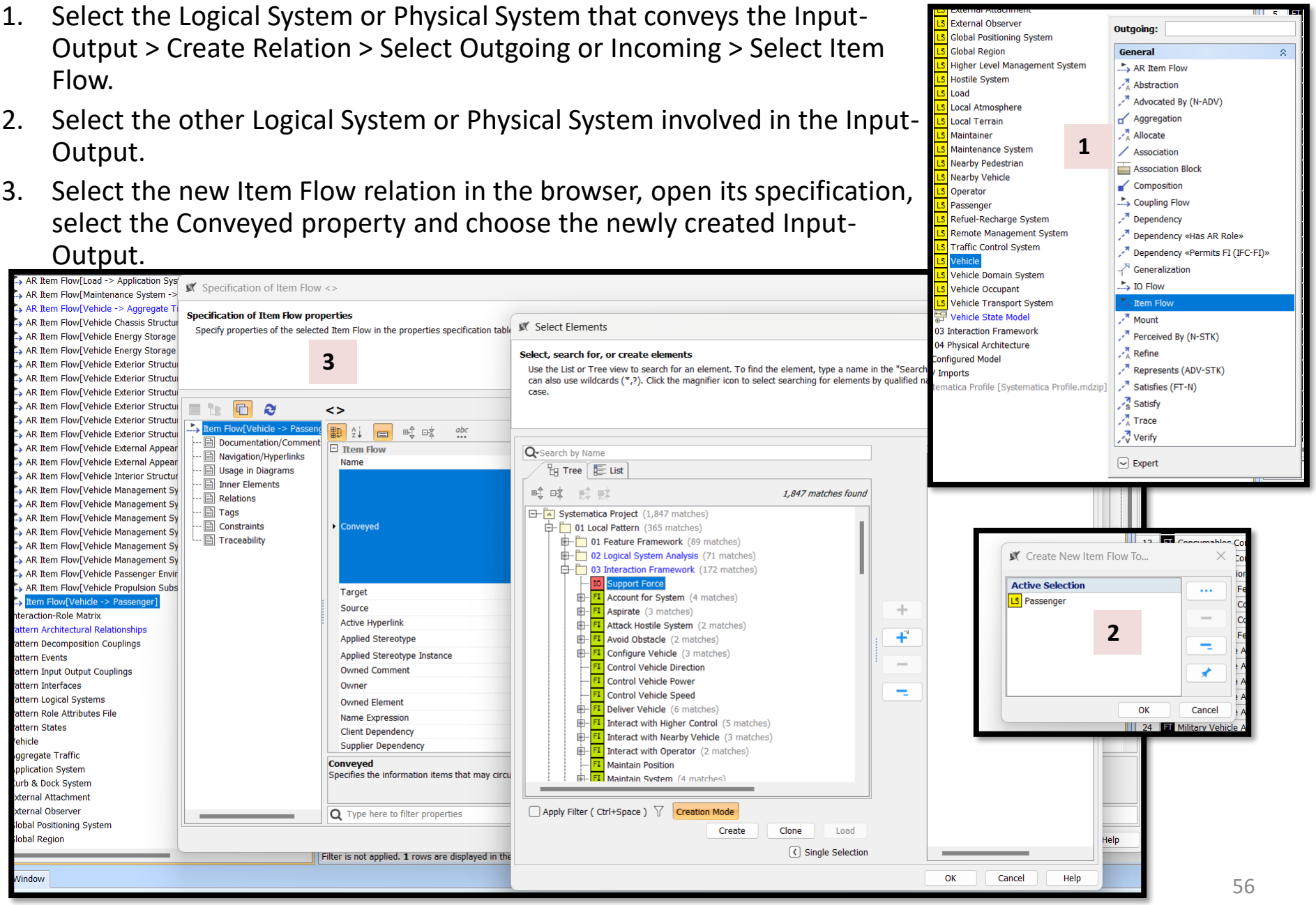
Interface Context: Populating Input-Outputs

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



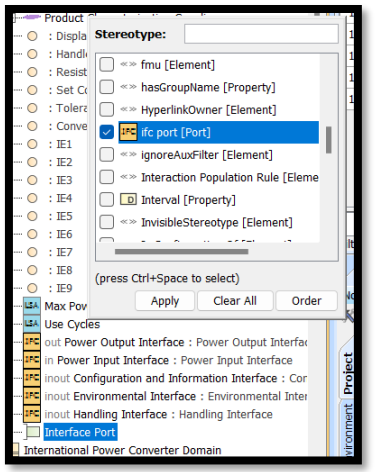
Interface Context: Populating Input-Outputs continued

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.

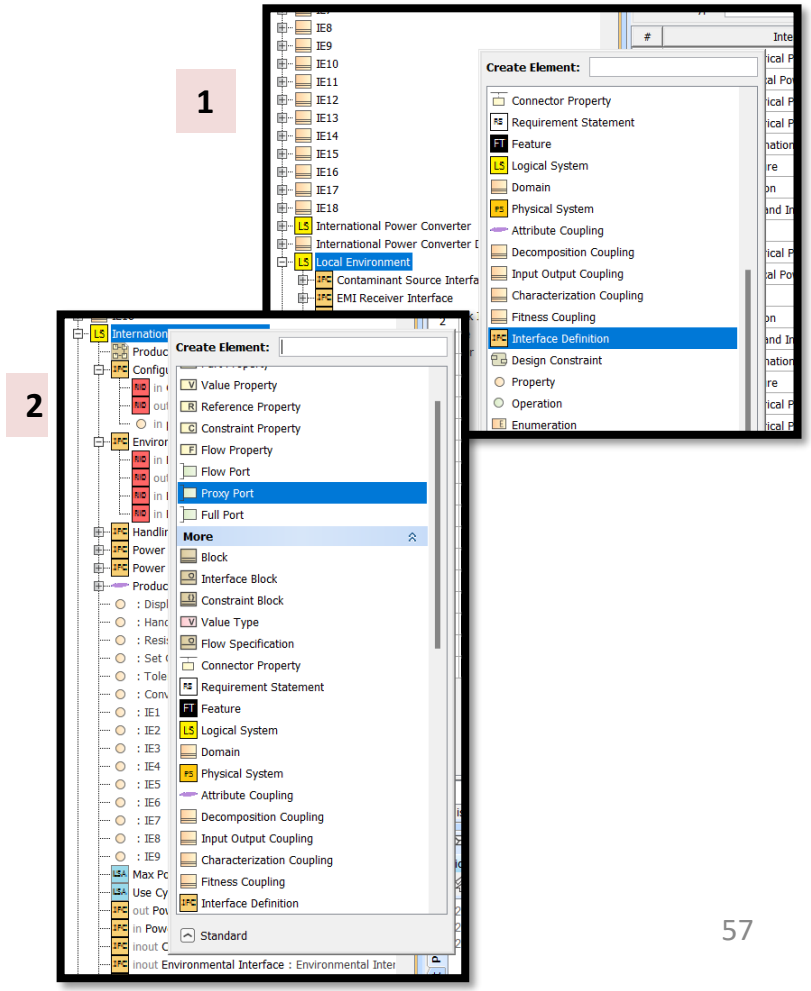


Interface Context: Populating Interfaces

- 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.

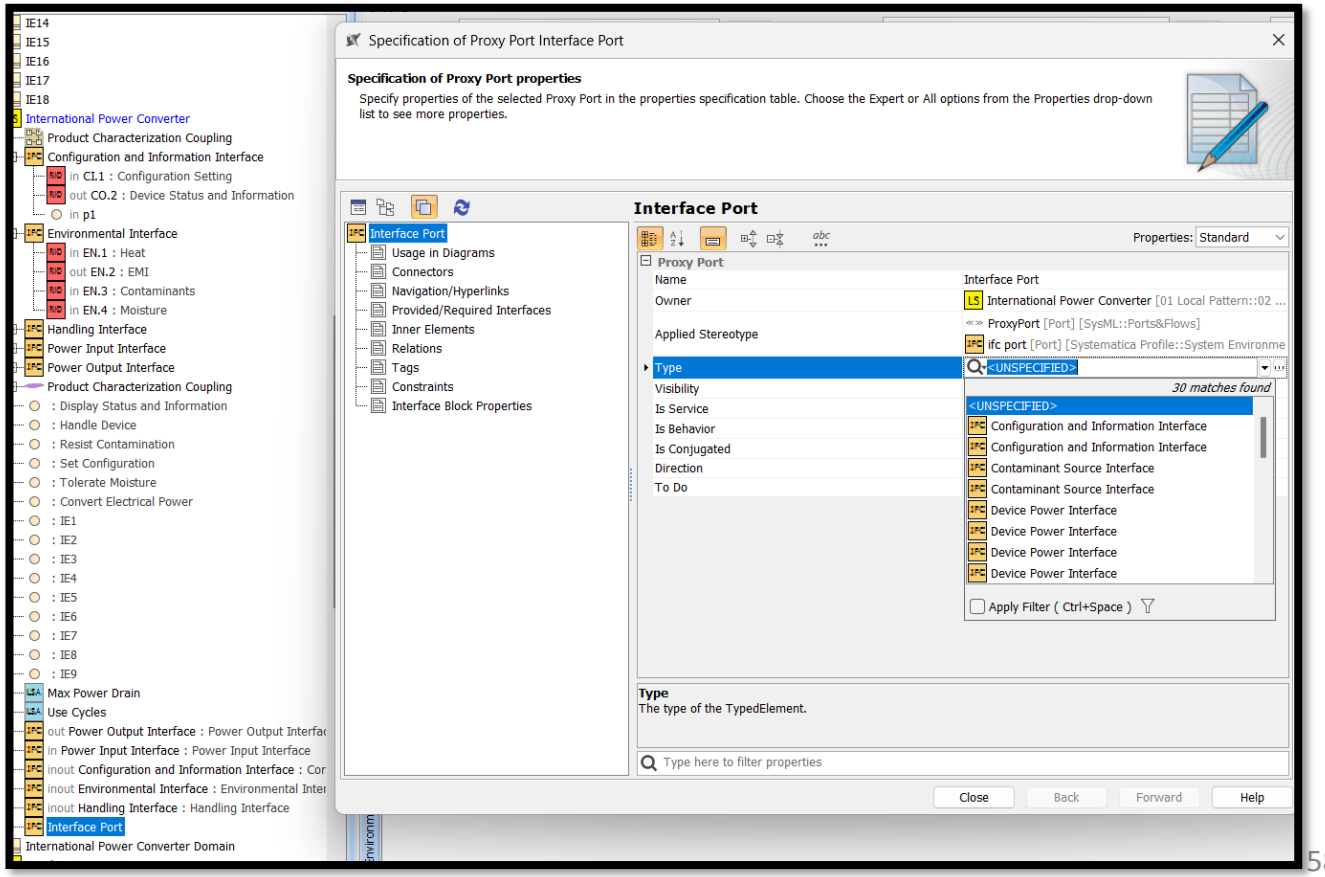


3



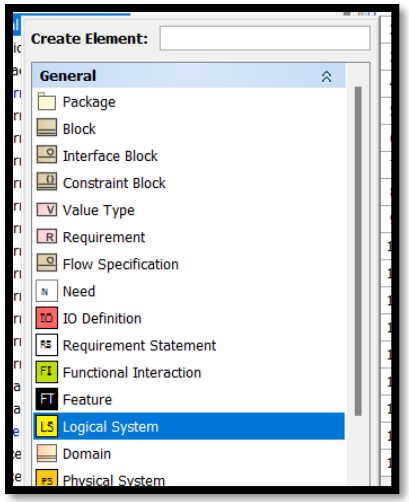
Interface Context: Populating Interfaces, continued

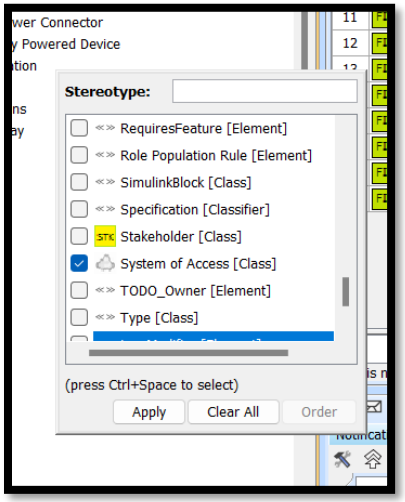
- Select the new ifc port in the browser, open its specification, select the Type property and choose the associated Interface Definition with the same name.



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.



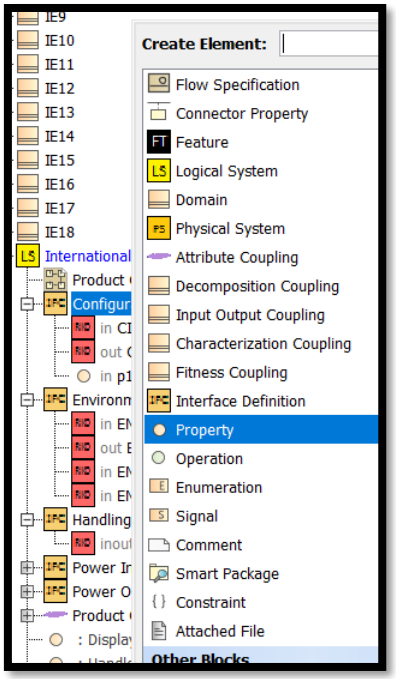


Interface Context: Populating Ports

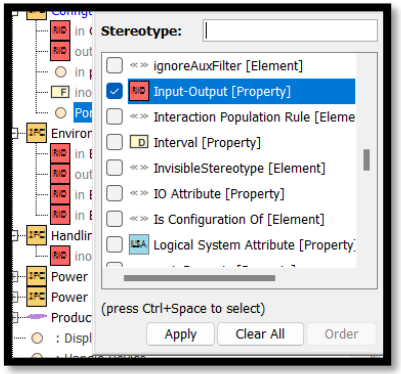
- Ports are created in package 01 Local Pattern::02 Logical System Analysis under the Interface Definition that owns the Port:

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.

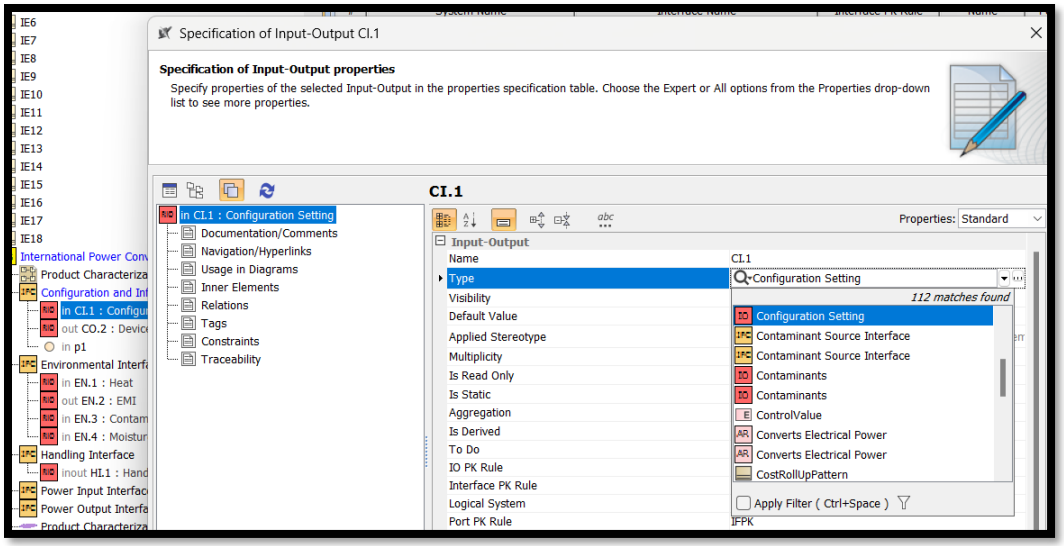
1



2

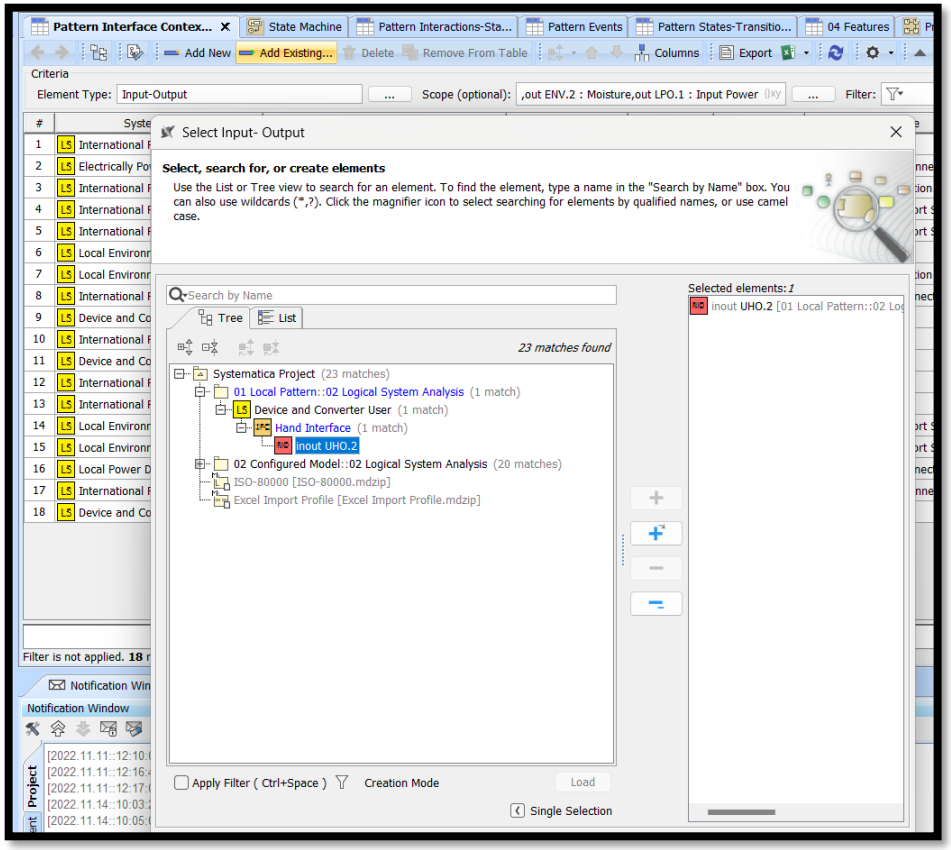


3



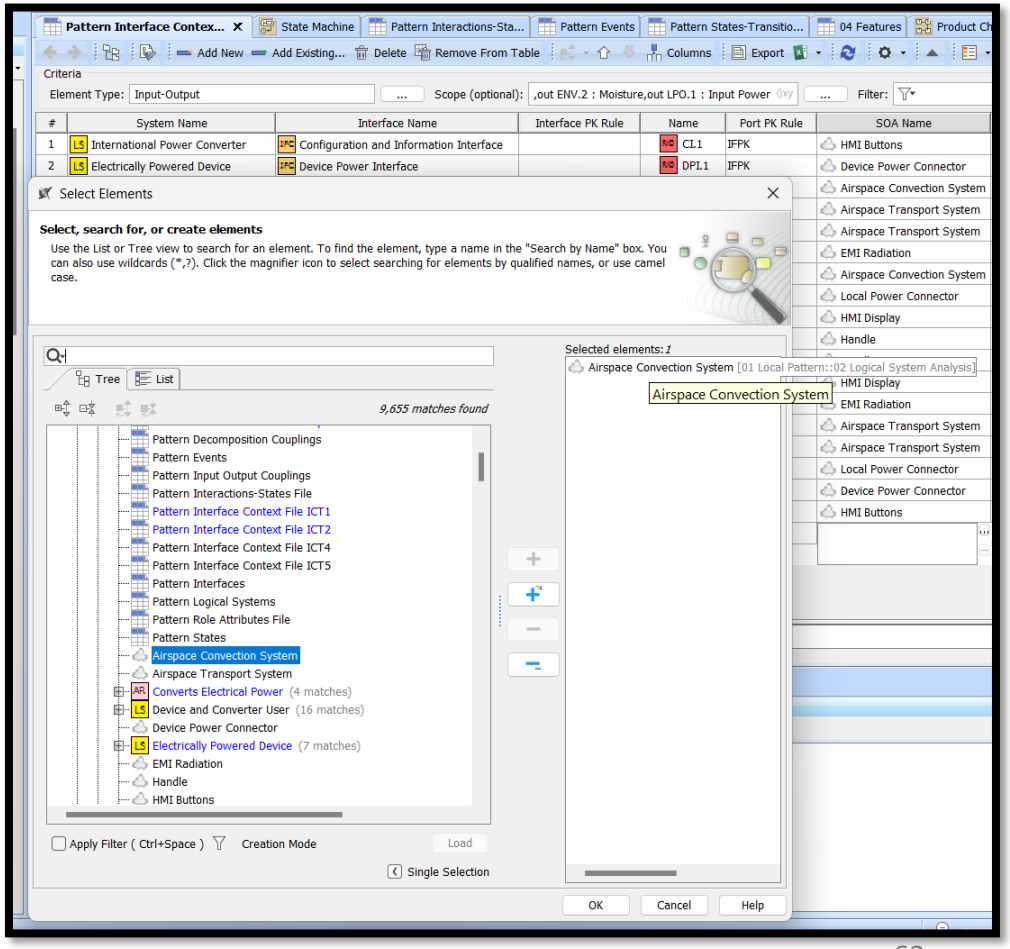
Interface Context: Ports, Table ICT1

- 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.



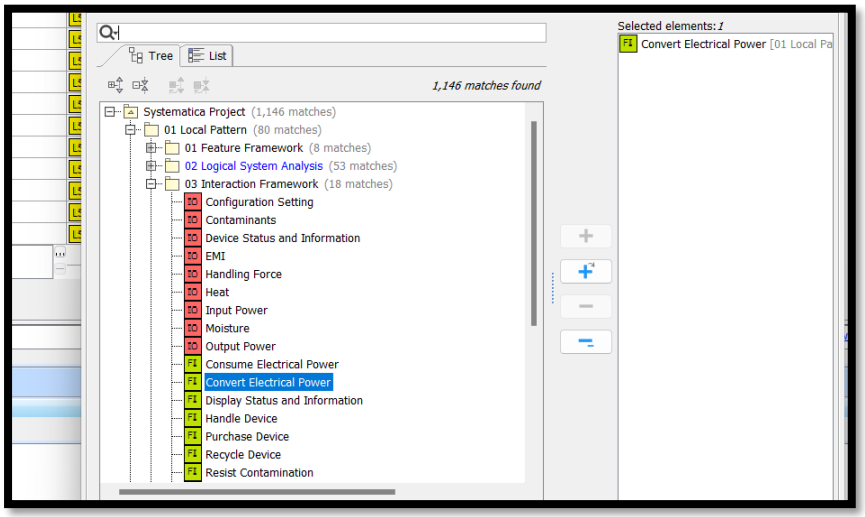
Interface Context: Ports, Table ICT1

#	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	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		

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

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		Local Environment	EMI	
18	IE18		Local Environment	Heat	

IE Name

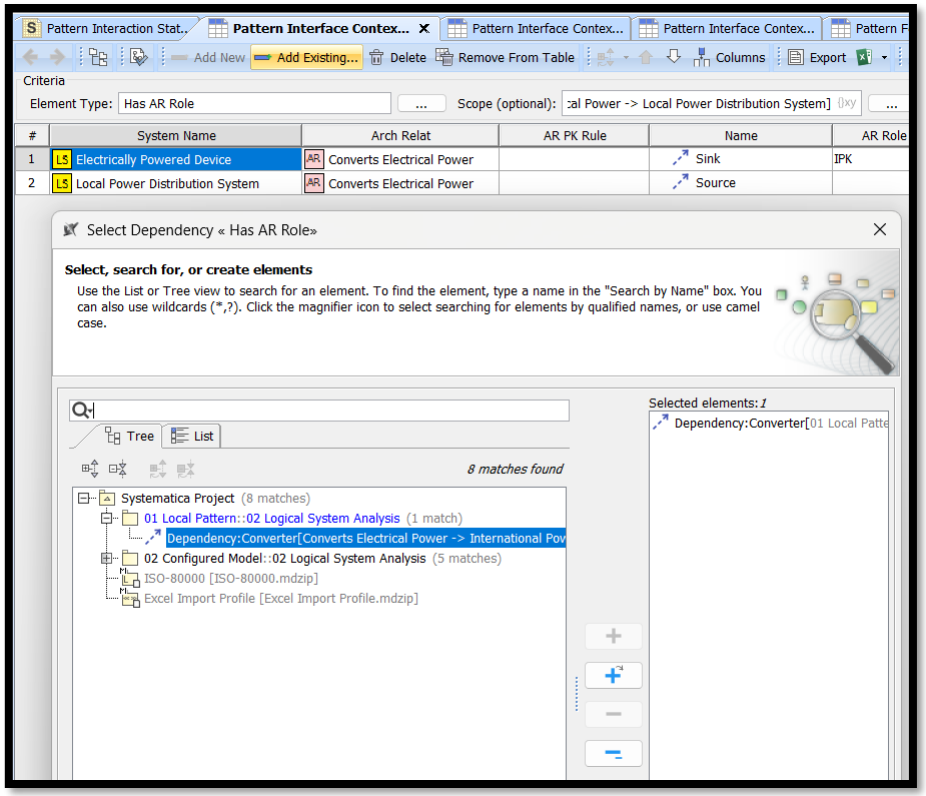
Interface Context: Ports, Table ICT2

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



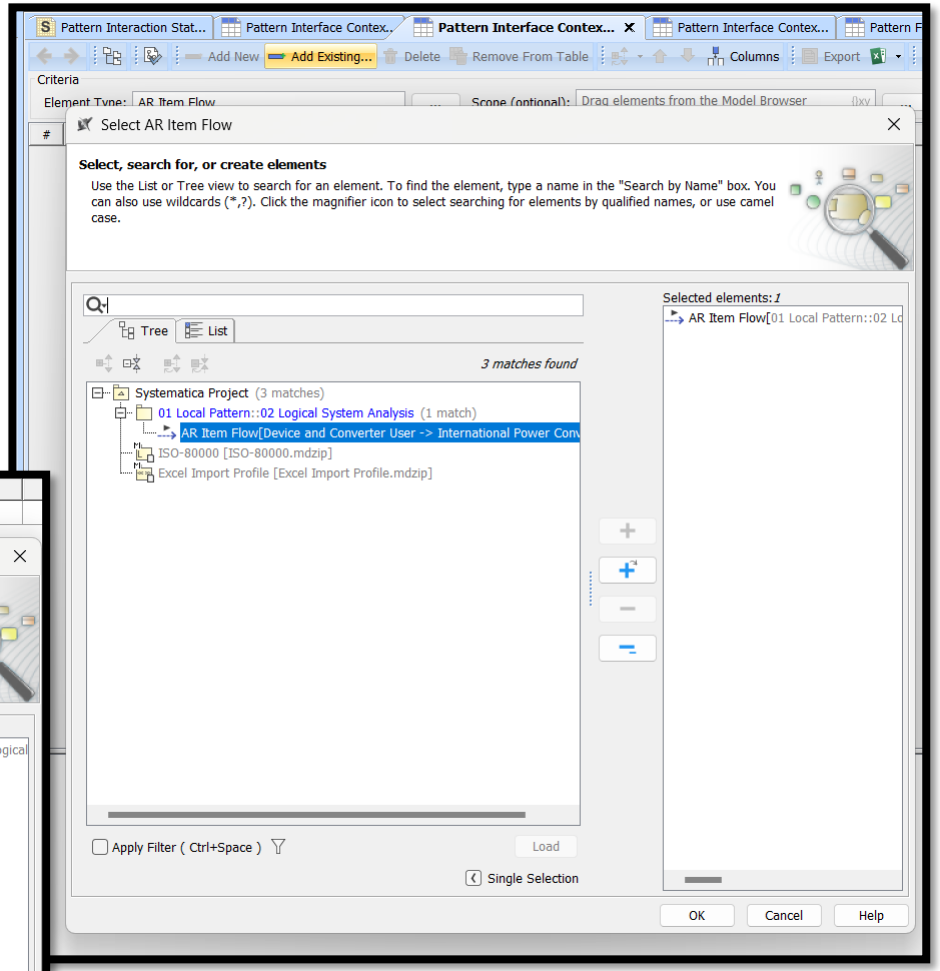
Interface Context: Architectural Relationships, Table ICT4

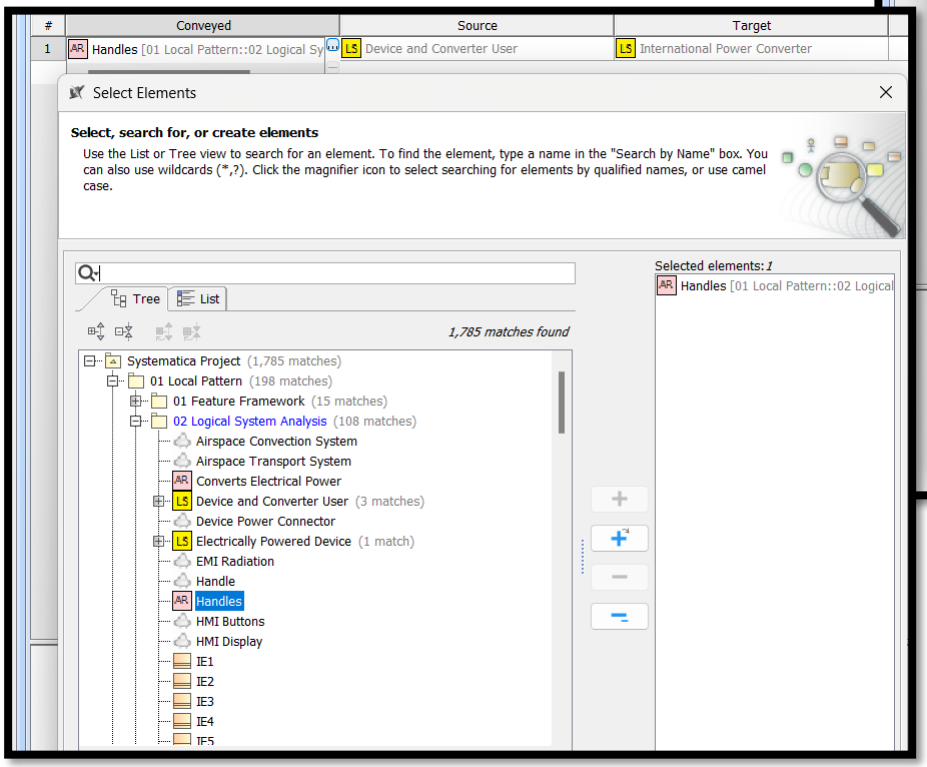
#	System Name	Arch Relat	AR PK Rule	AR Role Name	AR Role PK Rule	AR Internal or External	AR Complexity
1	LS Electrically Powered Device	AR Converts Electrical Power		Sink	IPK		Reified
2	LS Local Power Distribution System	AR Converts Electrical Power		Source			Reified
3	LS International Power Converter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





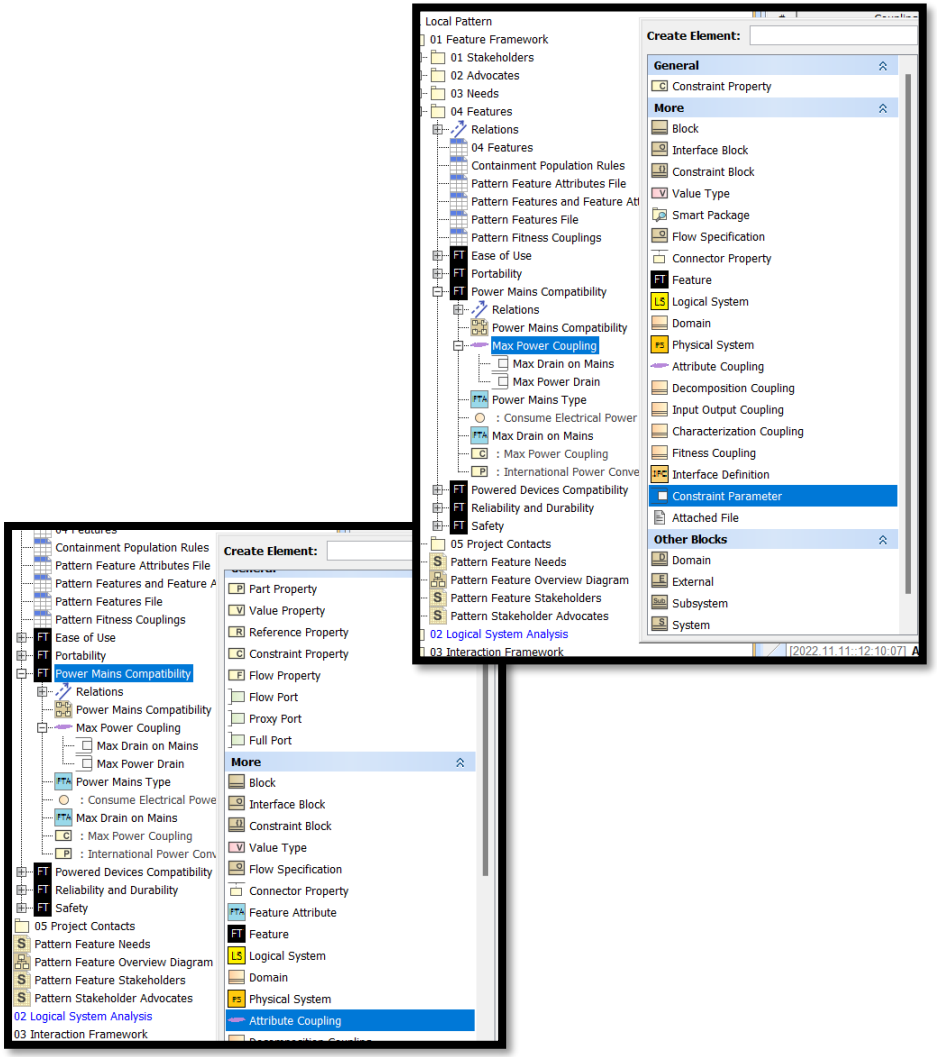
Interface Context: Architectural Relationships, Table ICT5

#	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

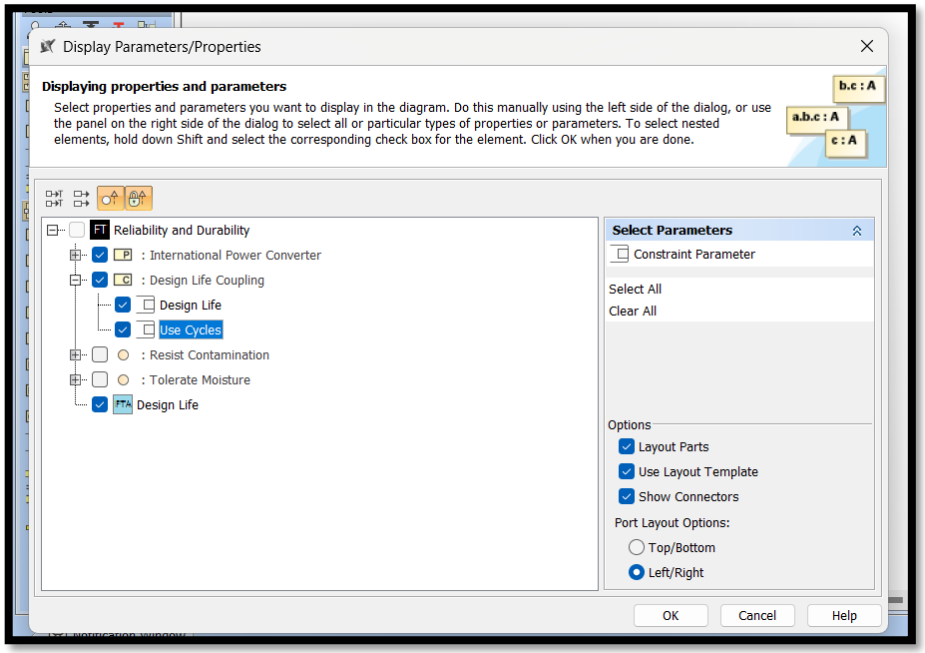
Attribute Couplings

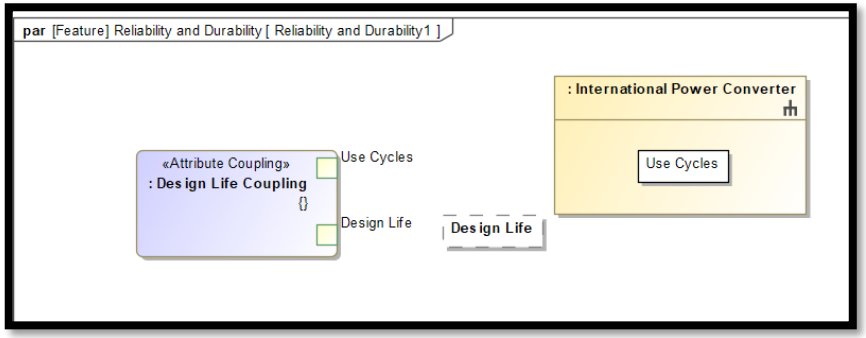
- Attribute Couplings are created under the element that owns them in various packages including Features, Roles, Design Components, and Input-Outputs:
  - A given Attribute Coupling involves multiple Attributes, so which one belongs to the Coupling's owner?
  - Each Coupling drives (in a causality sense) 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).

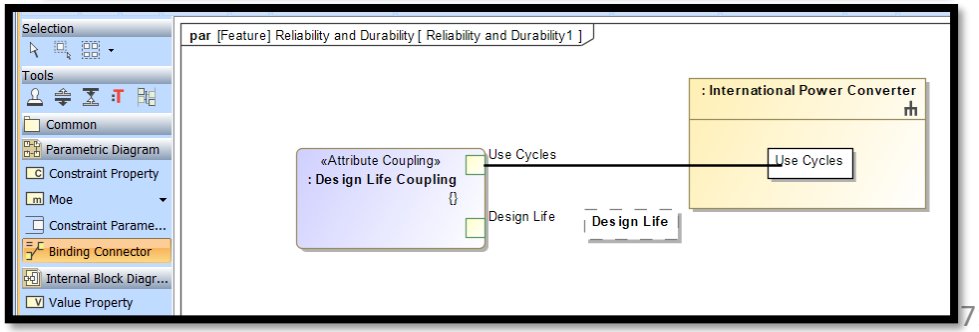


Attribute Couplings

- 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.

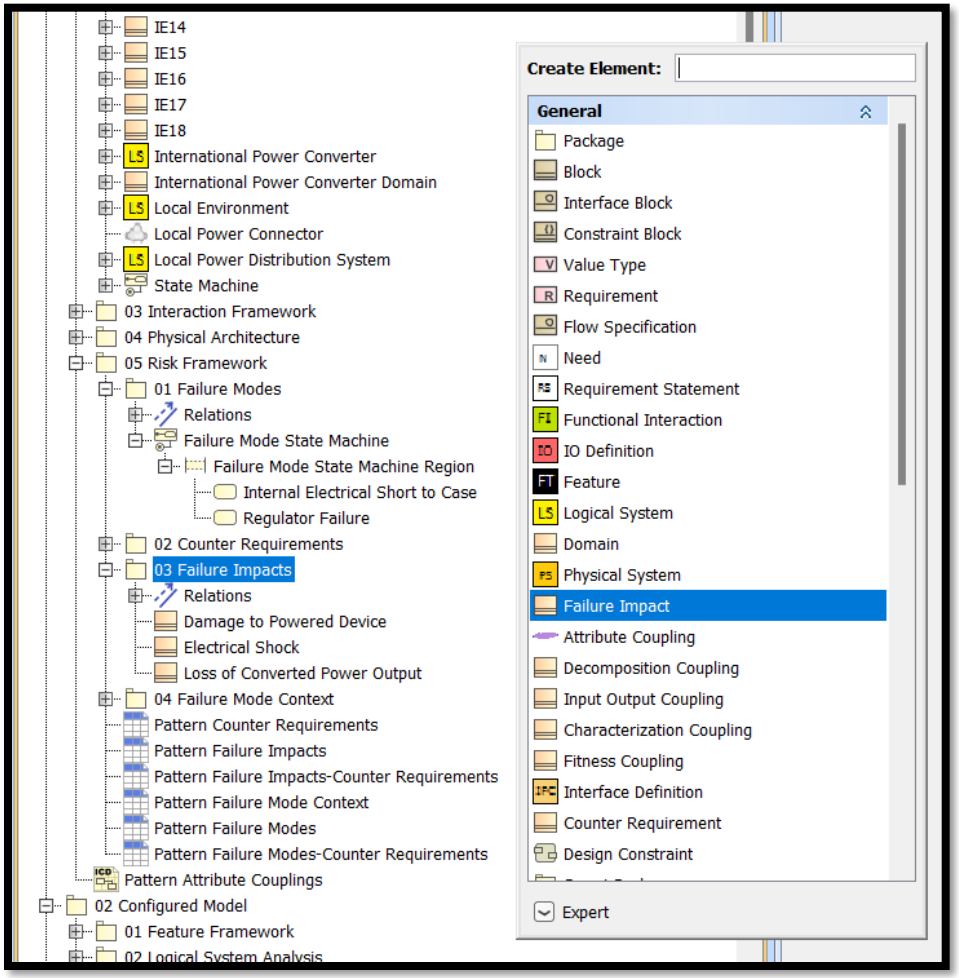






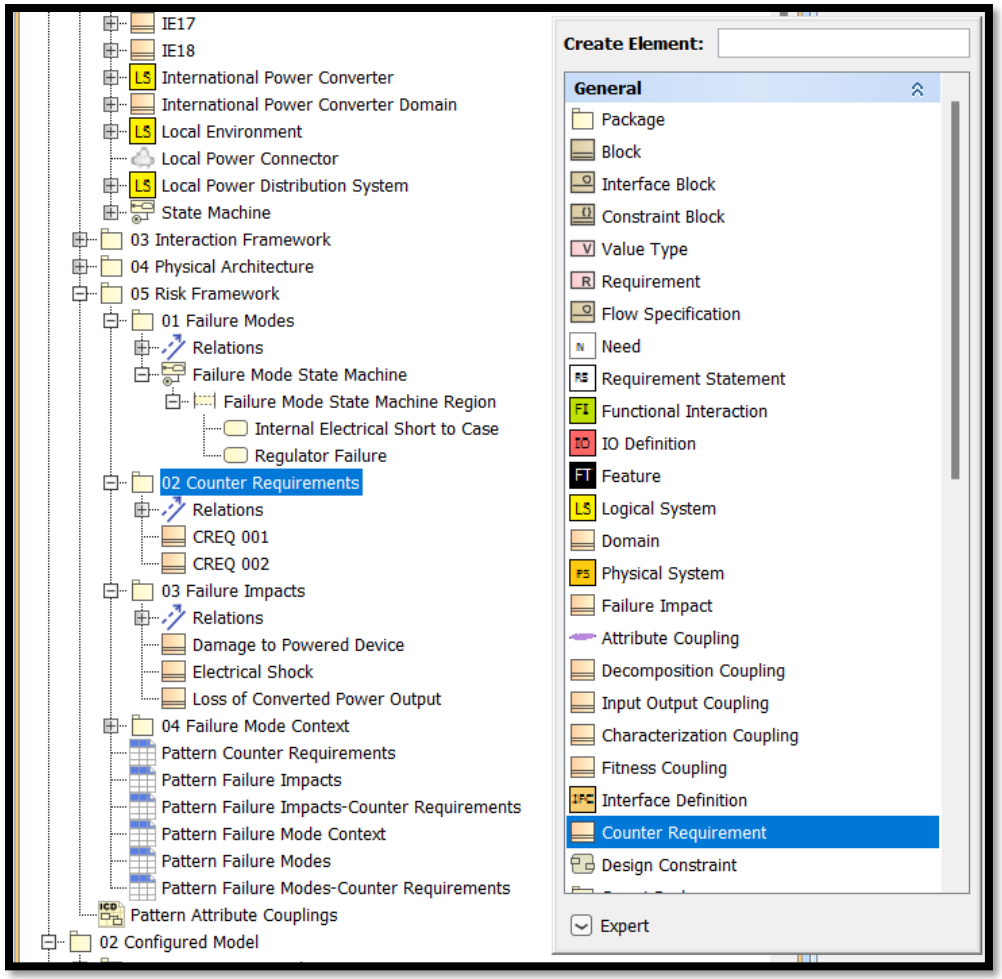
Failure Analysis: Creating Failure Impacts

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



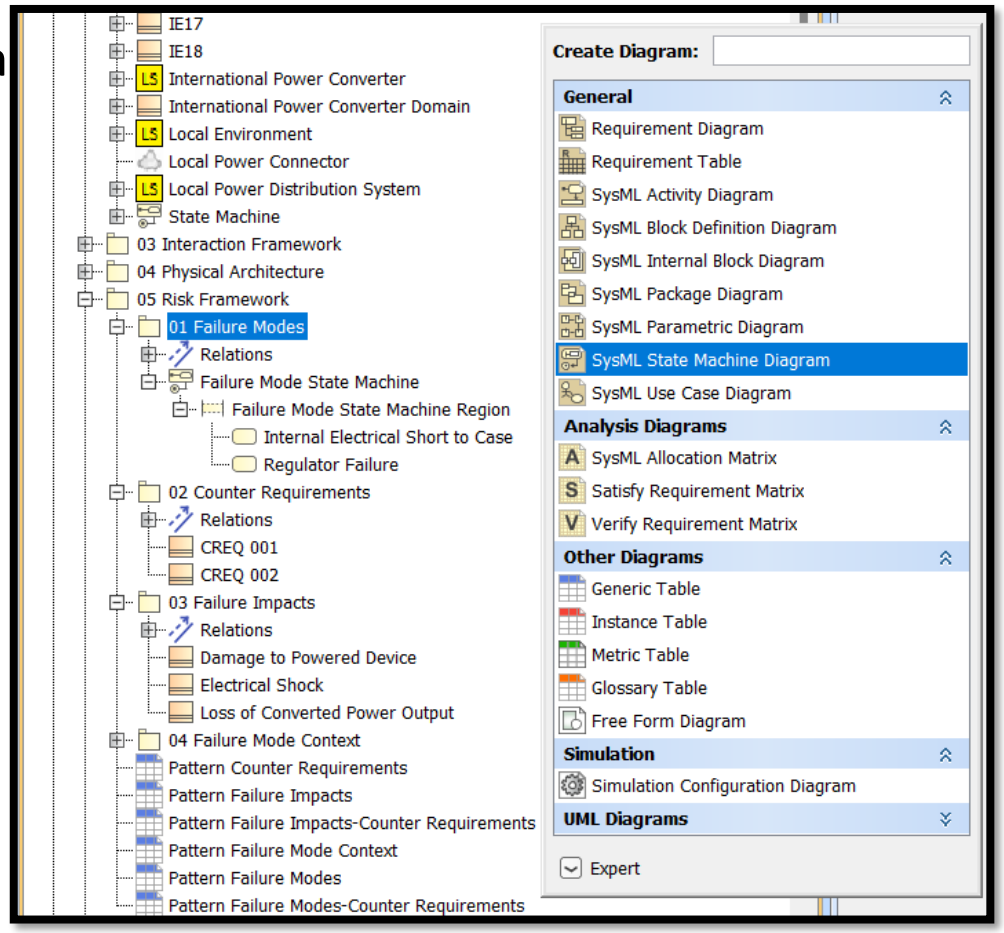
Failure Analysis: Creating Counter Requirements

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



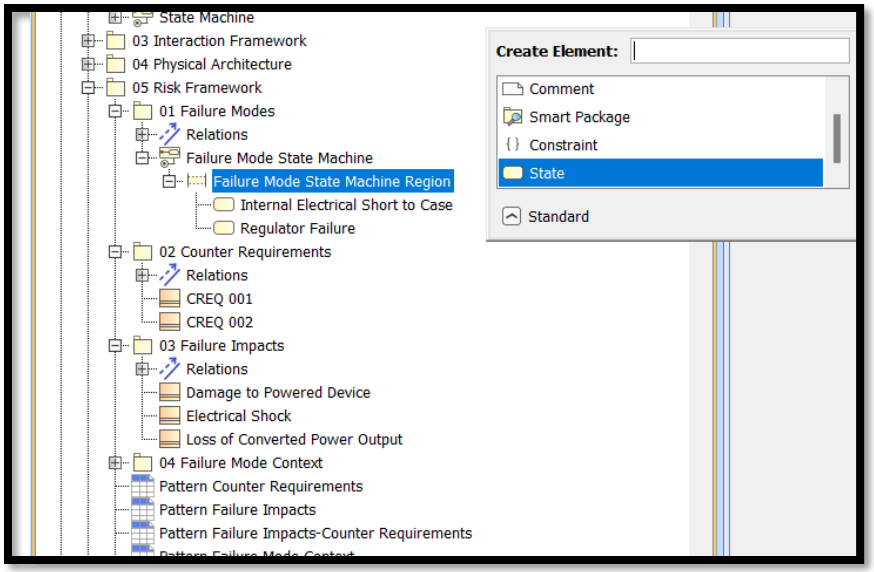
Failure Analysis: Creating Failure Mode State Machine and Failure Modes

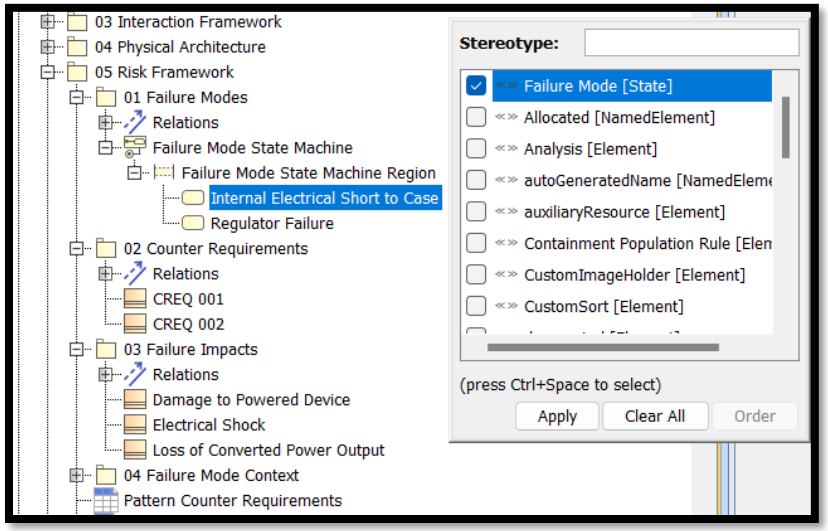
- Pattern Failure Mode State Diagram and Failure Modes are created 01 Local Pattern::05 Risk Framework::01 Failure Modes
- Use ~Create Diagram and choose SysML State Machine Diagram
- Name the Diagram “Failure Mode State Machine”.
- Name the Region “Failure Mode State Machine Region”.



Failure Analysis: Creating Failure Modes

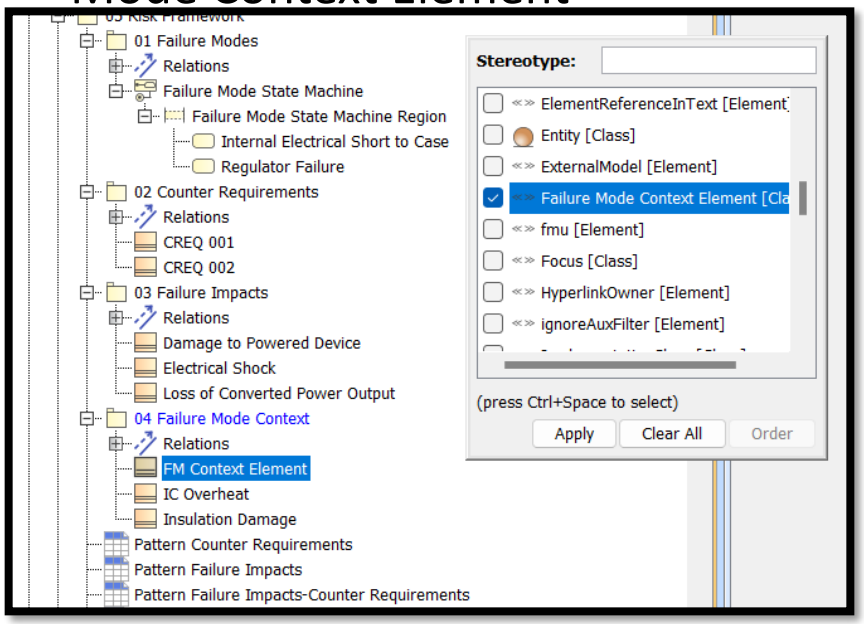
- Right-click the automatically created Region under the State Model and use ~Create Element and choose State. Make sure the menu is expanded to Expert.
- Right-click the newly created State, select Stereotype, and change it to Failure Mode as shown below.

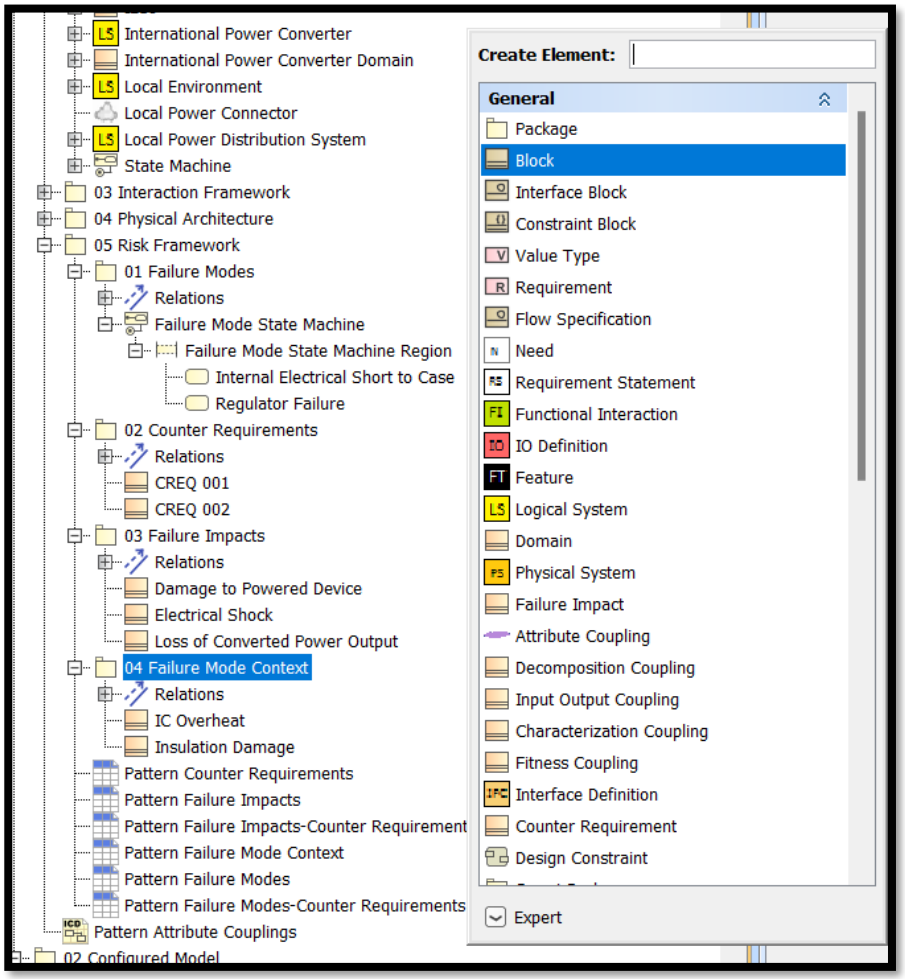




Failure Analysis: Creating Failure Mode Context

- 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



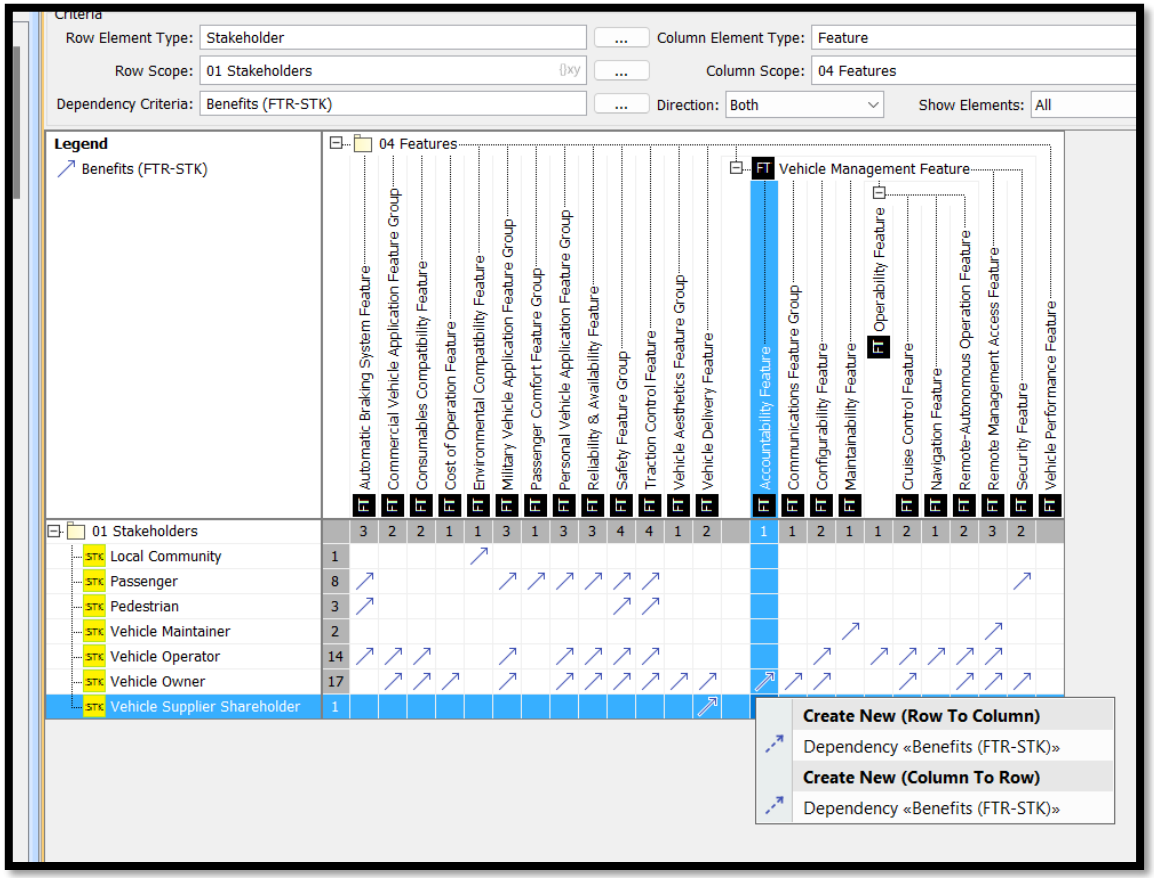


4. Creating S* Pattern Relationships

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.

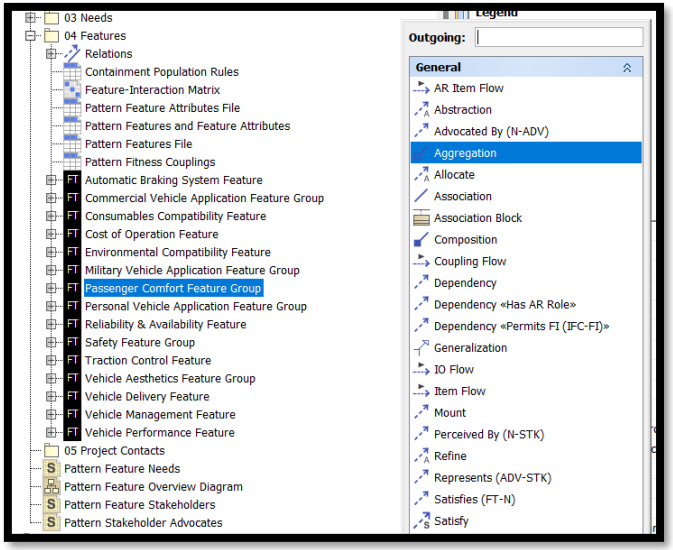


Metarelationships: Creating Feature-Interaction Relationships

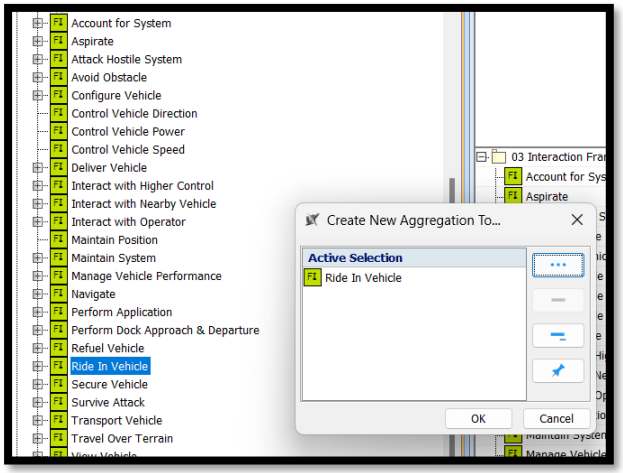
- Relationships between Features and Functional Interactions are SysML aggregations.
- Create an Association in which the Feature aggregates the Interaction.
- Add a “Uses Functional Interaction” stereotype to the new Association

Metarelationships: Creating Feature-Interaction Relationships

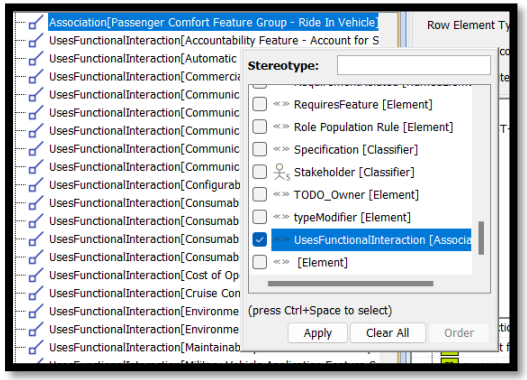
1. Right-Click the Feature, ~Create Relation, Select Outgoing, Choose Aggregation
2. Select the related Interaction(s)
3. Right-Click the new relationship, ~Stereotype, Select “Uses Functional Interaction”



1



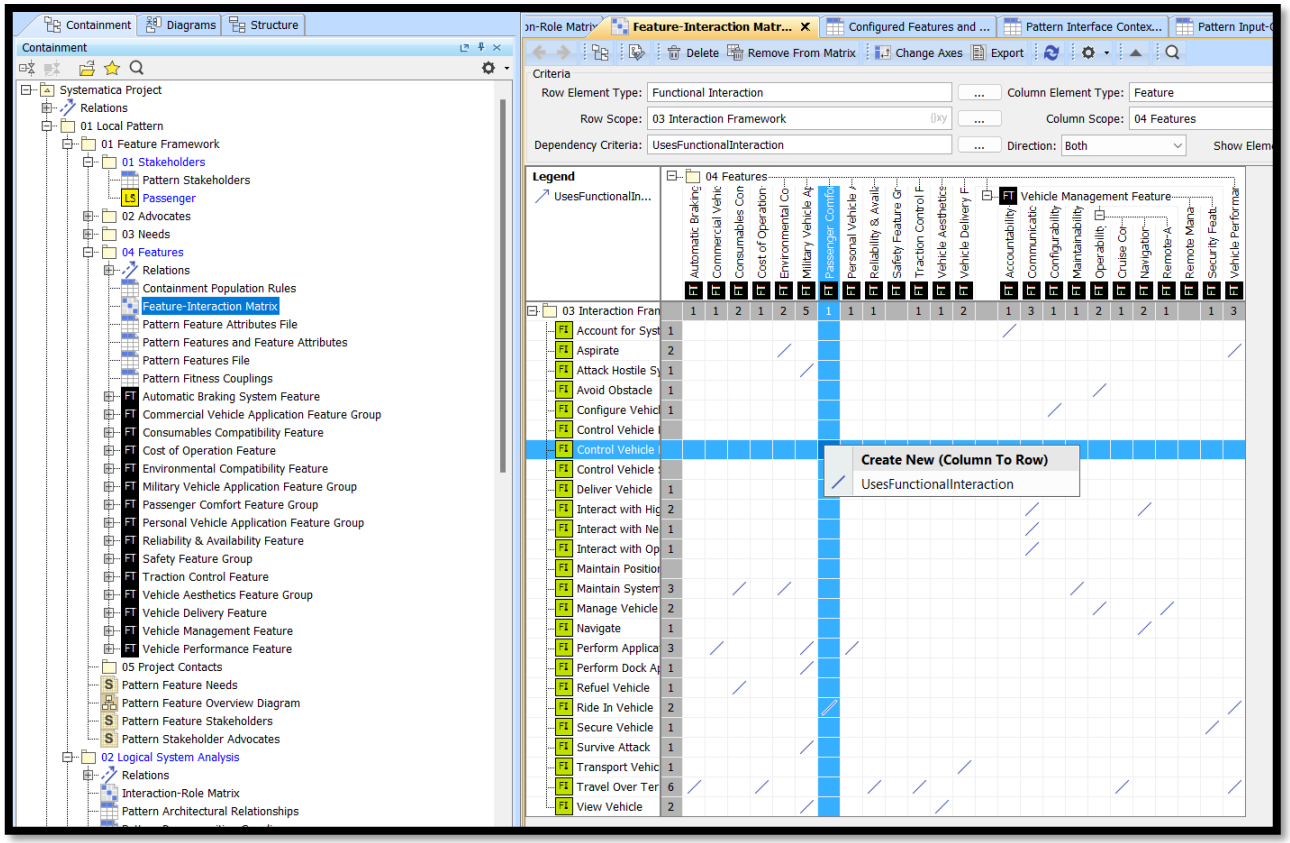
2



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.



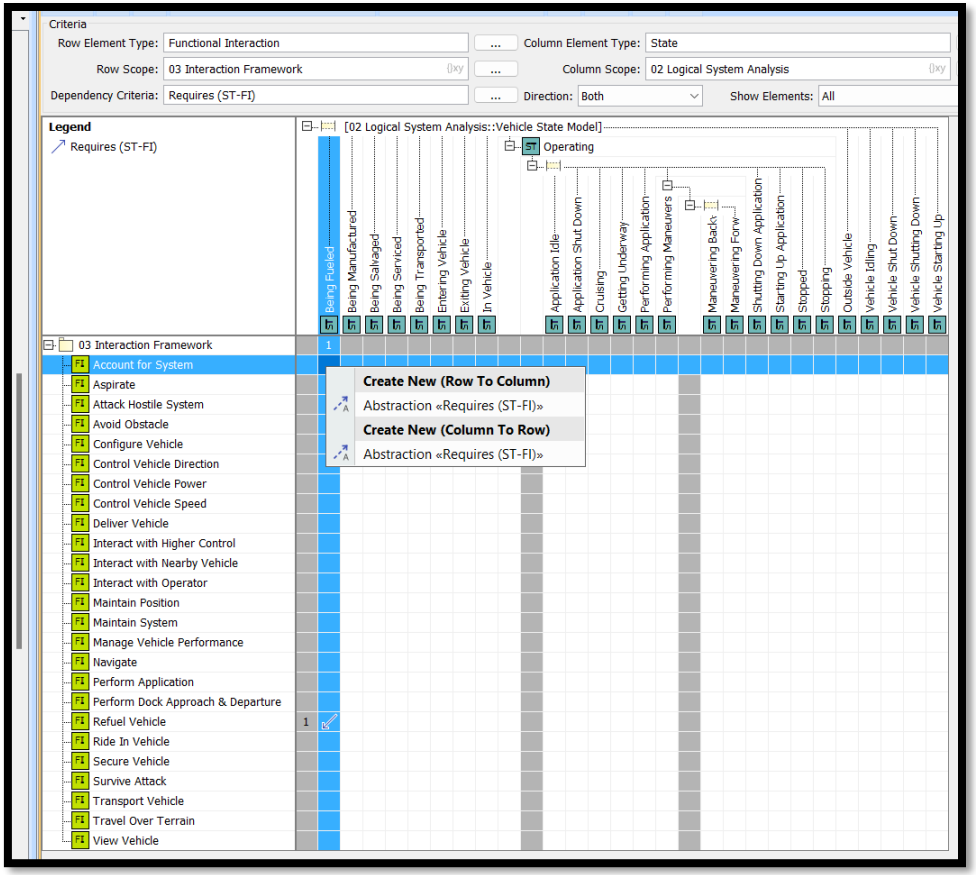
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

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.

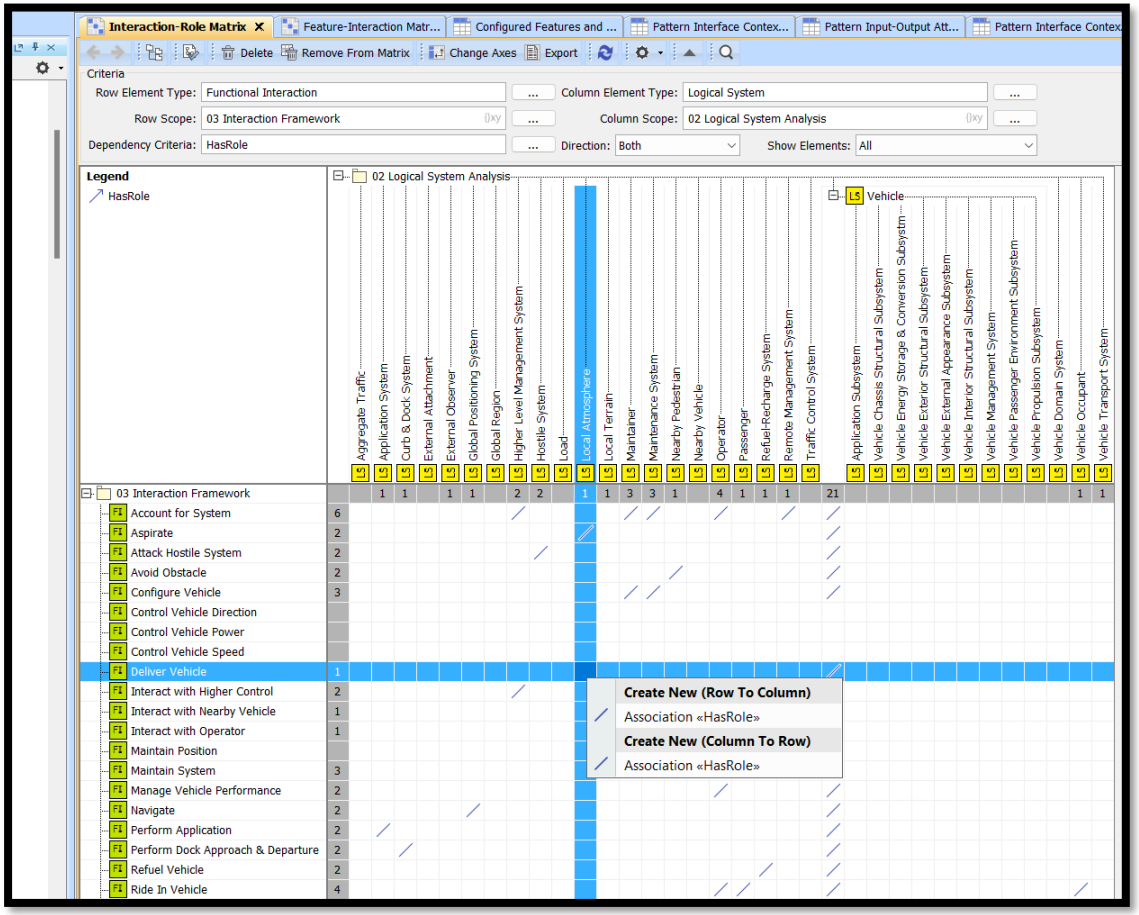
V2021x supports entering all relationship types via a matrix.

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.



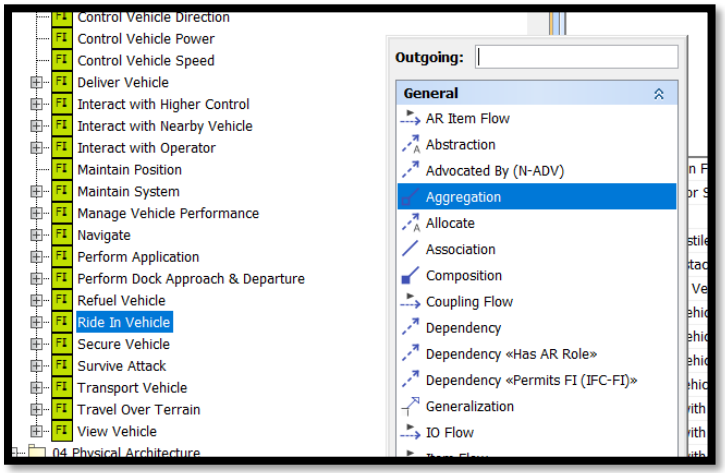
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

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

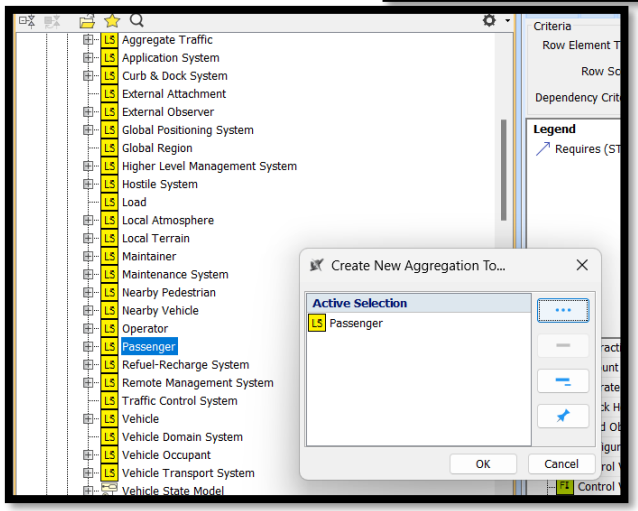
V2021x supports entering all relationship types via a matrix.

Metarelationships: Creating Interaction-Role Relationships

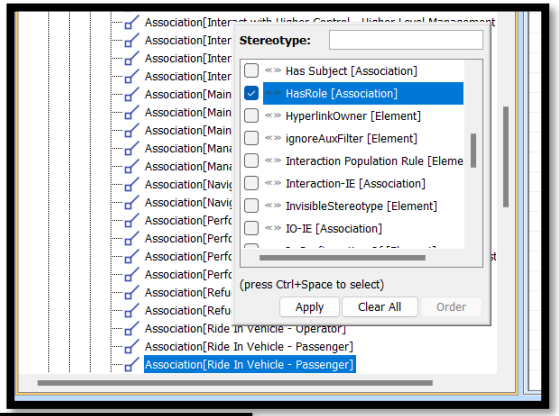
1. Right-Click the Interaction, ~Create Relation, Select Outgoing, Choose Aggregation
2. Select the related Role(s)
3. Right-Click the new relationship, ~Stereotype, Select “Has Role”



1



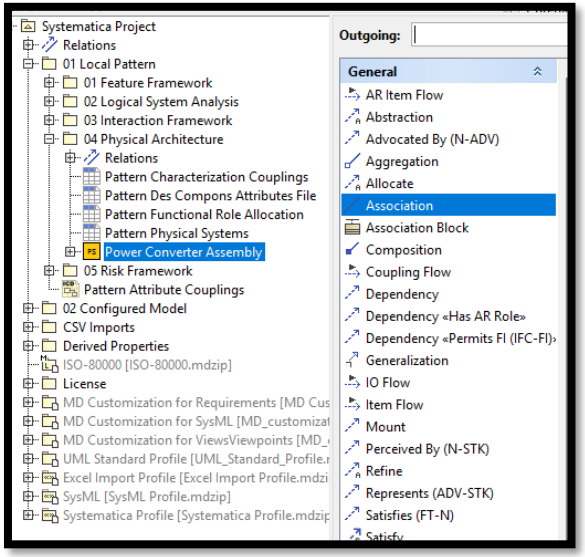
2



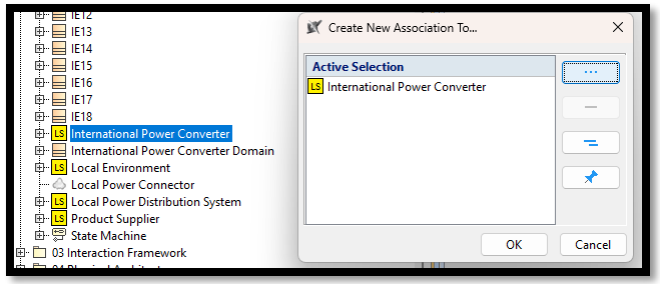
3

Metarelationships:
Creating Functional Role Allocations (Role-Design)

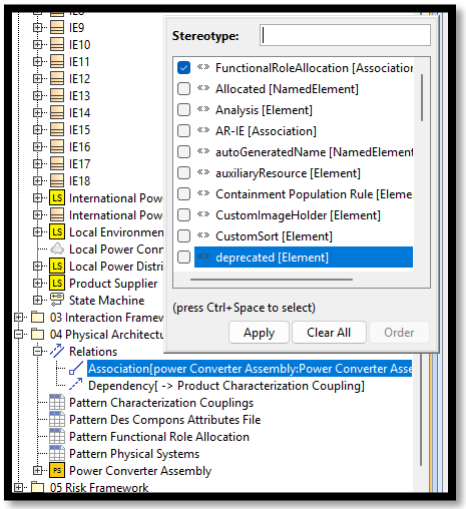
1. Relationships between Design Components (Physical Systems) and Logical Systems are SysML Associations.
2. Create an Association in which the Design Components (Physical System) aggregates the Logical System.
3. Add a “Functional Role Allocation” stereotype to the new Association



1

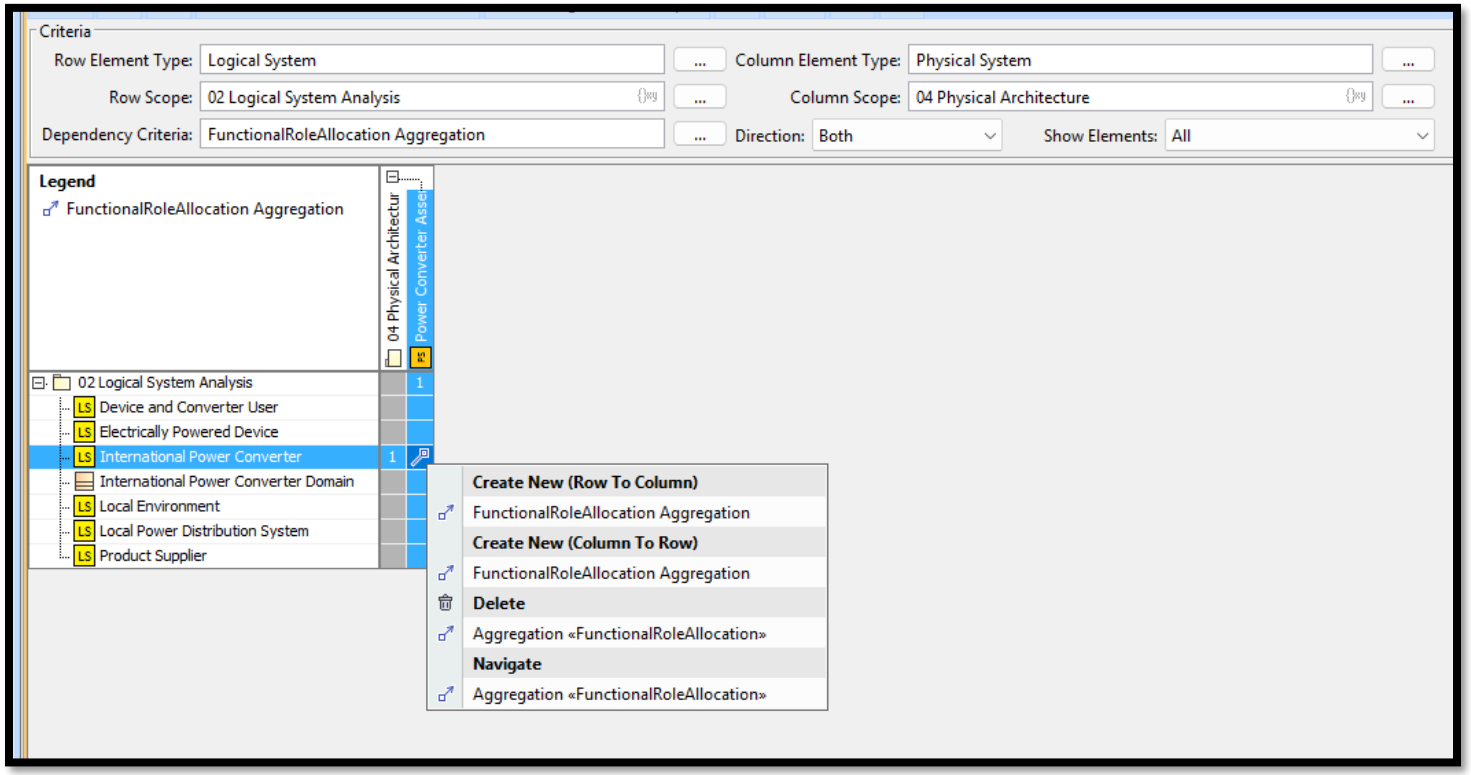


2



3

Metarelationships:
Creating Functional Role Allocations (Role-Design)



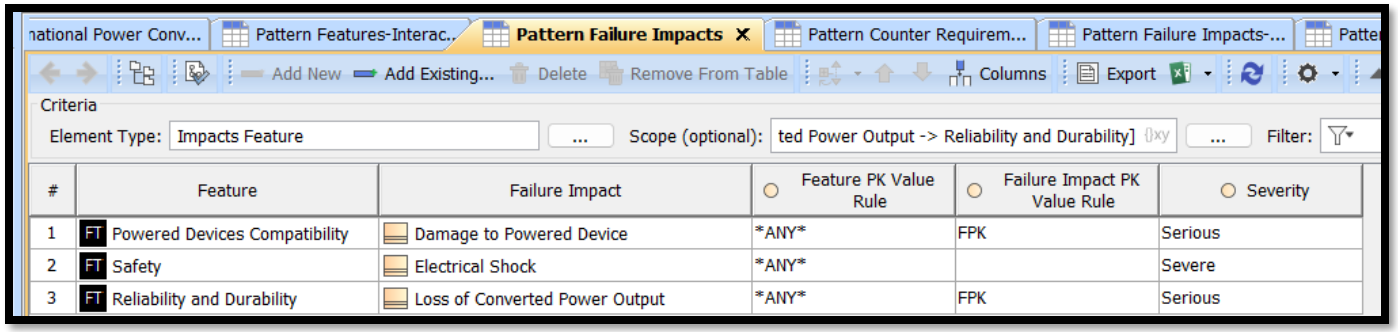
*Note: In Cameo Systems Modeler V19, SysML Aggregation type relationships cannot be entered via matrix, but they can be viewed there.

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

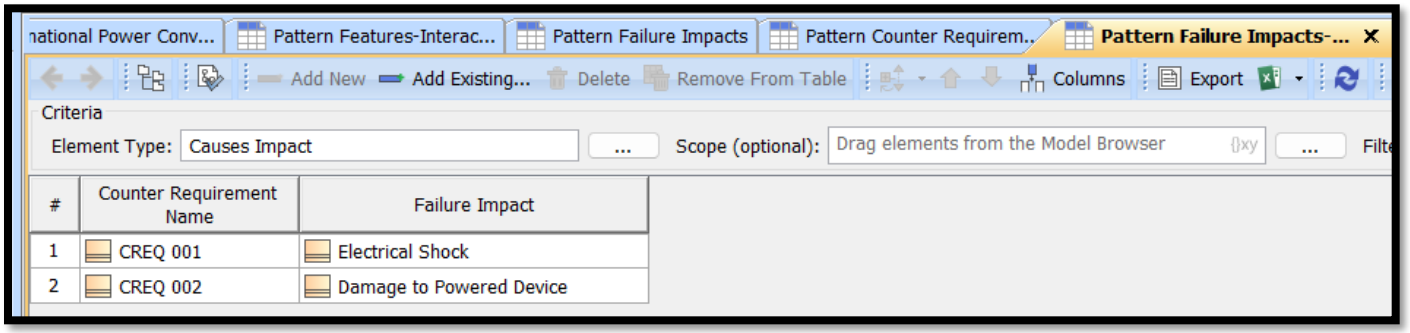


#	Feature	Failure Impact	Feature PK Value Rule	Failure Impact PK Value Rule	Severity
1	FT Powered Devices Compatibility	Damage to Powered Device	*ANY*	FPK	Serious
2	FT Safety	Electrical Shock	*ANY*		Severe
3	FT Reliability and Durability	Loss of Converted Power Output	*ANY*	FPK	Serious

Example Pattern Failure Impacts Table

Metarelationships: Creating Failure Analysis Relationships—Causes Impact

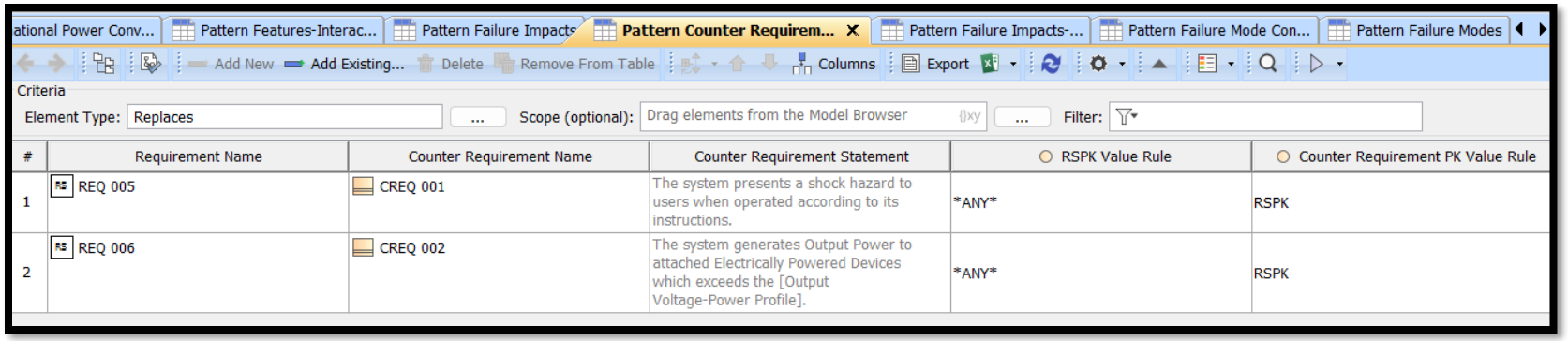
- Create a Dependency relationship with the Counter Requirement as the Source and the Failure Impact as the Target.
- Add a “Causes Impact” stereotype to the new relationship



Example Pattern Failure Impacts-Counter Requirements Table

Metarelationships: Creating Failure Analysis Relationships—Replaces

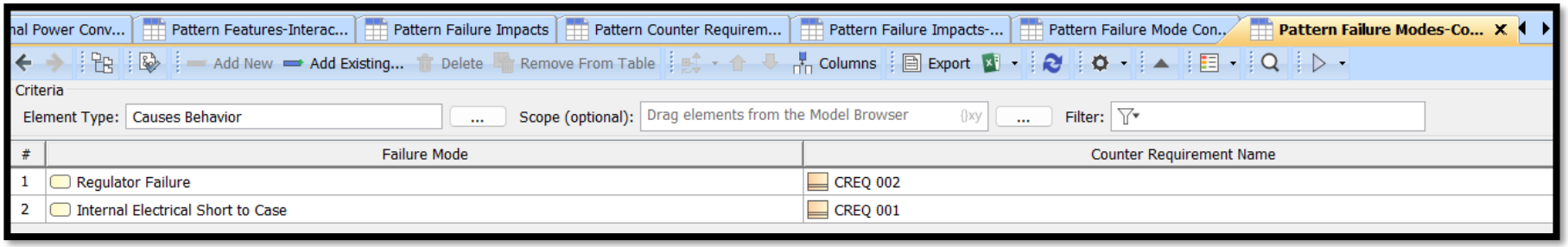
- Create a Dependency relationship with the Counter Requirement as the Source and the Requirement as the Target.
- Add a “Replaces” stereotype to the new relationship.



Example Pattern Counter Requirements Table

Metarelationships: Creating Failure Analysis Relationships—Causes Behavior

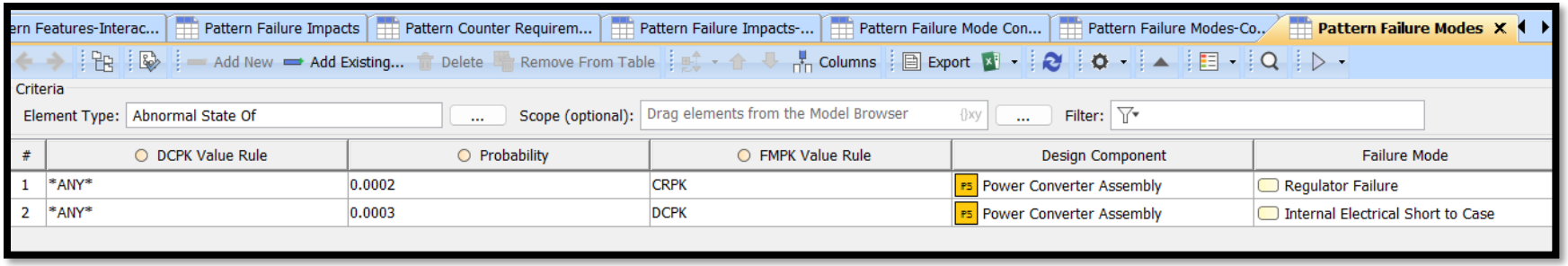
- Create a Dependency relationship with the Failure Mode as the Source and the Counter Requirement as the Target.
- Add a “Causes Behavior” stereotype to the new relationship.



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.

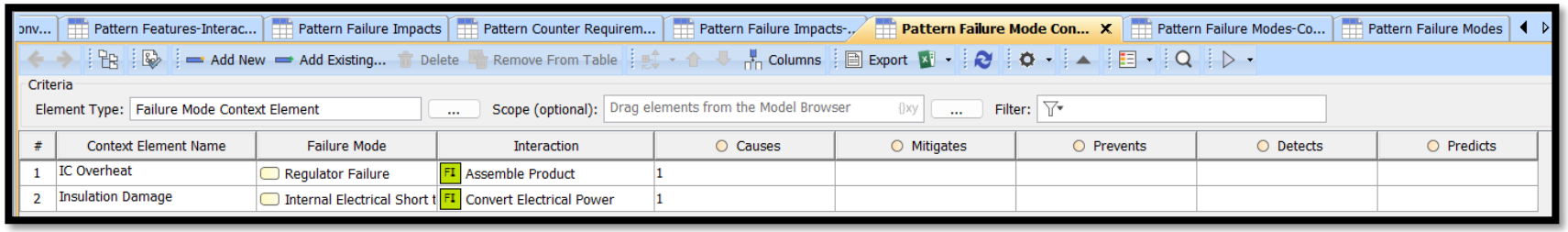


#	DCPK Value Rule	Probability	FMPK Value Rule	Design Component	Failure Mode
1	*ANY*	0.0002	CRPK	Power Converter Assembly	Regulator Failure
2	*ANY*	0.0003	DCPK	Power Converter Assembly	Internal Electrical Short to Case

Example Pattern Failure Modes Table

Metarelationships: Creating Failure Analysis Relationships—Provides Failure Mode Context

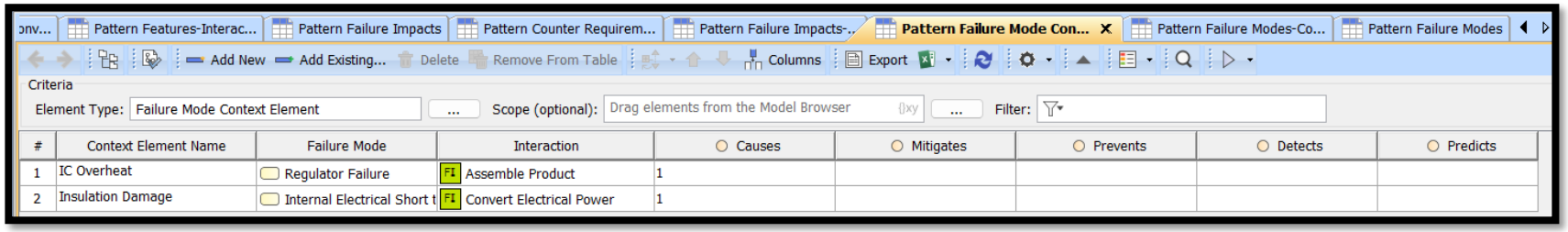
- Create a Dependency relationship with the Failure Mode Context Element as the Source and the Failure Mode as the Target.
- Add a “Provides Failure Mode Context” stereotype to the new relationship.



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.



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	Operating	Convert Electrical Power	*ANY*	International Power Converter	*ANY*	IPK	Simple
2	Being Purchased	Purchase Device		International Power Converter	*ANY*		Simple
3	Being Recycled	Recycle Device		International Power Converter	*ANY*		Simple
4	In Service	Consume Electrical Power		International Power Converter	*ANY*		Simple
5	In Service	Display Status and Information		International Power Converter	*ANY*		Simple
6	In Service	Handle Device		International Power Converter	*ANY*		Simple
7	In Service	Resist Contamination		International Power Converter	*ANY*		Simple
8	In Service	Set Configuration	*ANY*	International Power Converter	*ANY*		Simple
9	In Service	Tolerate Moisture		International Power Converter	*ANY*		Simple
10	Being a Warranty Record	Provide Warranty Data		International Power Converter	*ANY*		Simple
11	Off	Convert Electrical Power	*ANY*	International Power Converter	*ANY*	IPK	Empty
12	Initial IPC	Purchase Device		International Power Converter	*ANY*		Initial
13	Terminal IPC	Recycle Device		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	Off	Operating	Turn On Request	Set Configuration	Simple			IPK	EPK	EPK
2	Off	Being Recycled	Start Retirement	Recycle Device	Join	Wait for Retire			*ANY*	*ANY*
3	Operating	Off	Turn Off Request	Set Configuration	Simple			IPK	EPK	EPK
4	Being Purchased	Being a Warranty Record	Device Purchased	Purchase Device	Fork				*ANY*	*ANY*
5	Being Purchased	Off	Device Purchased	Purchase Device	Fork				*ANY*	*ANY*
6	Being a Warranty Record	Being Recycled	Complete Retirement	Recycle Device	Join	Wait for Retire			*ANY*	*ANY*
7	Initial IPC	Being Purchased	Request Purchase	Purchase Device	Simple				*ANY*	*ANY*
8	Being Recycled	Terminal IPC	Recycling Completed	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	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].

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		

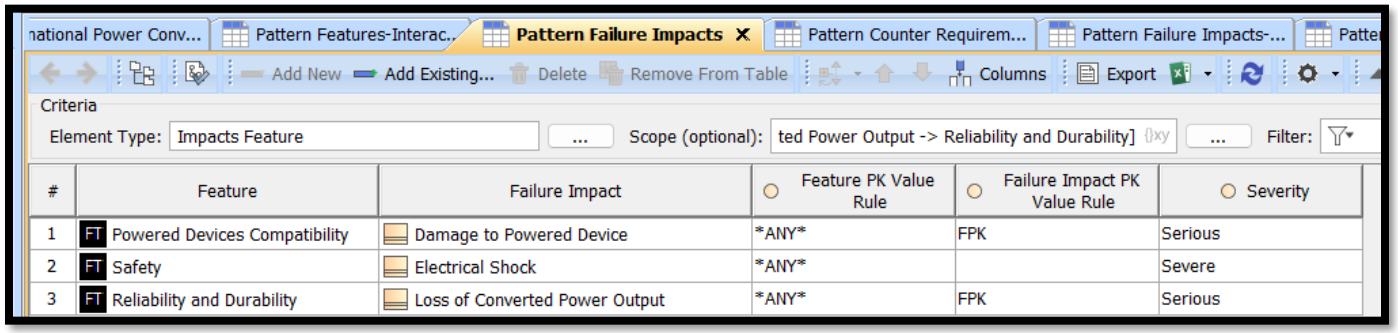
Population Rules: Editing Attribute Coupling Population Rules

- 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	PS Power Converter Assembly	PS Physical System [Class]	Part Number	
2	Max Power Coupling	in	LS International Power Converter	LS Logical System [Class]	Max Power Drain	
3	Max Power Coupling	out	FT Power Mains Compatibility	FT Feature [Class]	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.

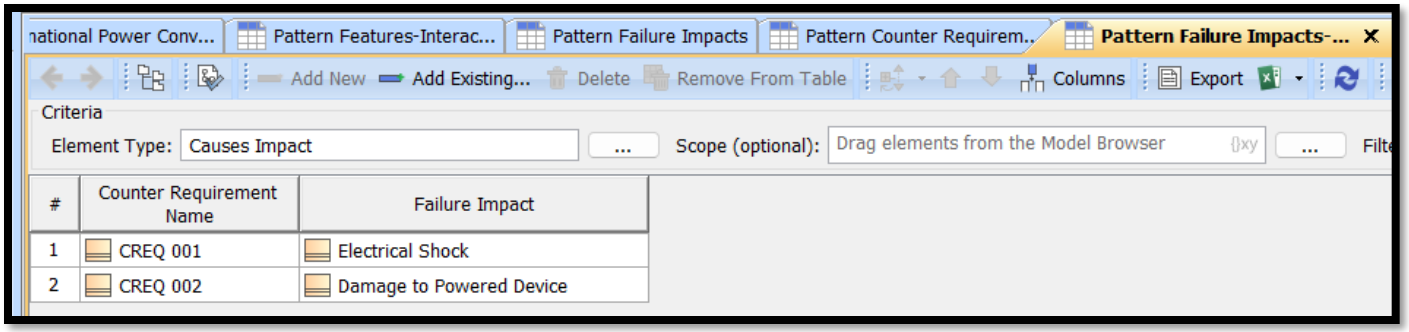


#	Feature	Failure Impact	Feature PK Value Rule	Failure Impact PK Value Rule	Severity
1	FT Powered Devices Compatibility	Damage to Powered Device	*ANY*	FPK	Serious
2	FT Safety	Electrical Shock	*ANY*		Severe
3	FT Reliability and Durability	Loss of Converted Power Output	*ANY*	FPK	Serious

Example Pattern Failure Impacts Table

Population Rules: Editing Causes Impact Relationship Population Rules

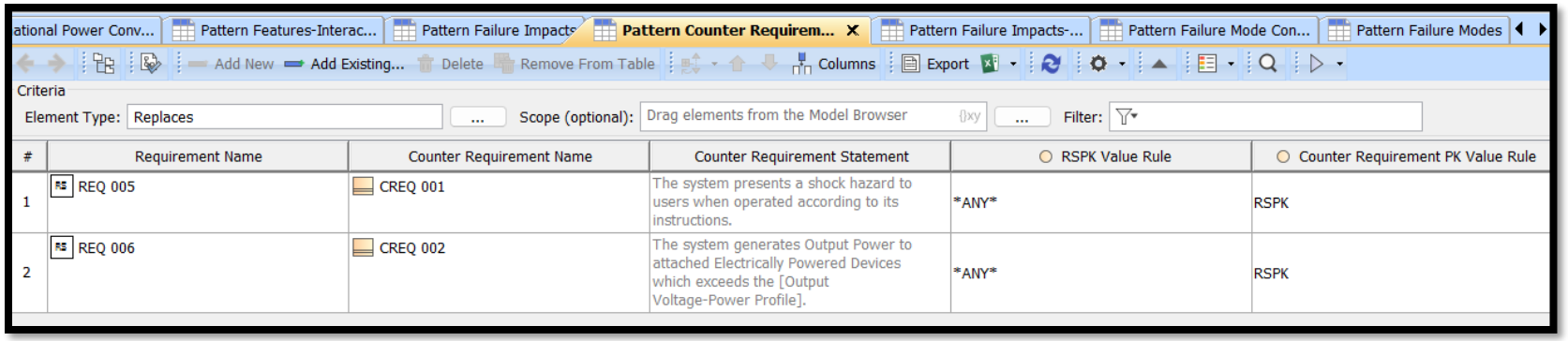
- The “Pattern Failure Impacts-Counter Requirements” table under the Local Pattern package allows the user to edit the Causes Impact Population Rules
- The table governs when and how Causes Impact relationships (associating Counter Requirements with the Failure Impacts they cause) are populated based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



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.

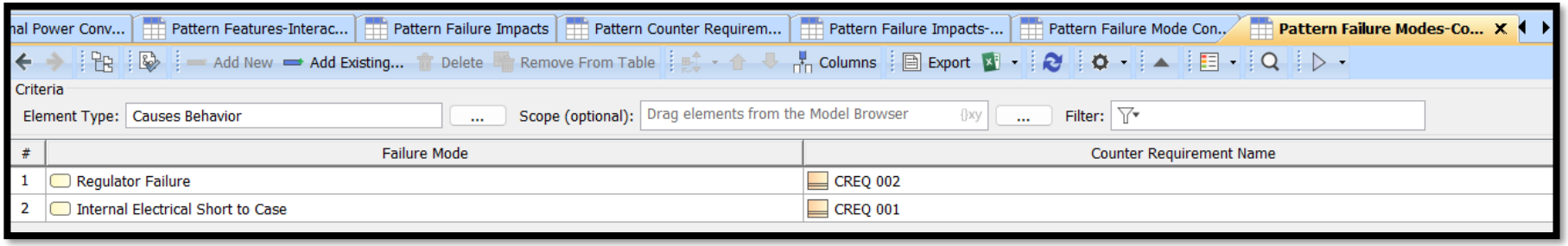


#	Requirement Name	Counter Requirement Name	Counter Requirement Statement	RSPK Value Rule	Counter Requirement PK Value Rule
1	REQ 005	CREQ 001	The system presents a shock hazard to users when operated according to its instructions.	*ANY*	RSPK
2	REQ 006	CREQ 002	The system generates Output Power to attached 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

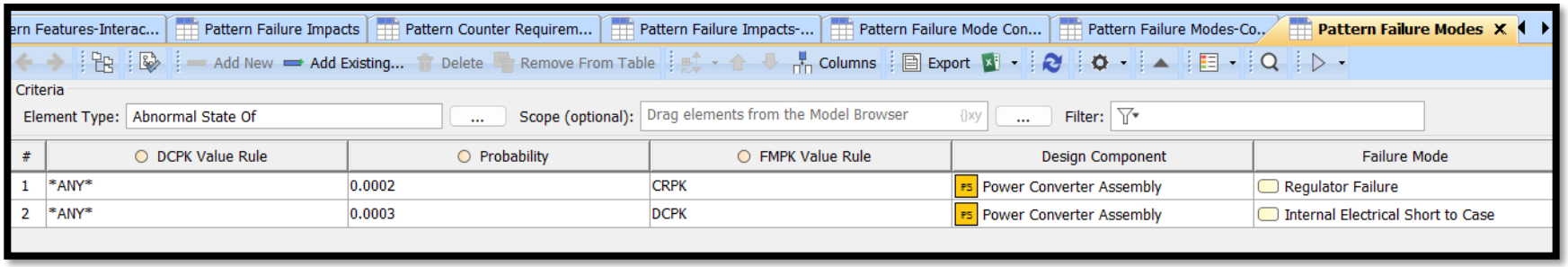
- The “Pattern Failure Modes-Counter Requirements” table under the Local Pattern package allows the user to edit the Causes Behavior Relationship Population Rules
- The table governs when and how Causes Behavior relationships (associating Failure Modes with the Counter Requirements they cause) are populated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



Example Pattern Failure Modes-Counter Requirements Table

Population Rules: Editing Failure Mode Population Rules

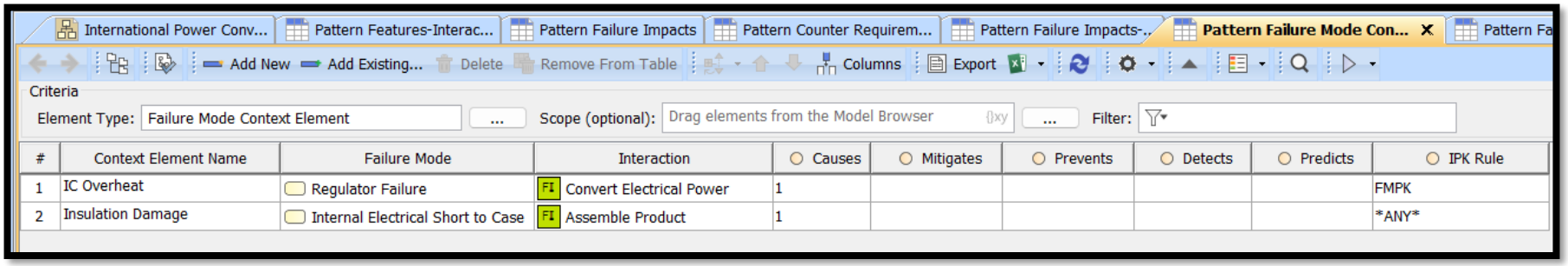
- The “Pattern Failure Modes” table under the Local Pattern package allows the user to edit the Failure Mode Population Rules
- The values in the PK column govern when and how Failure Modes are populated based on which Design Components have been populated.
- The value in the FMPK Value Rule column may 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.



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.



#	Context Element Name	Failure Mode	Interaction	Causes	Mitigates	Prevents	Detects	Predicts	IPK Rule
1	IC Overheat	Regulator Failure	Convert Electrical Power	1					FMPK
2	Insulation Damage	Internal Electrical Short to Case	Assemble Product	1					*ANY*

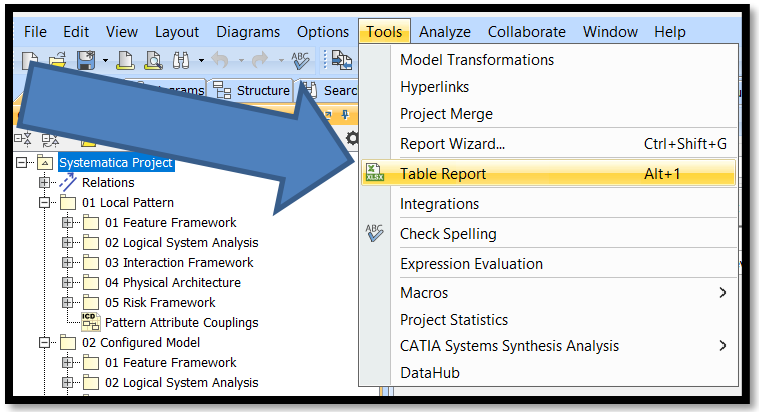
Example Pattern Failure Mode Context Table

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, (2) Importing Configured Model data from the Pattern Configuration Wizard into Cameo Systems Modeler from multiple CSV files and (3) defining the imports (when necessary) and updating the file path references. Refer to the diagram on Slide 8 and the Configuration Wizard Guide.

Exporting Pattern Tables Using the Report Wizard

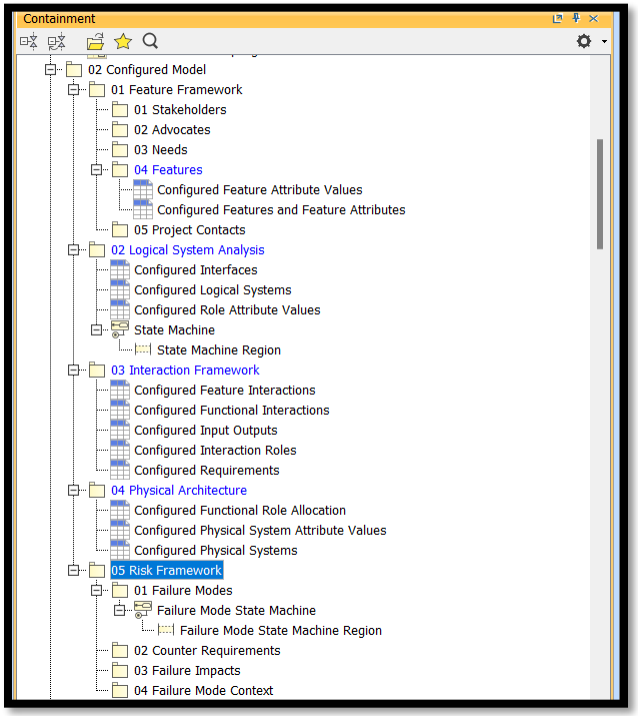
After defining the Export Tables Report as shown in Section 2, it can be invoked by selecting it from the Tools Menu.



If the Report does not work correctly, refer to Section 2 to define it again.

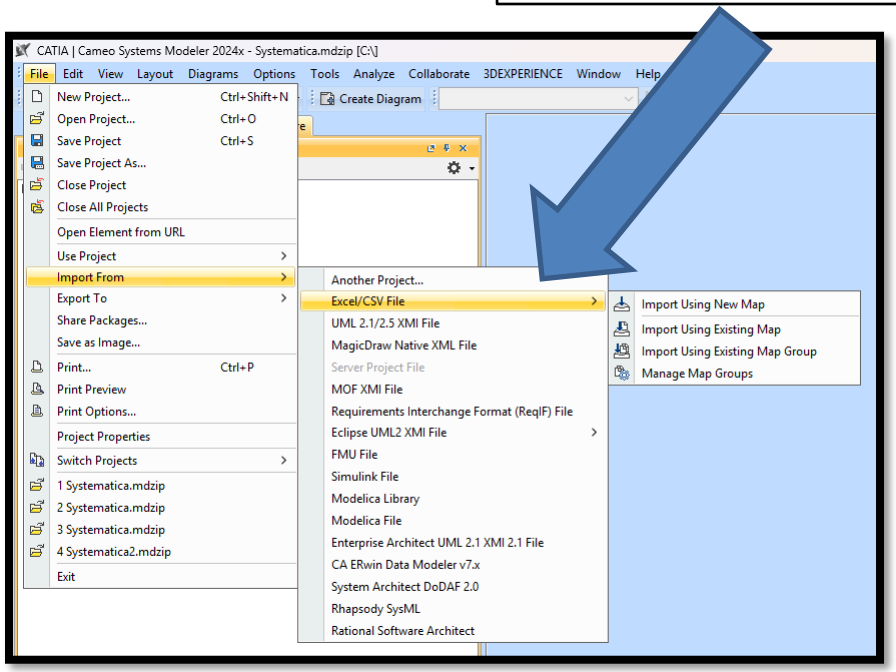
Emptying the Configured Model Packages Before the Import

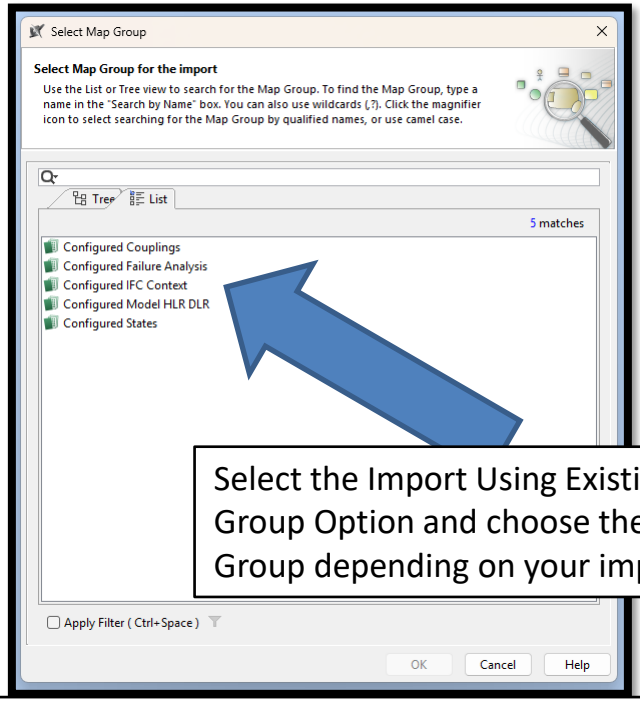
- Prior to importing a configured model into the repository, any previous model content should be removed from there, leaving the basic framework elements in place. Once it is emptied of previous model data, the empty 02 Configured Model package of the Containment Browser should look like the example 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 From > Excel/CSV File option.

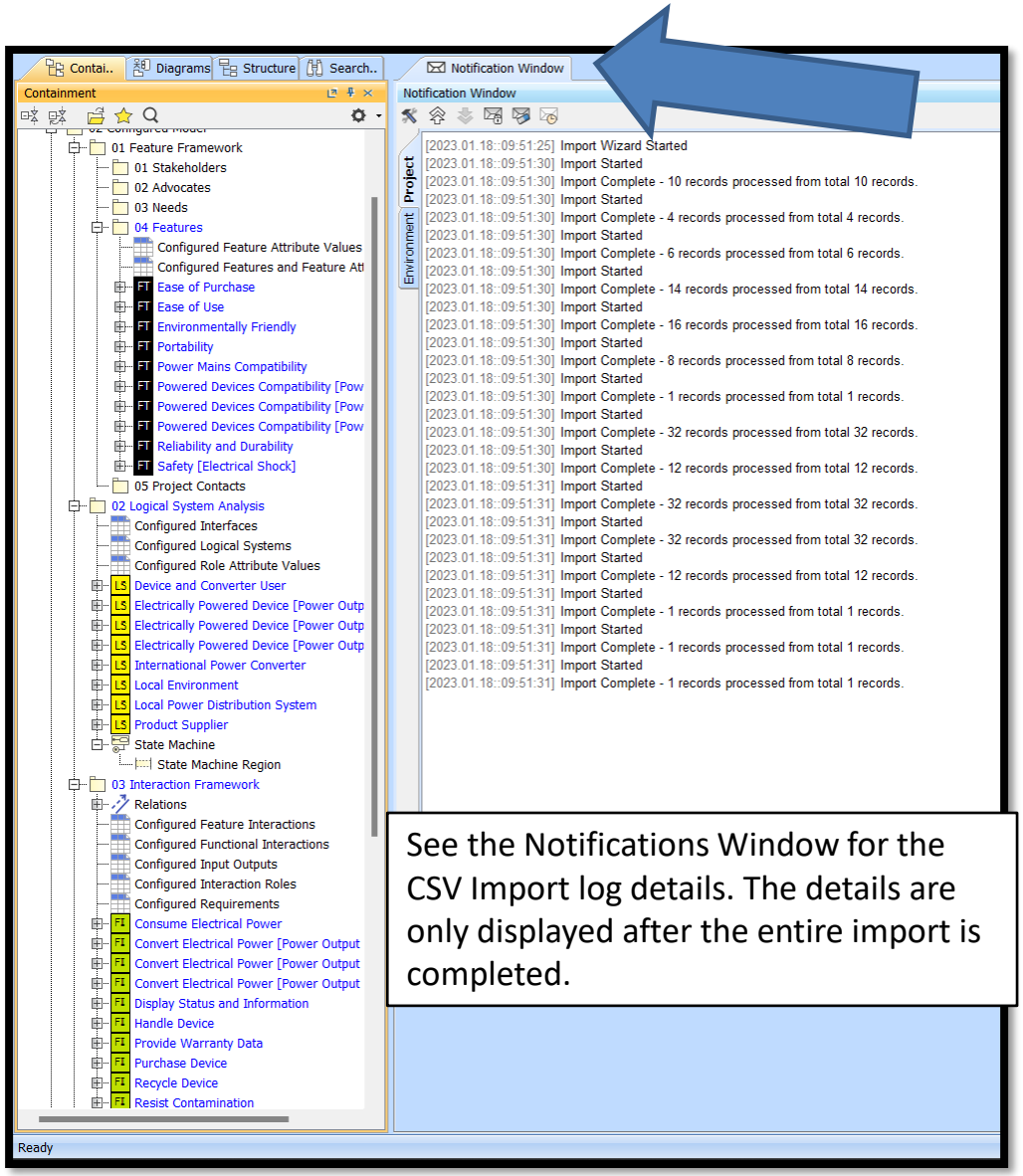




Select the Import Using Existing Map Group Option and choose the Map Group depending on your import needs.

-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 Wizard Guide)

Importing the Configured Model Data

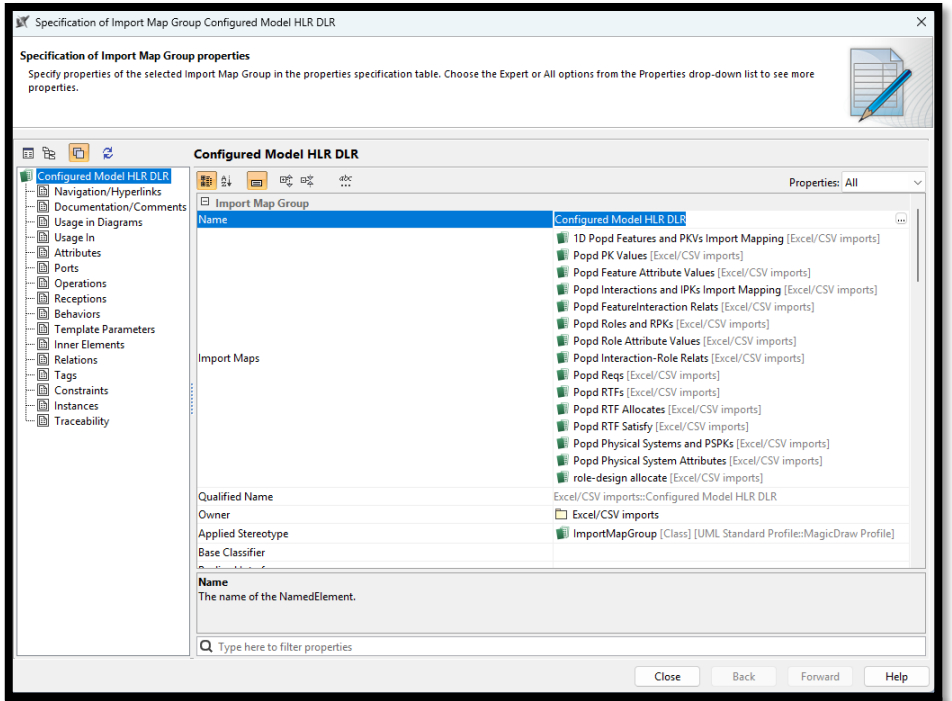


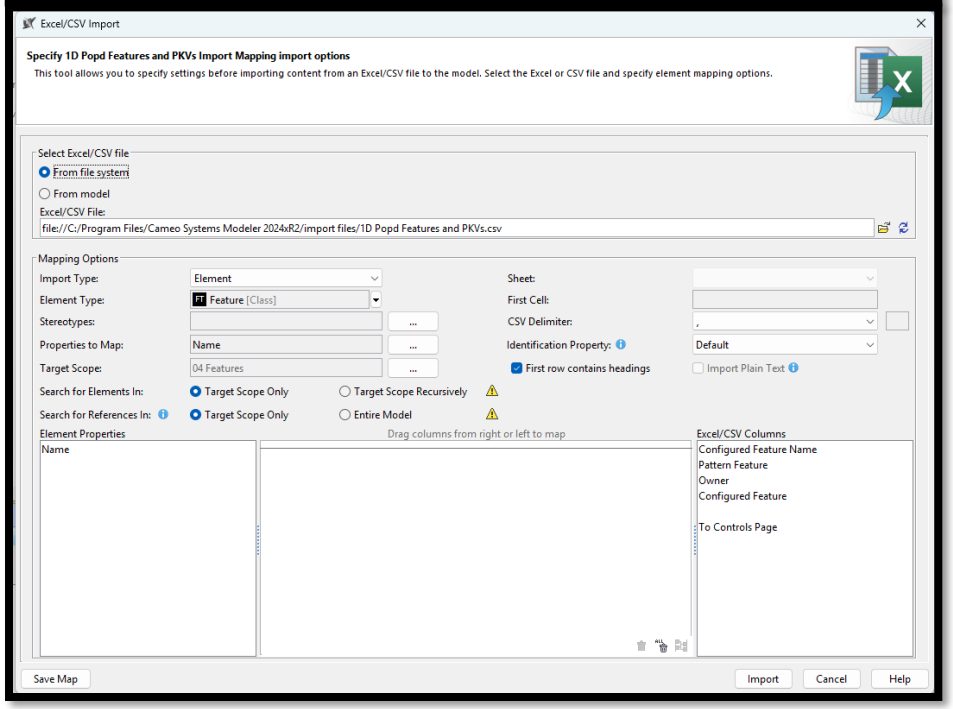
See the Notifications Window for the CSV Import log details. The details are only displayed after the entire import is completed.

7. Pre-defined Cameo Model CSV Import Mapping Definitions

The following import mapping definitions are predefined and for administration of file path names (typically once at installation time) and details of import mappings (very rarely), but not necessary for performing an import. Refer to the diagram on Slide 8 and the Configuration Wizard Guide.

Map Group: Configured Model HLR DLR

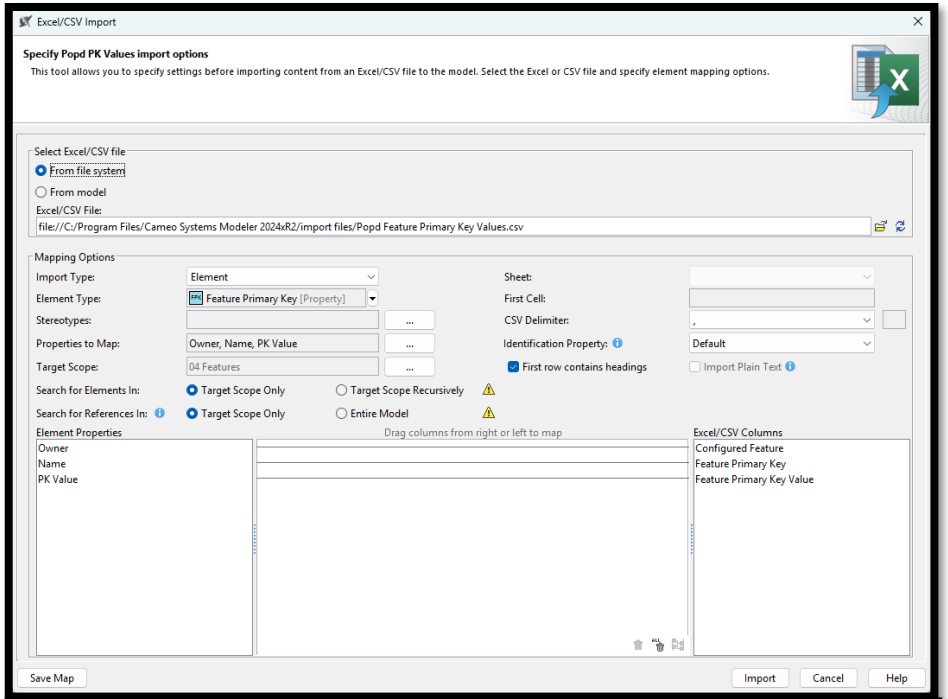




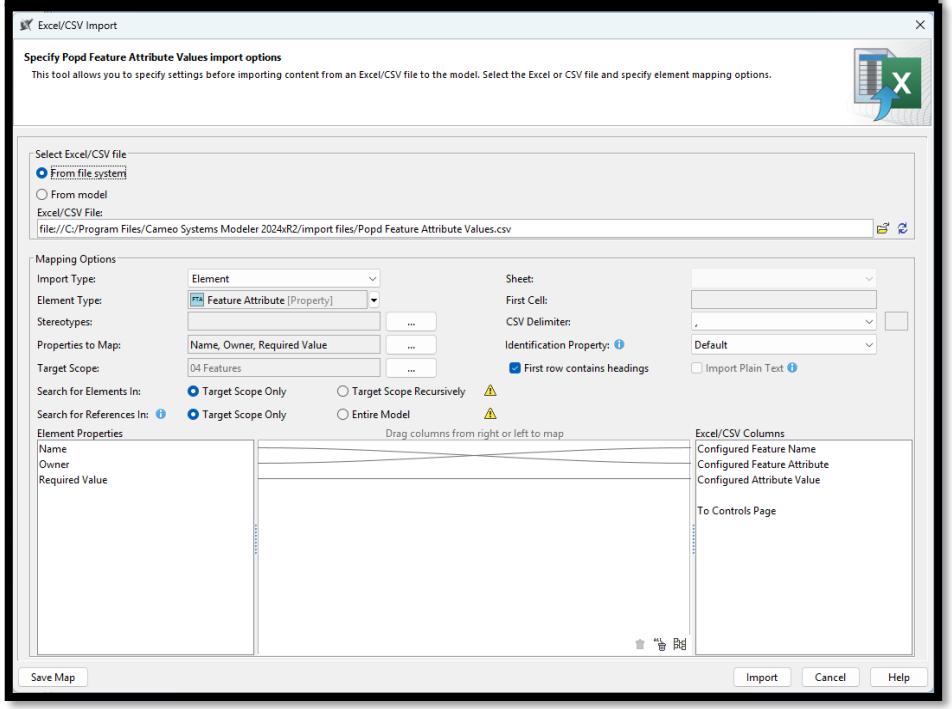
1D Popd Features and PKVs

***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd PK Values

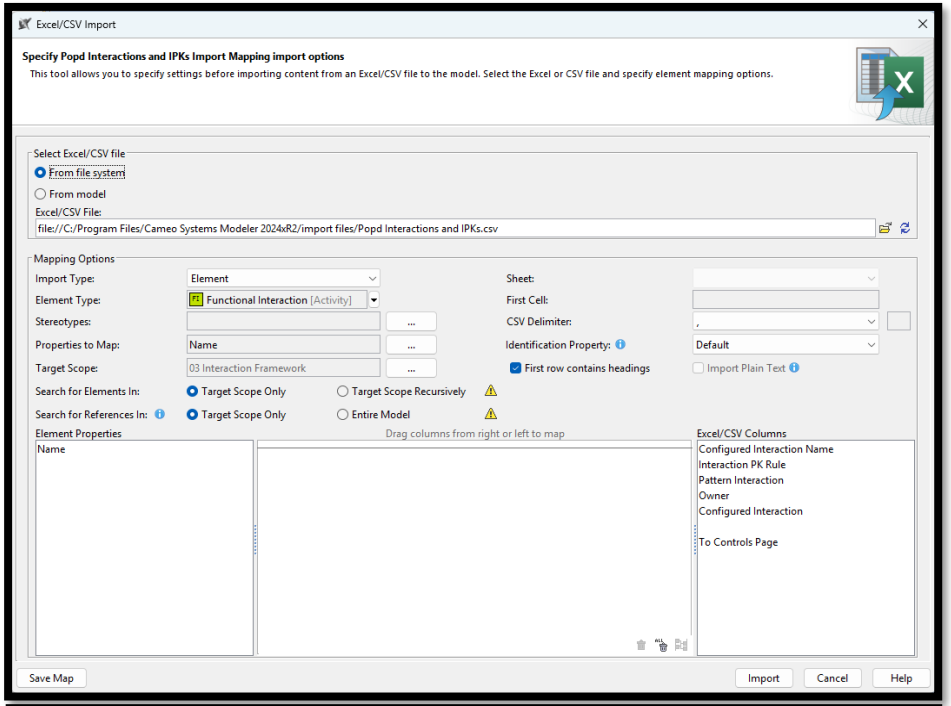


Popd Feature Attribute Values

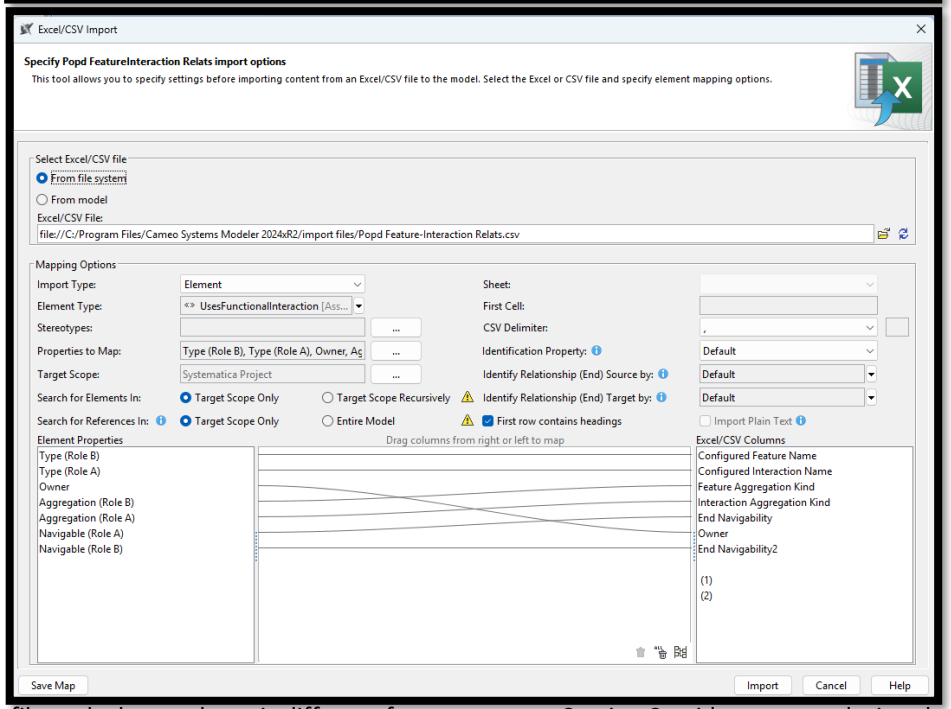


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Interactions and IPKs Import Mapping

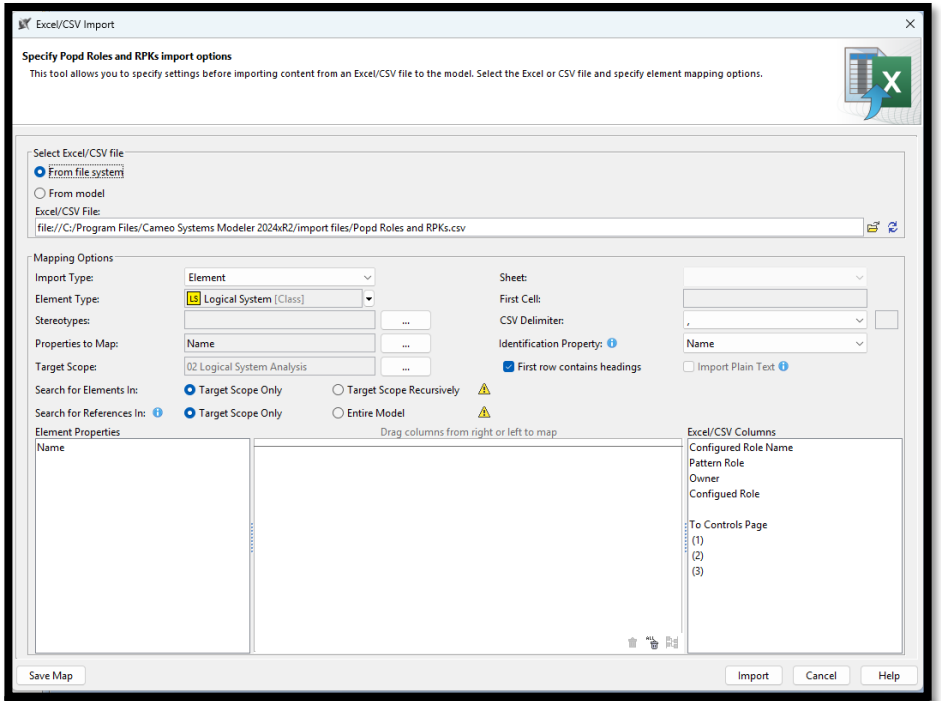


Popd FeatureInteraction Relats

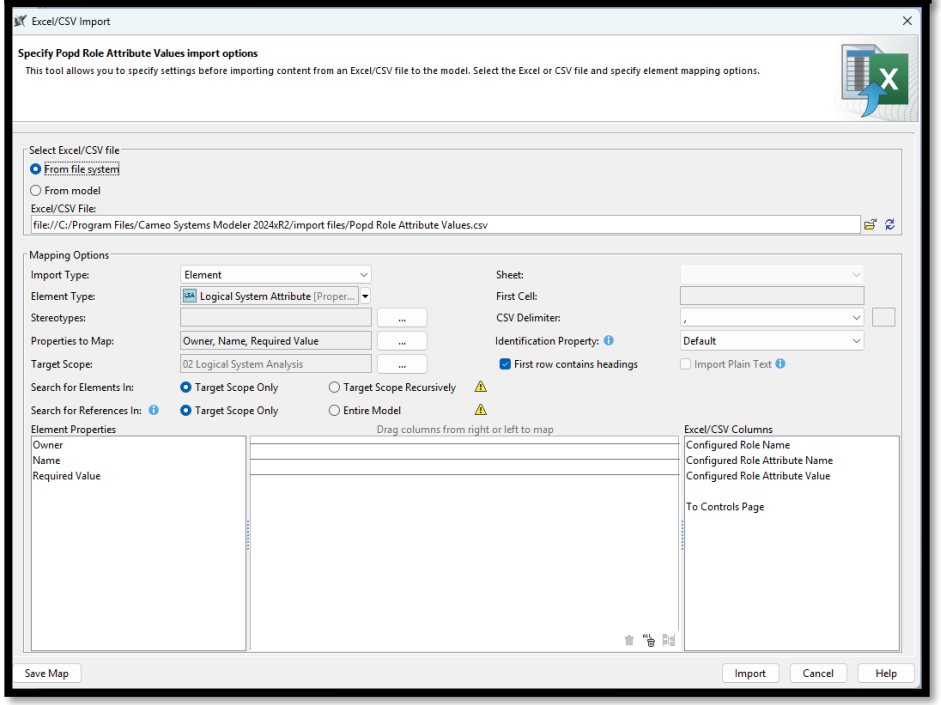


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Roles and RPKs

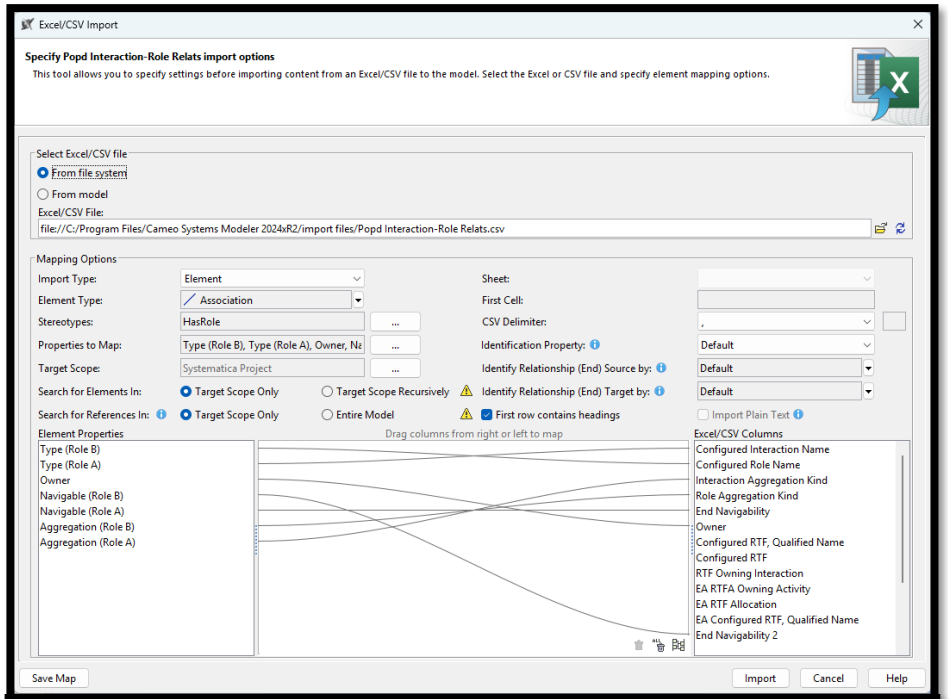


Popd Role Attribute Values

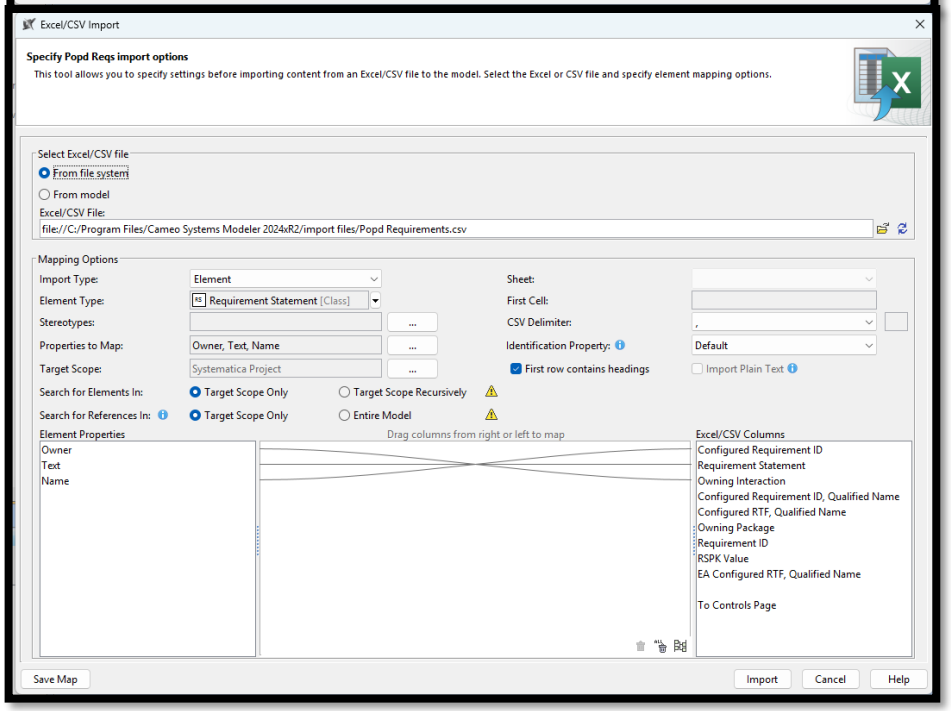


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Interaction-Role Relats

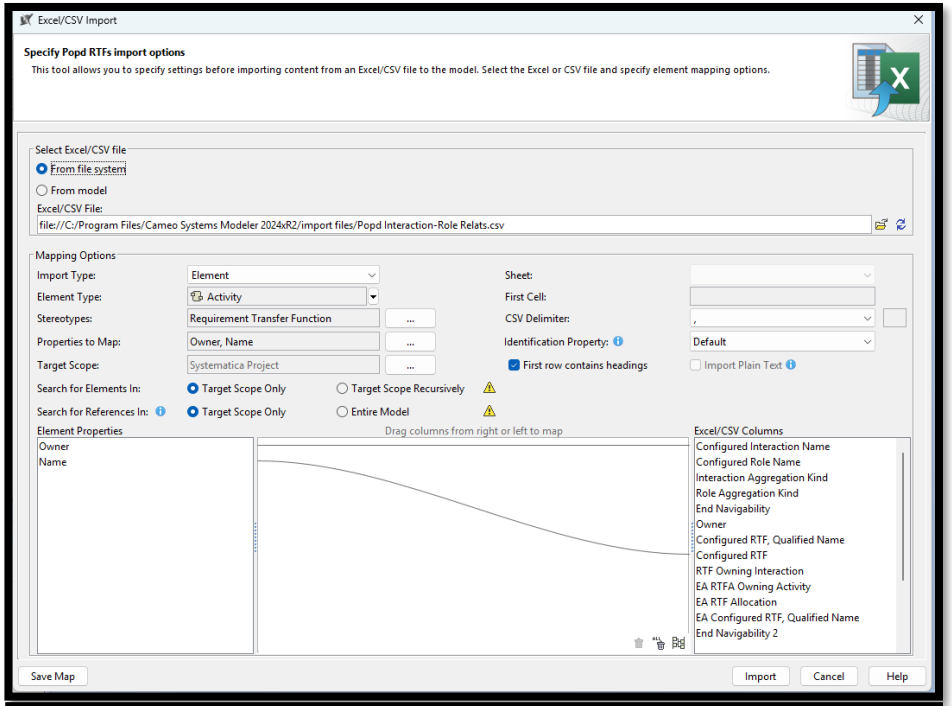


Popd Reqs

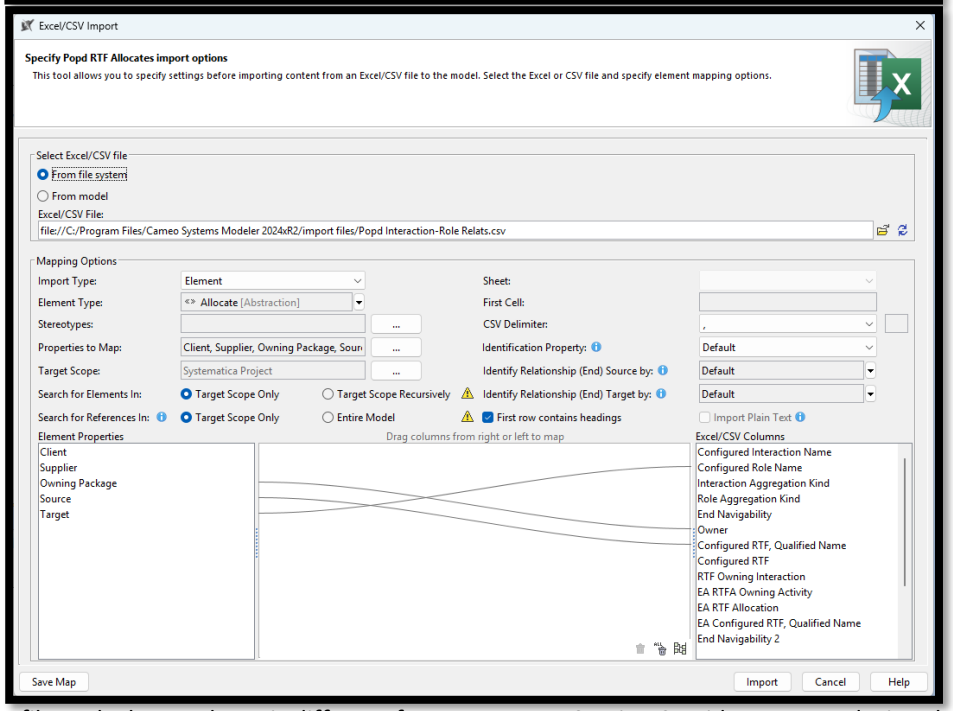


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd RTFs

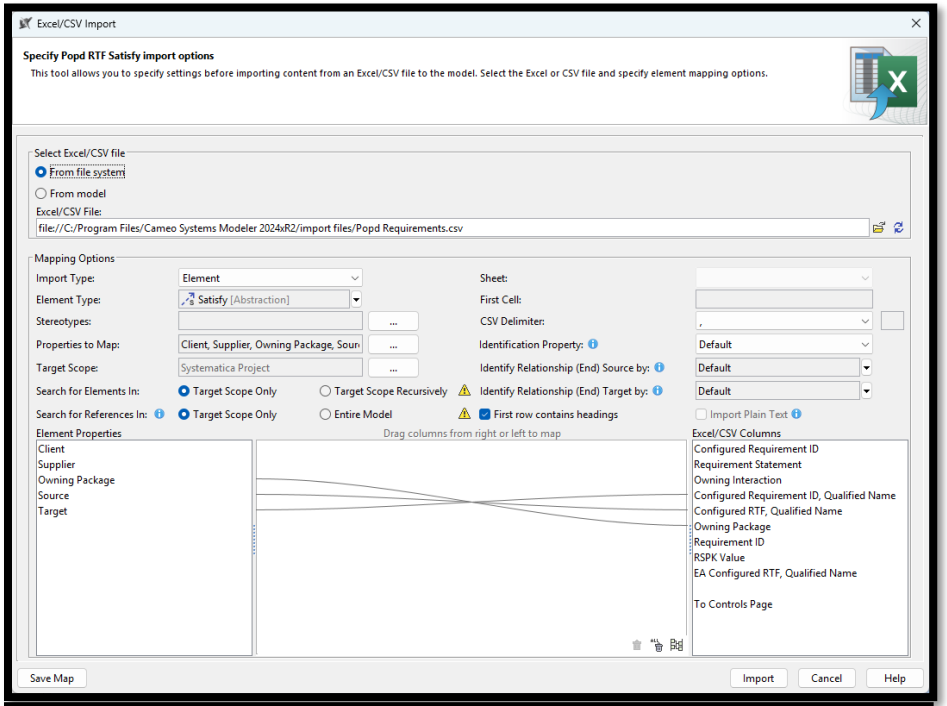


Popd RTF Allocates

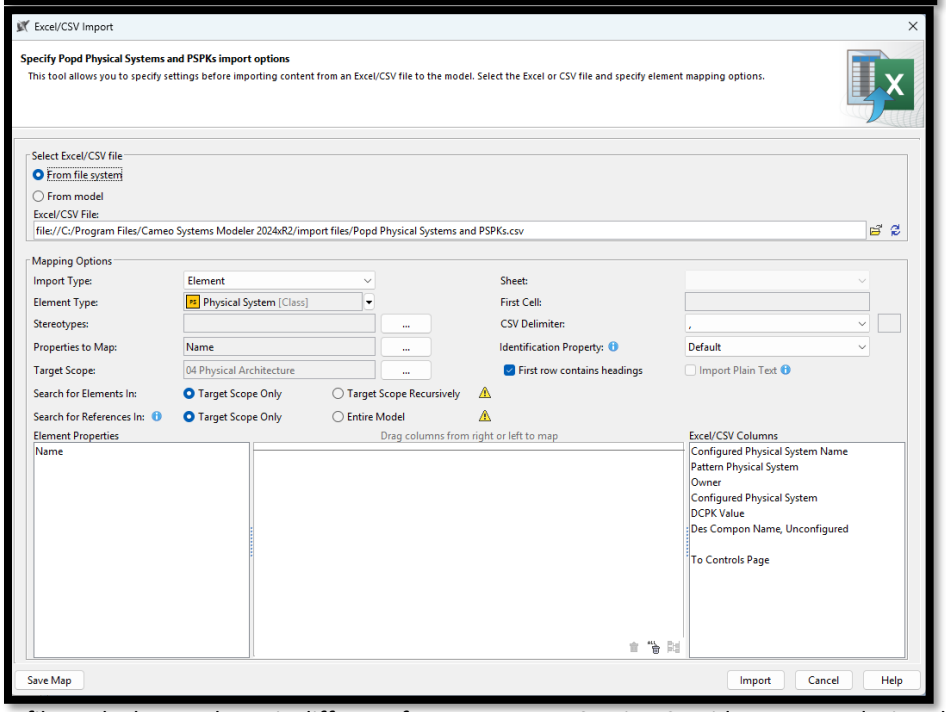


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd RTF Satisfy

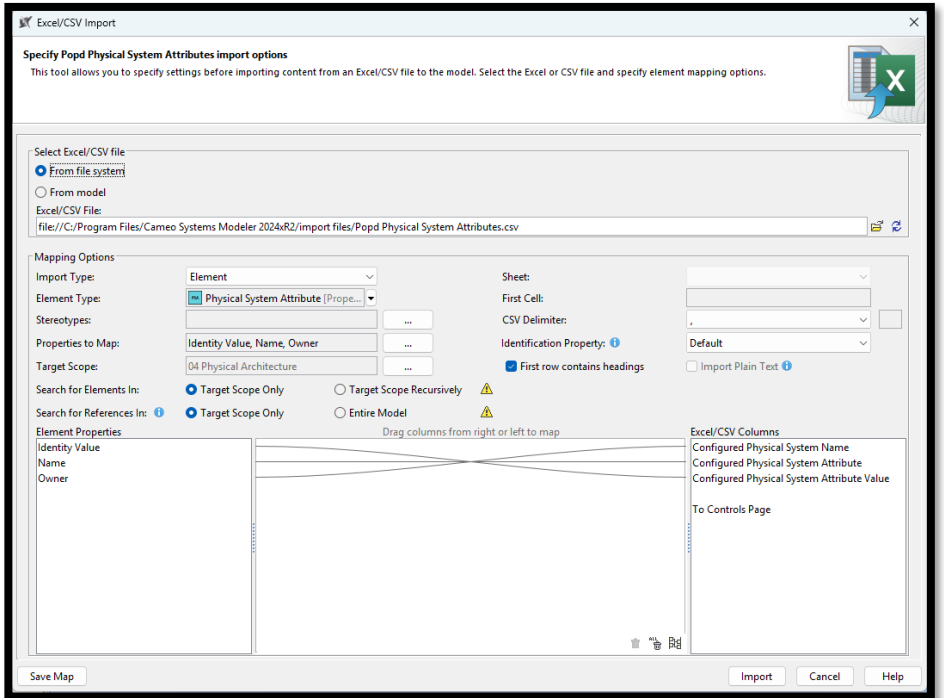


Popd Physical Systems and PSPKs

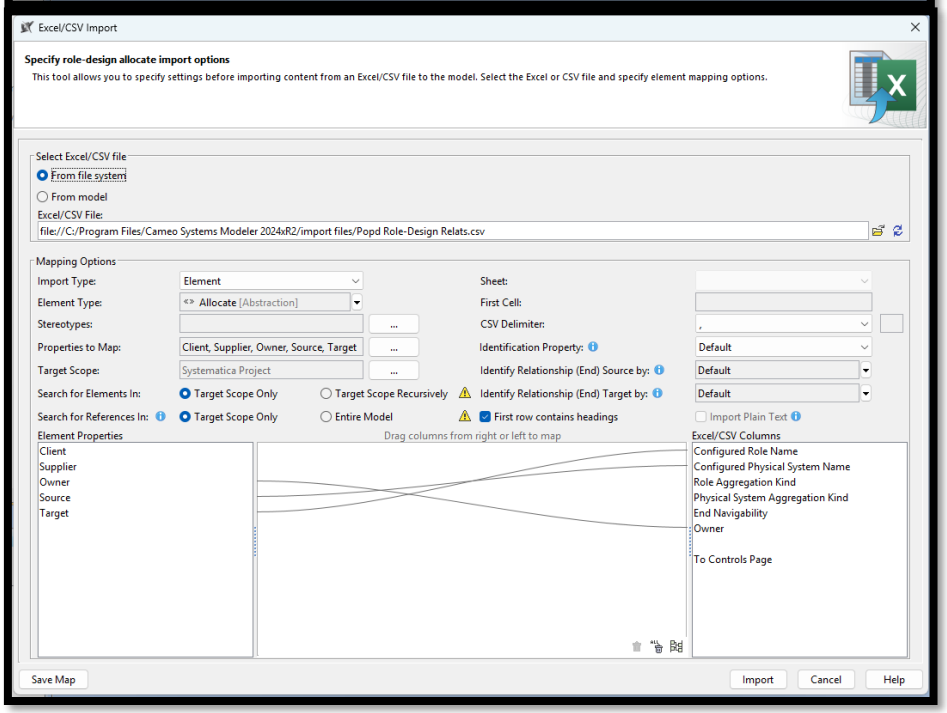


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Physical System Attributes

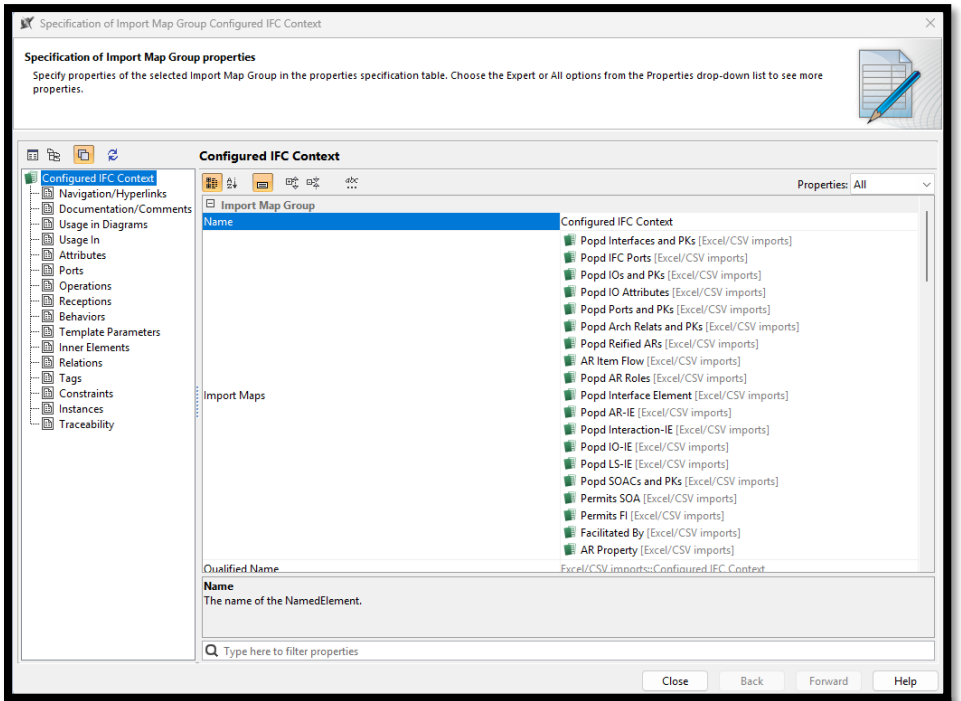


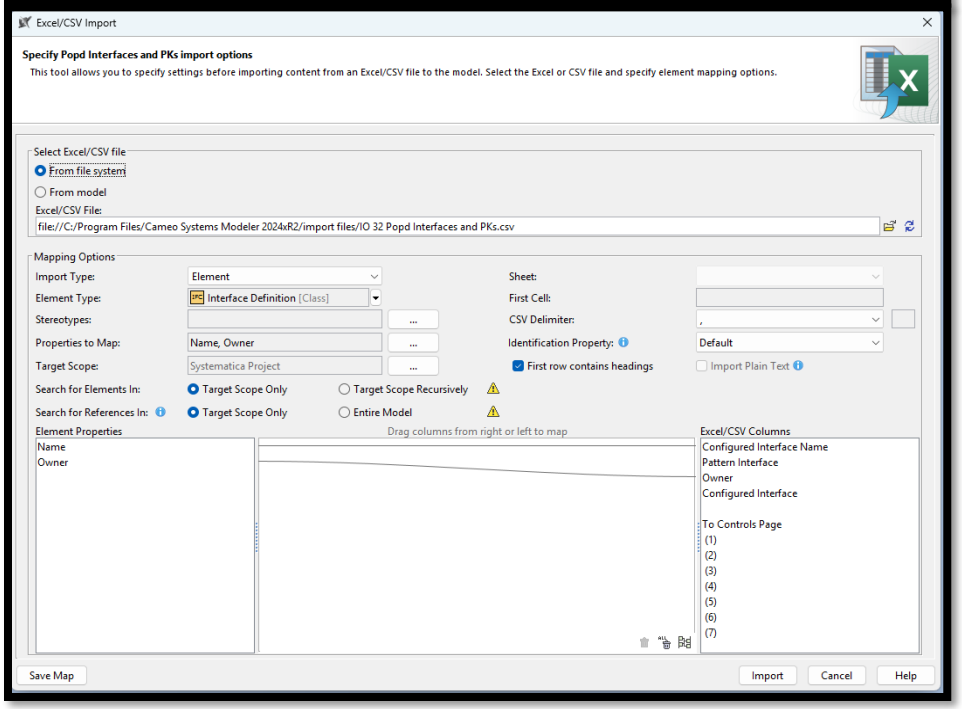
role-design allocate



***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Map Group: Configured IFC Context

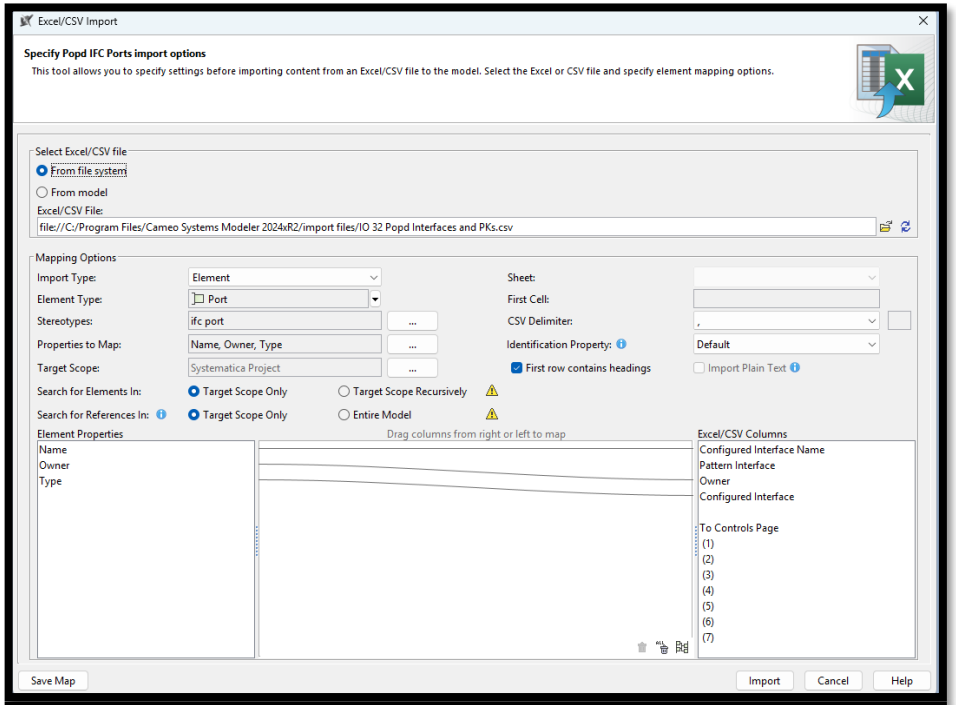




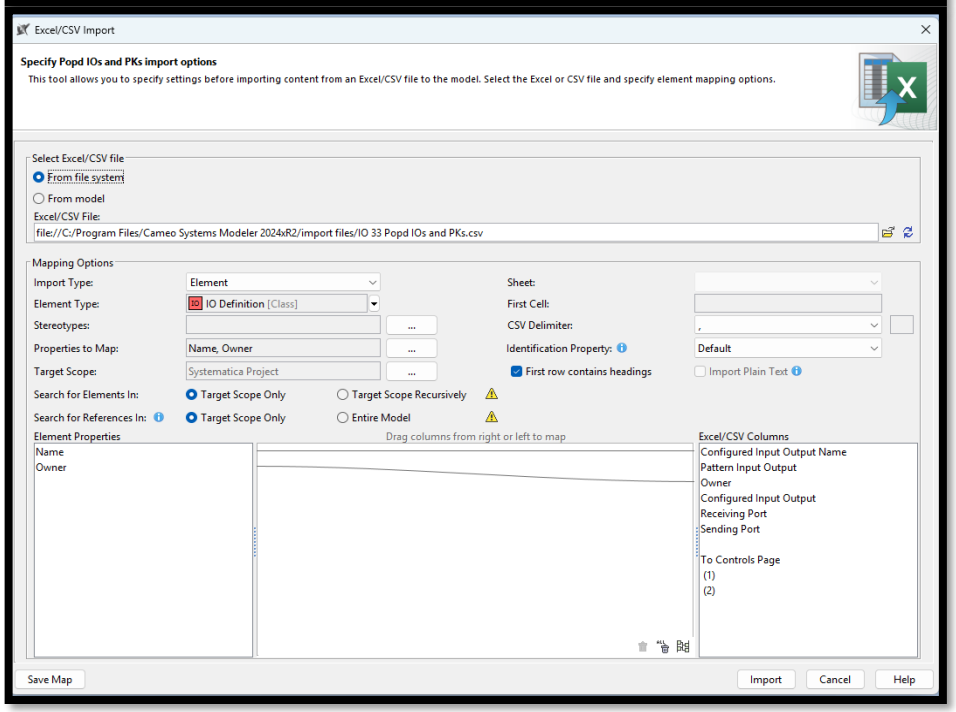
Popd Interfaces and PKs

***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd IFC Ports

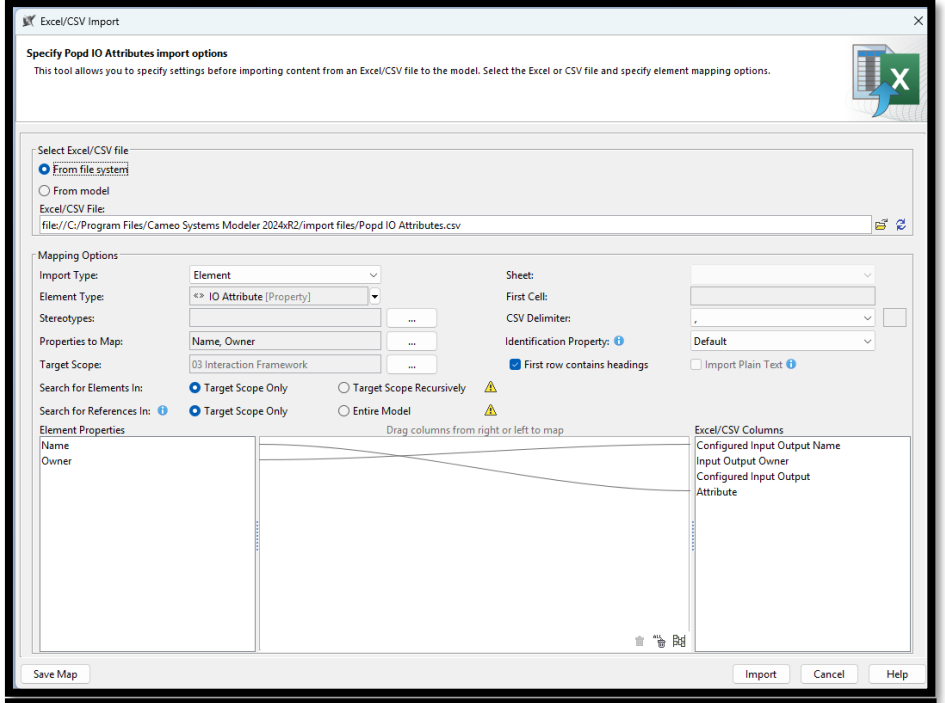


Popd IOs and PKs

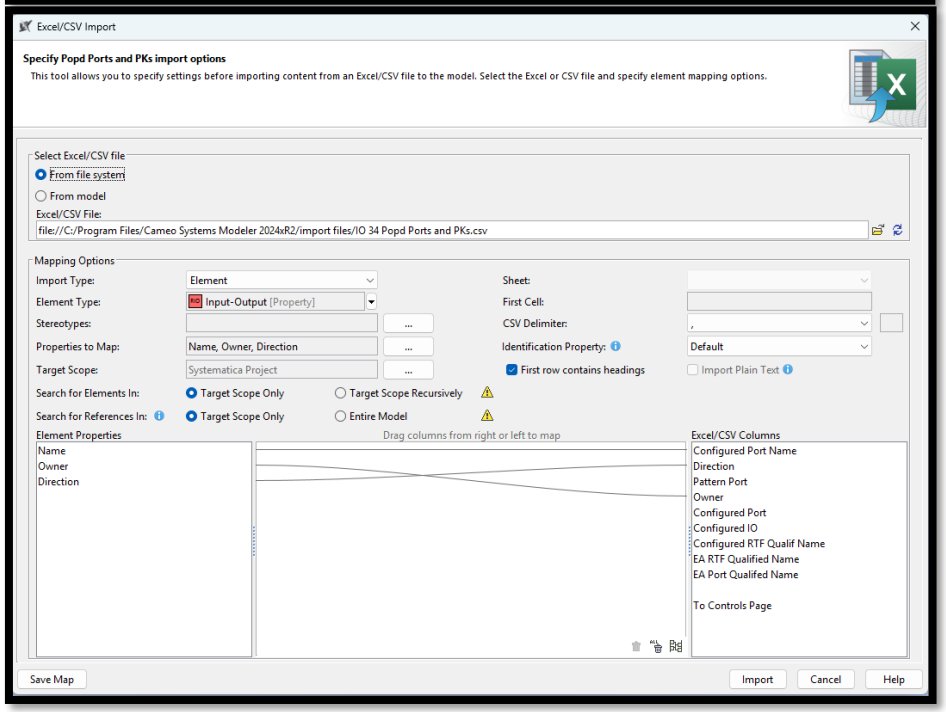


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd IO Attributes

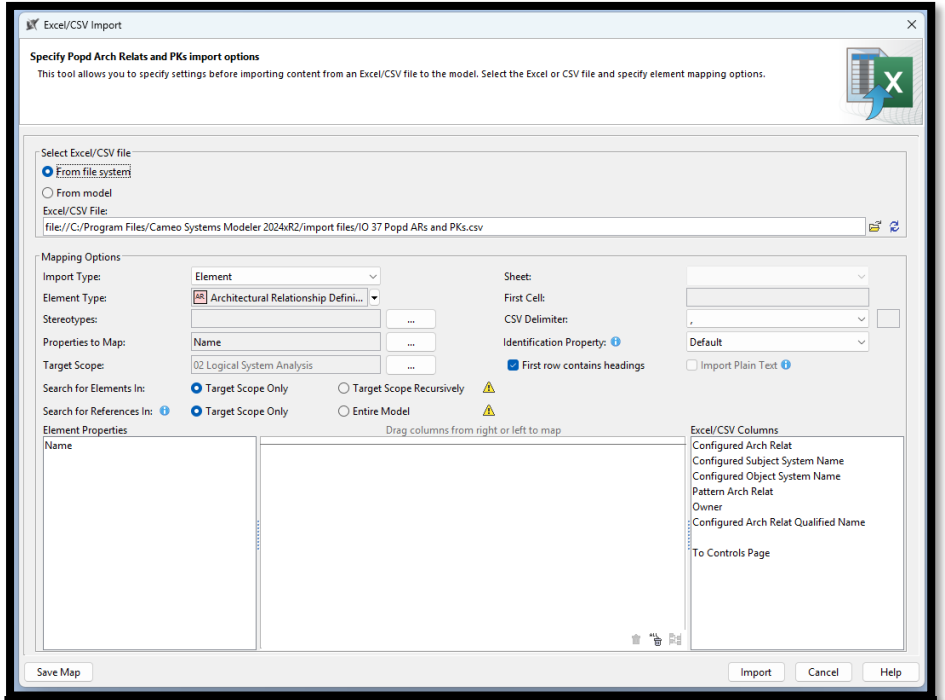


Popd Ports and PKs

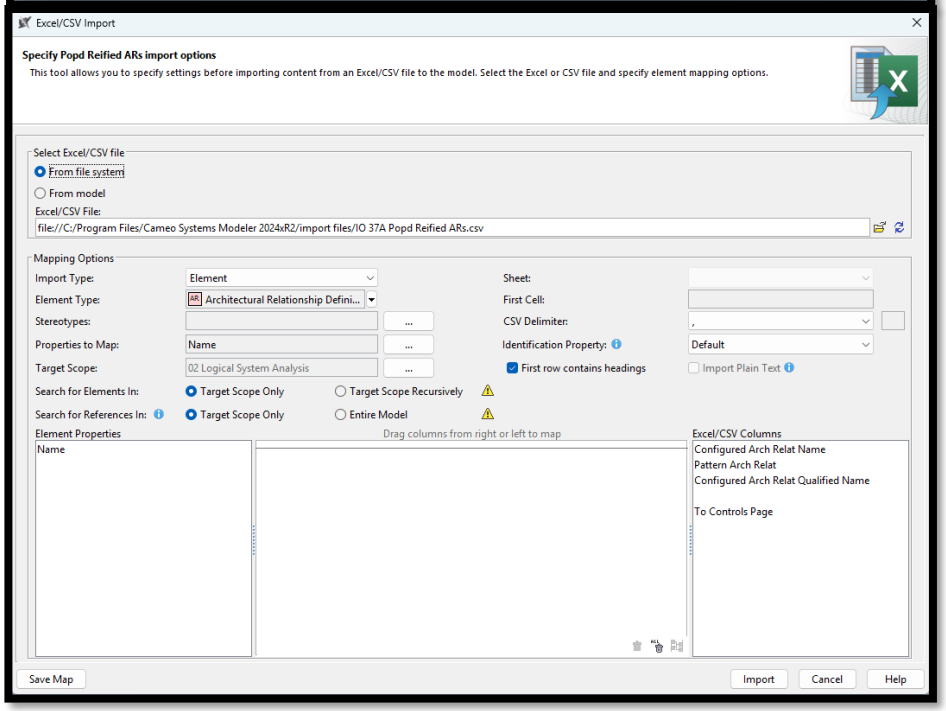


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Arch Relats and PKs

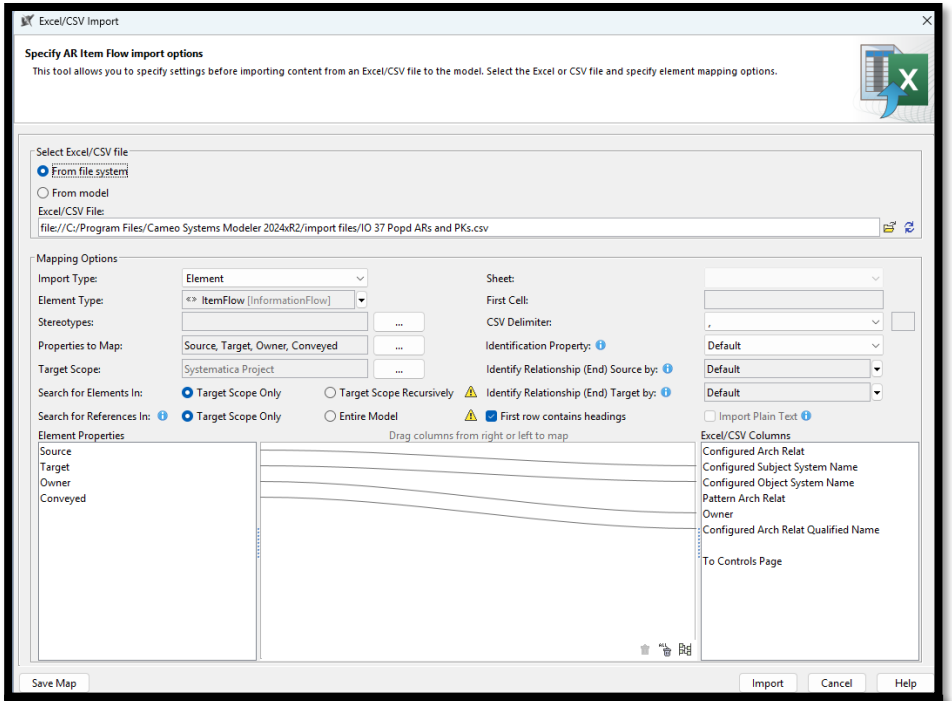


Popd Reified ARs

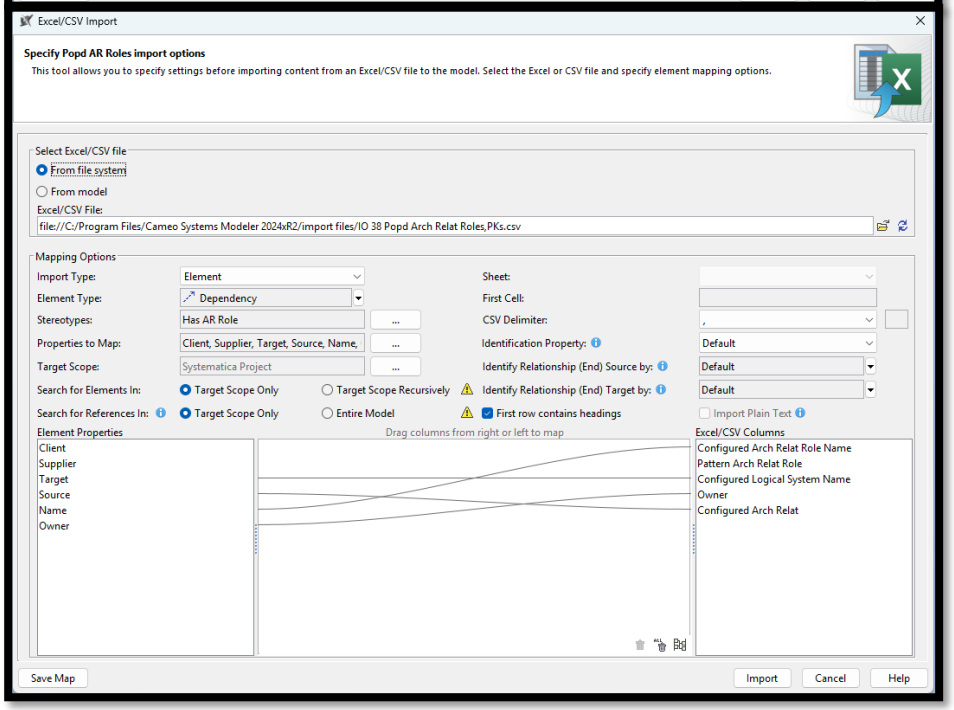


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd AR Item Flow

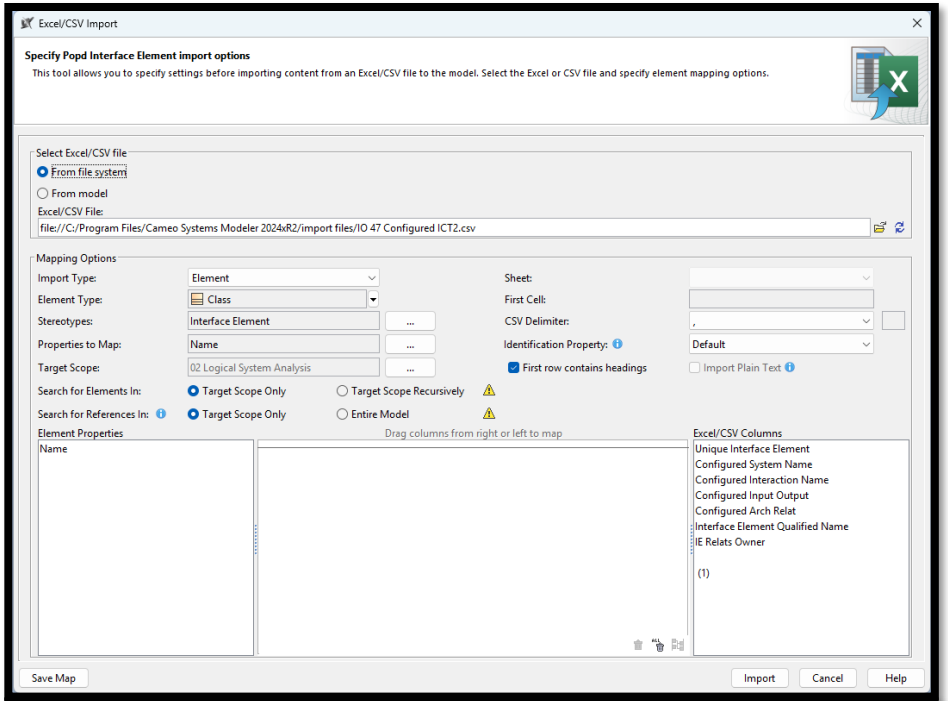


Popd AR Roles

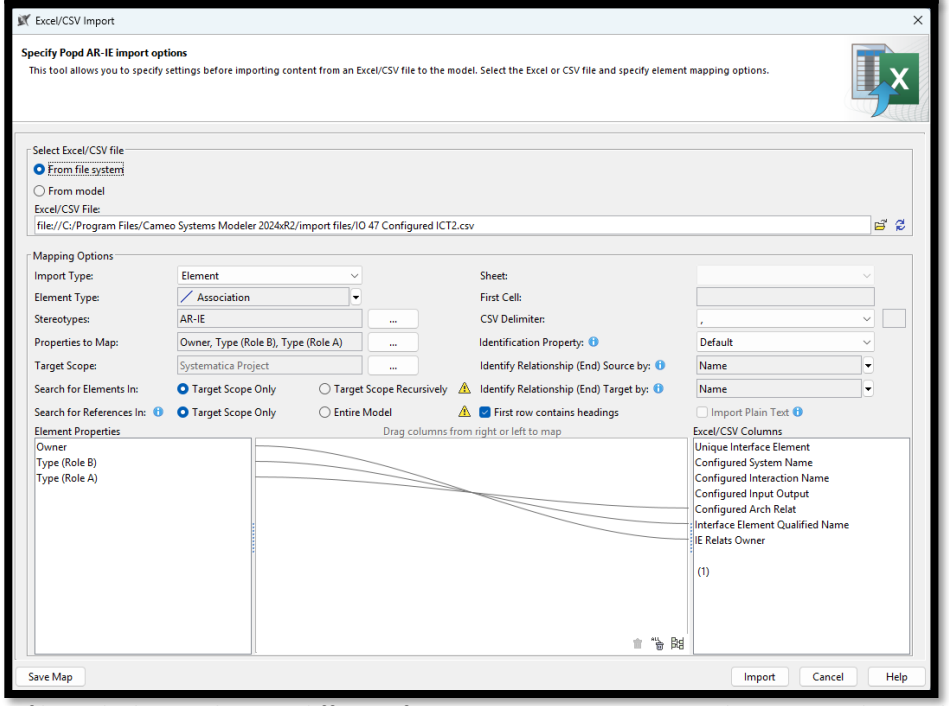


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Interface Element

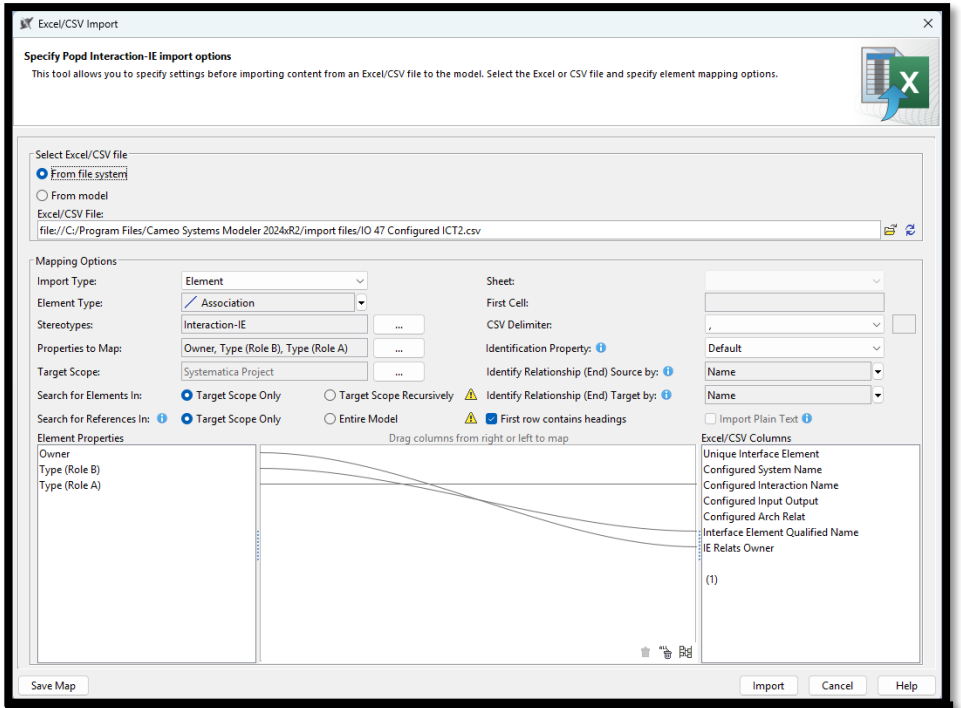


Popd AR-IE

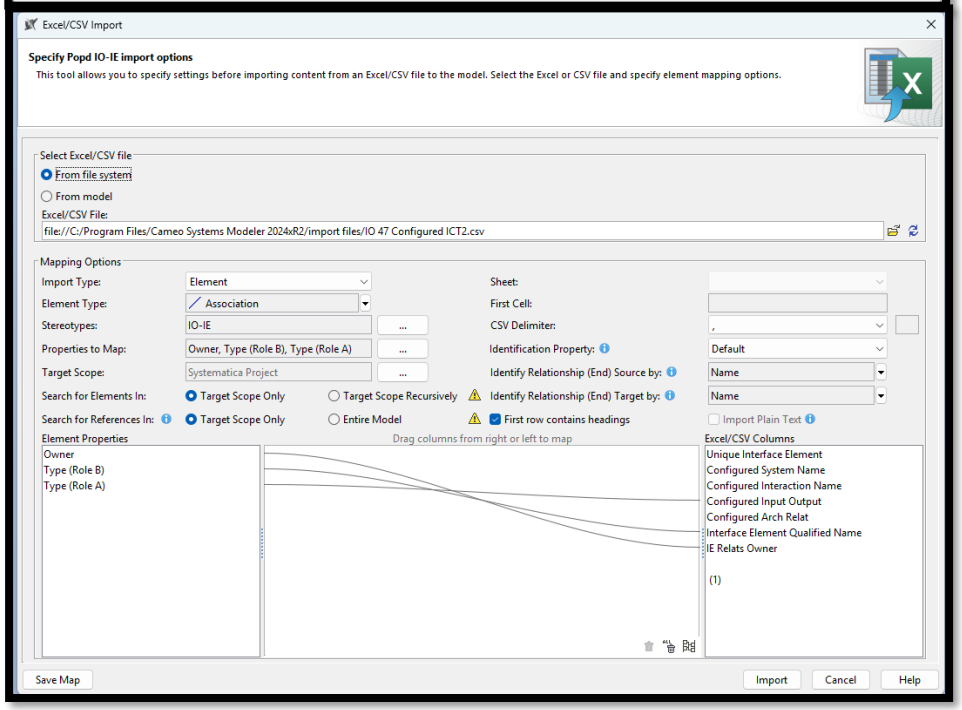


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Interaction-IE

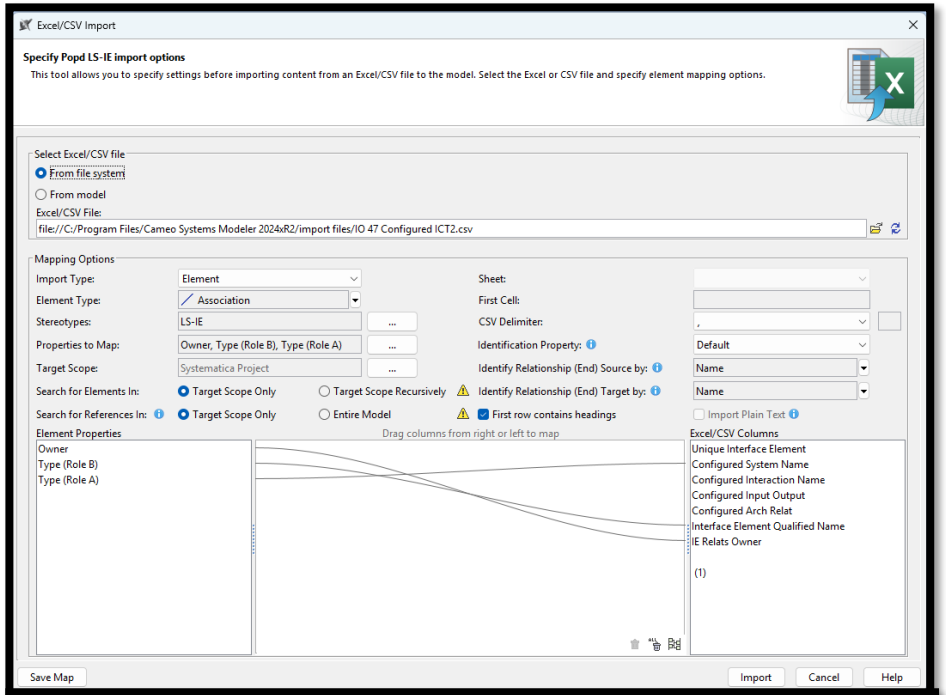


Popd IO-IE

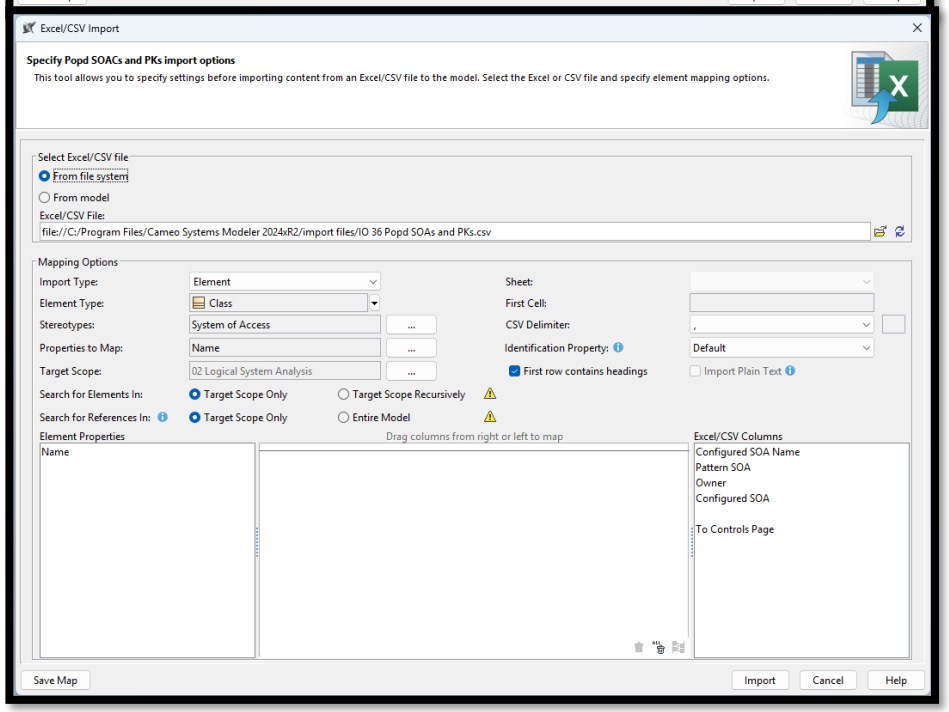


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd LS-IE

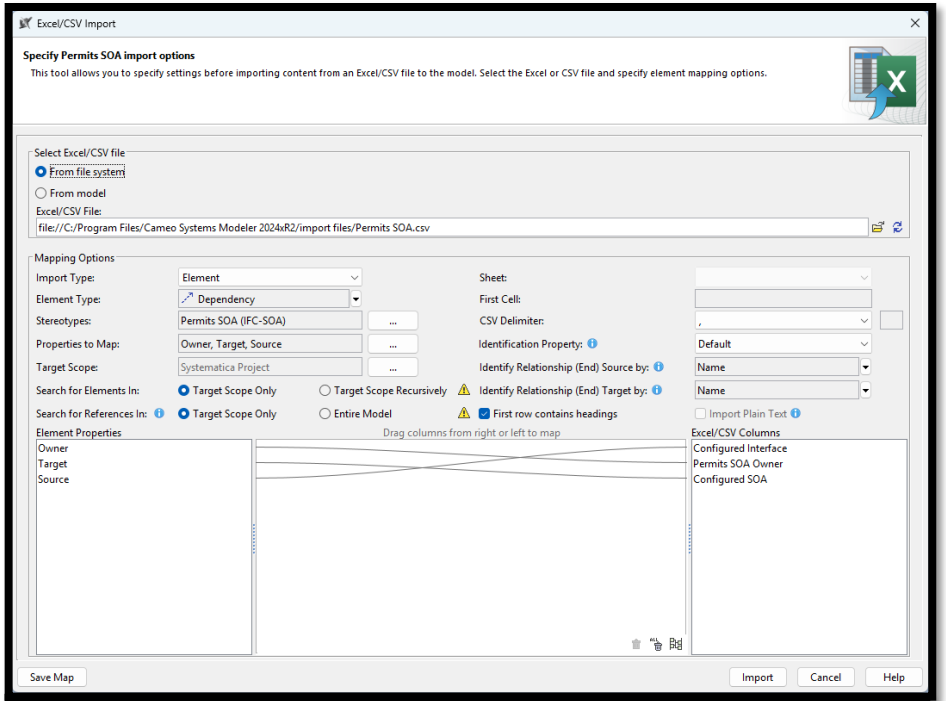


Popd SOACs and PKs

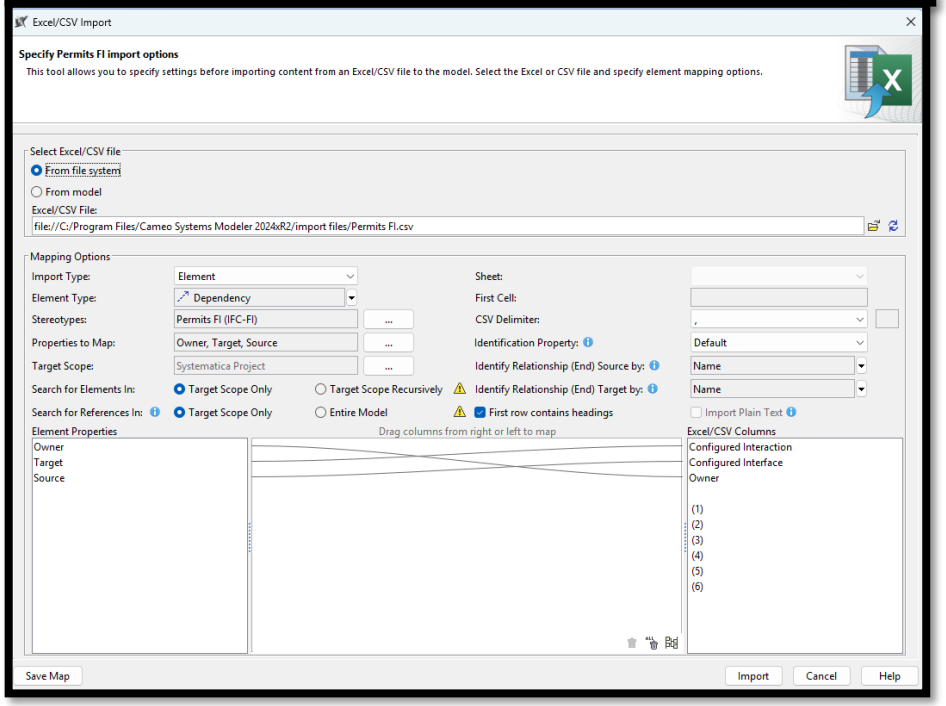


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Permits SOA

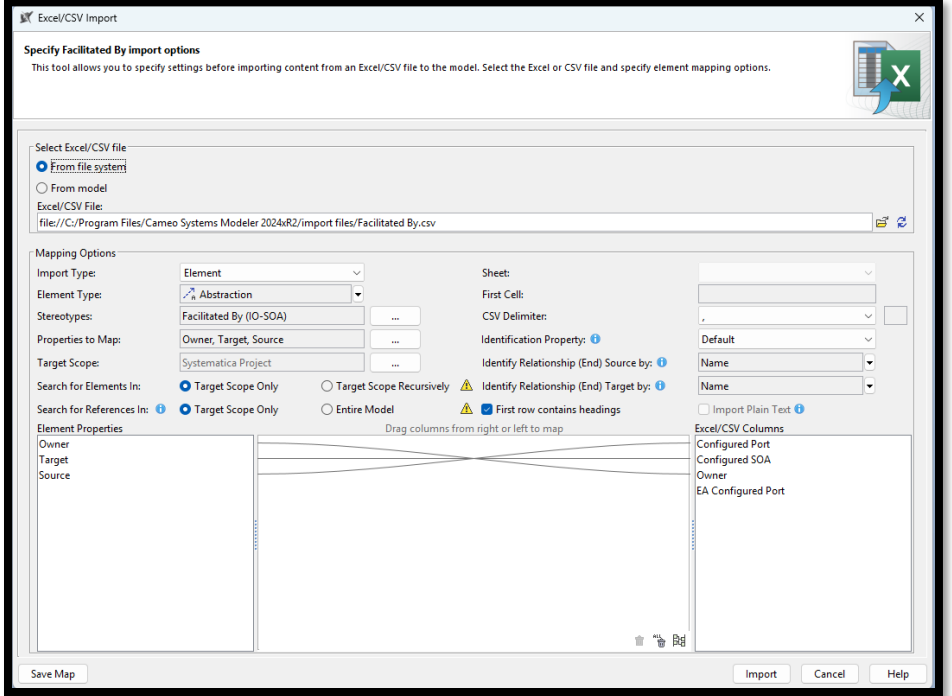


Popd Permits FI

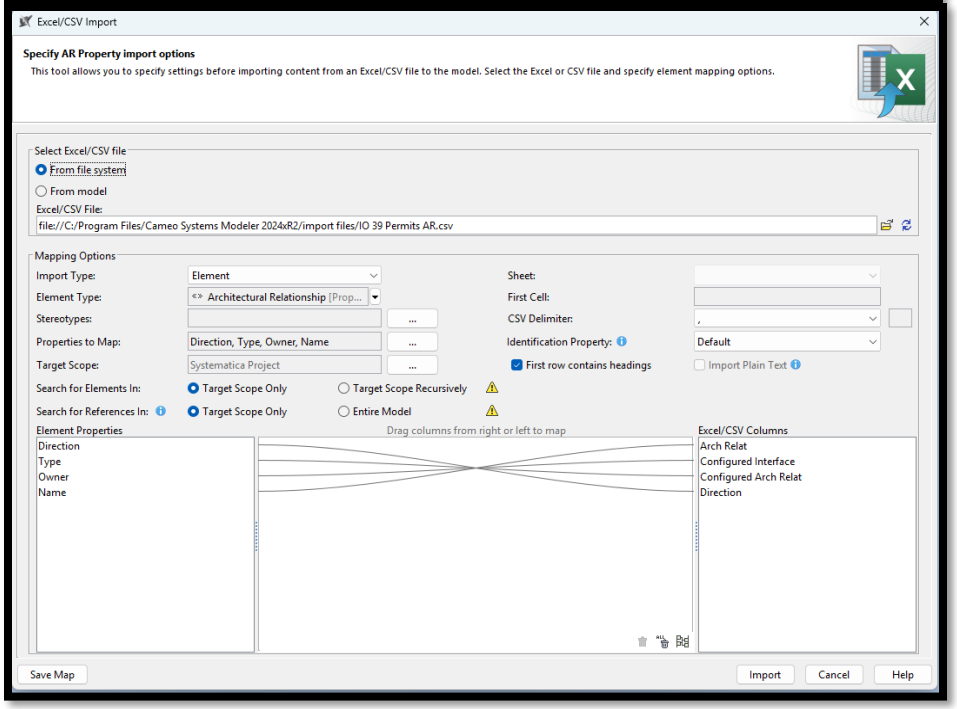


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Facilitated By

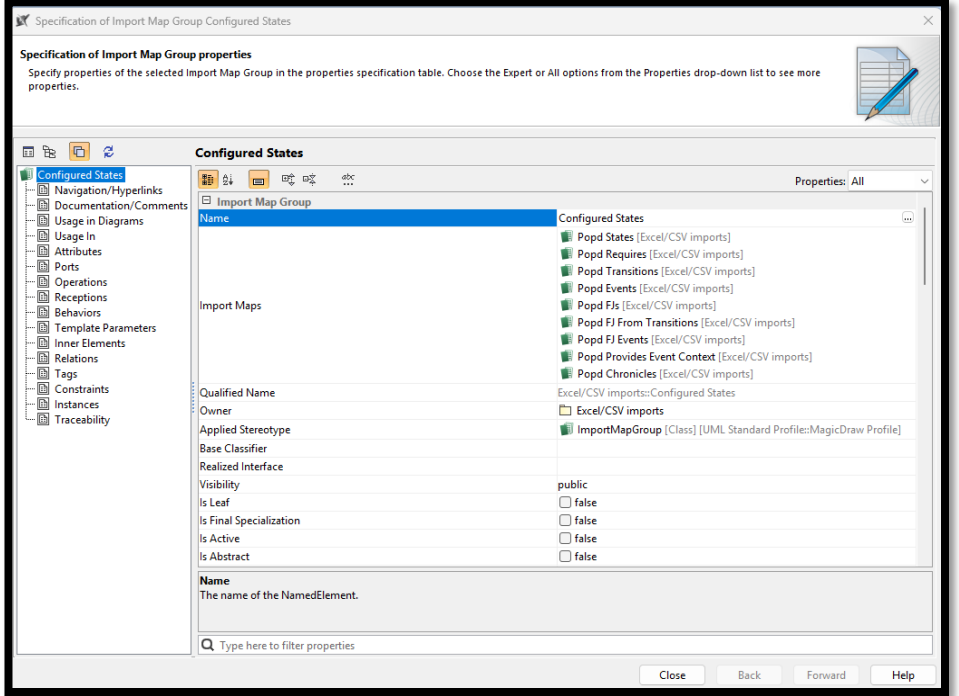


Popd AR Property

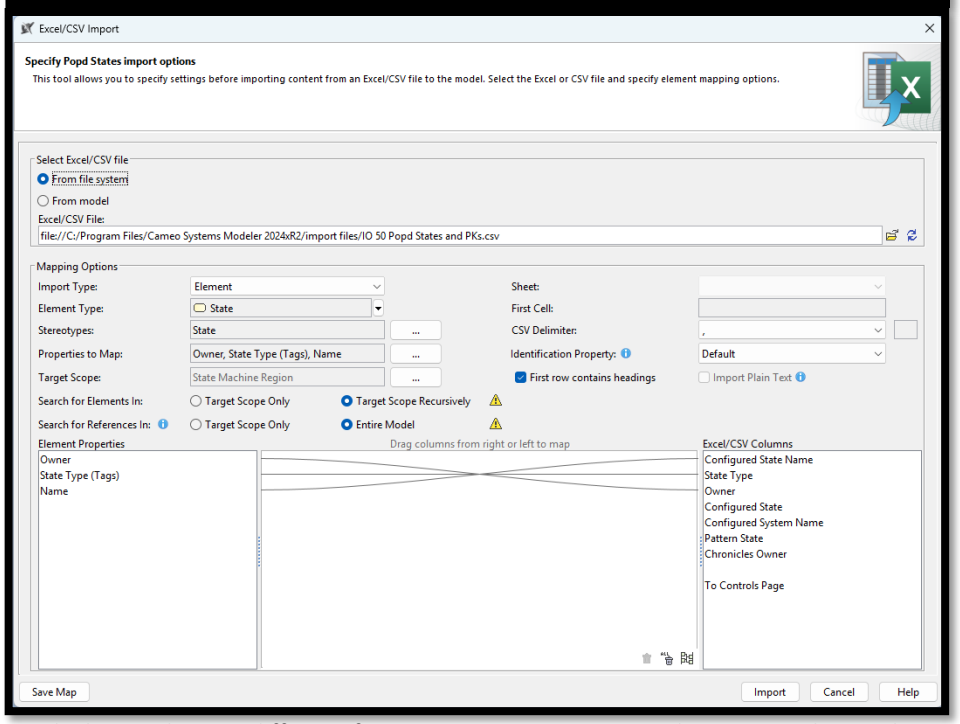


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Map Group: Configured States

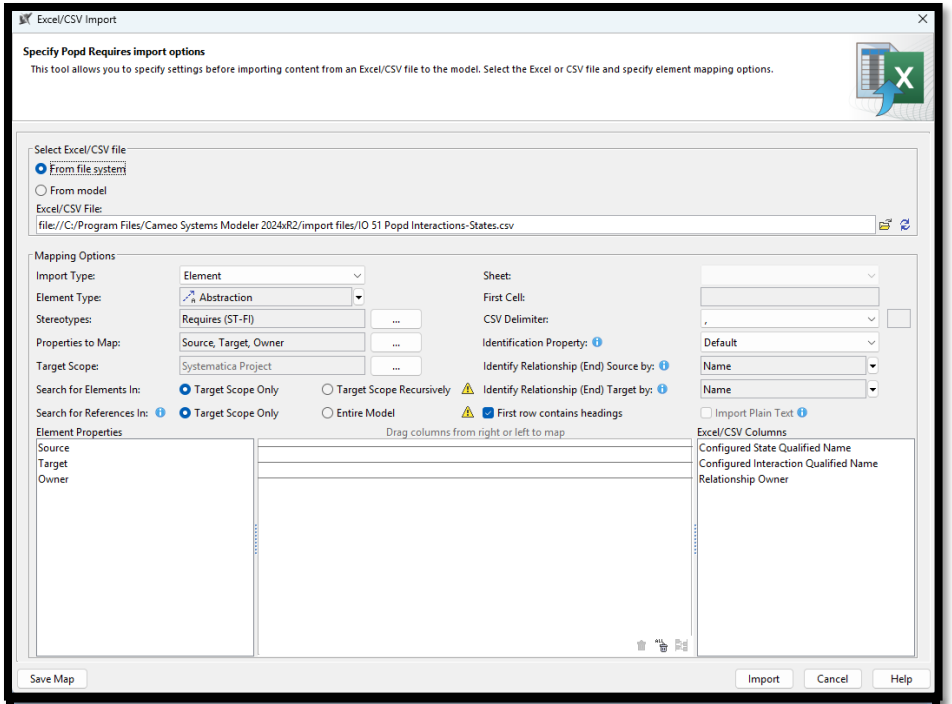


Popd States

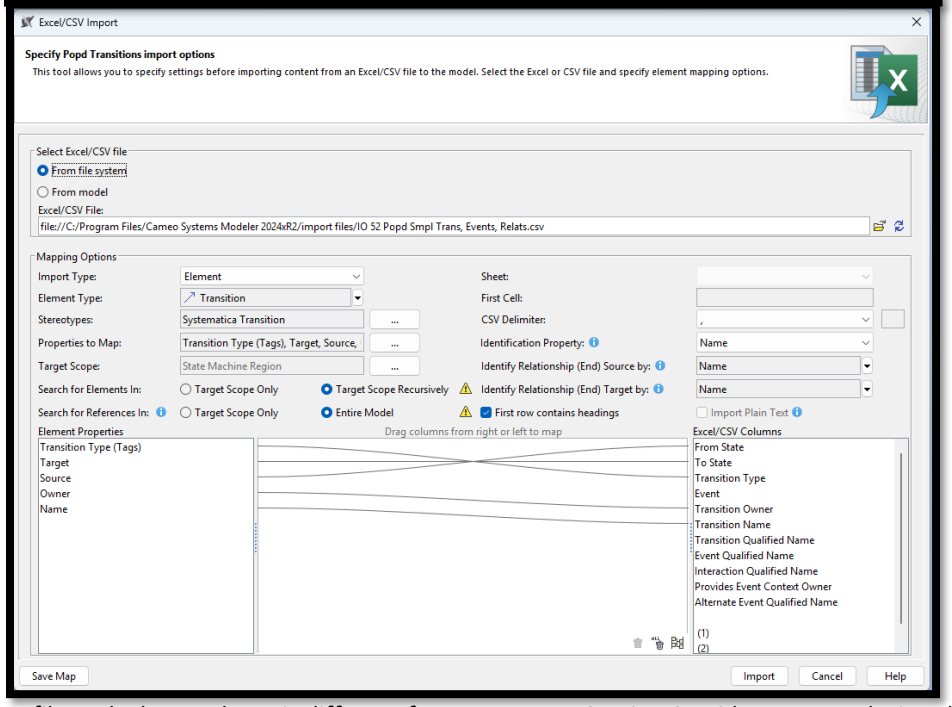


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Requires

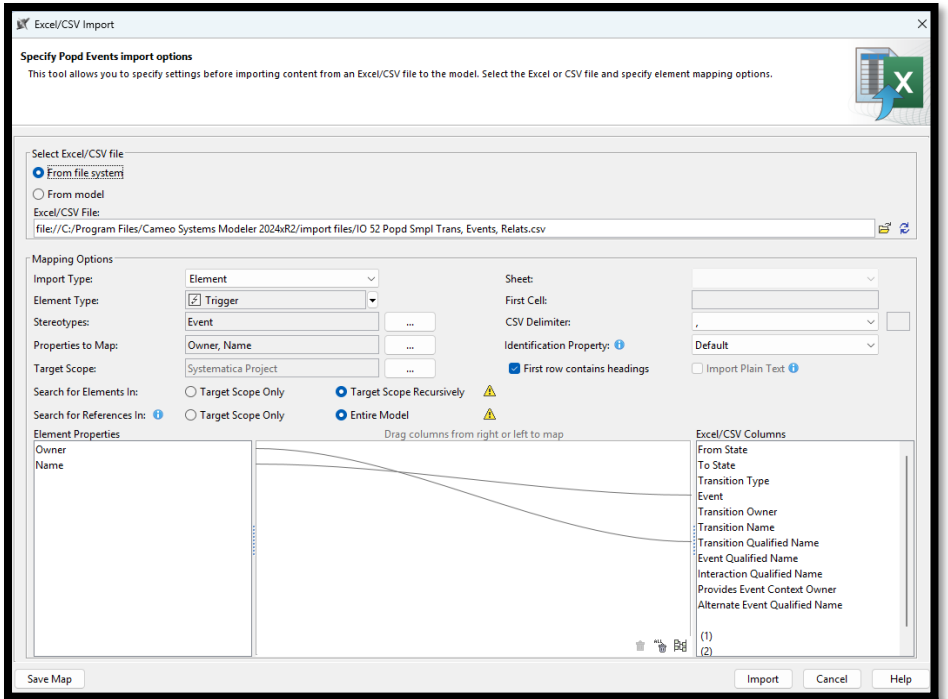


Popd Transitions

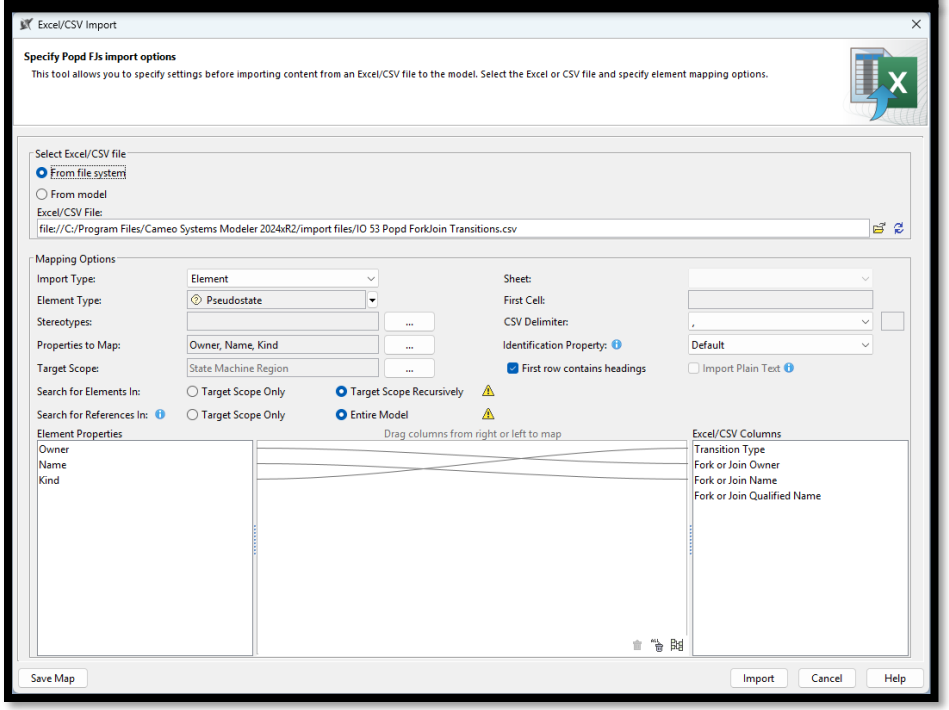


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Events

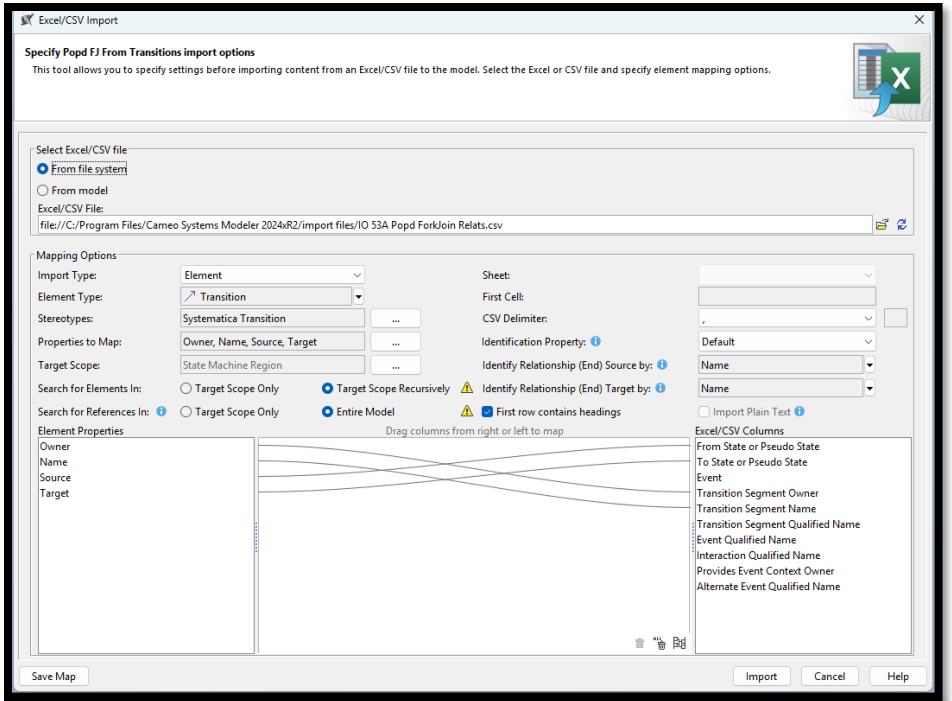


Popd FJs

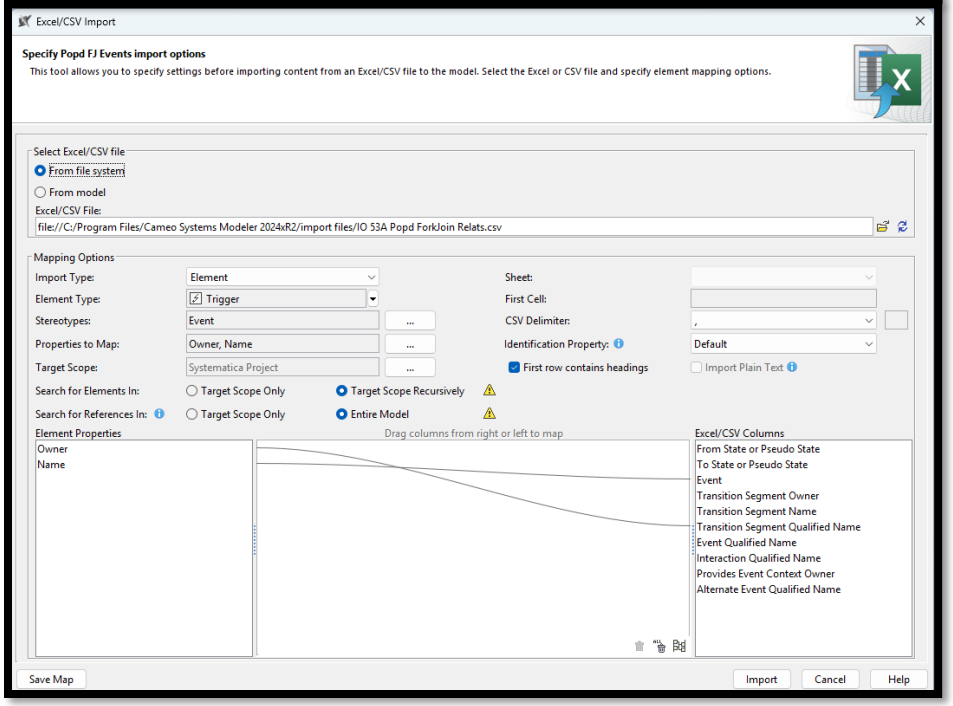


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd FJ From Transitions

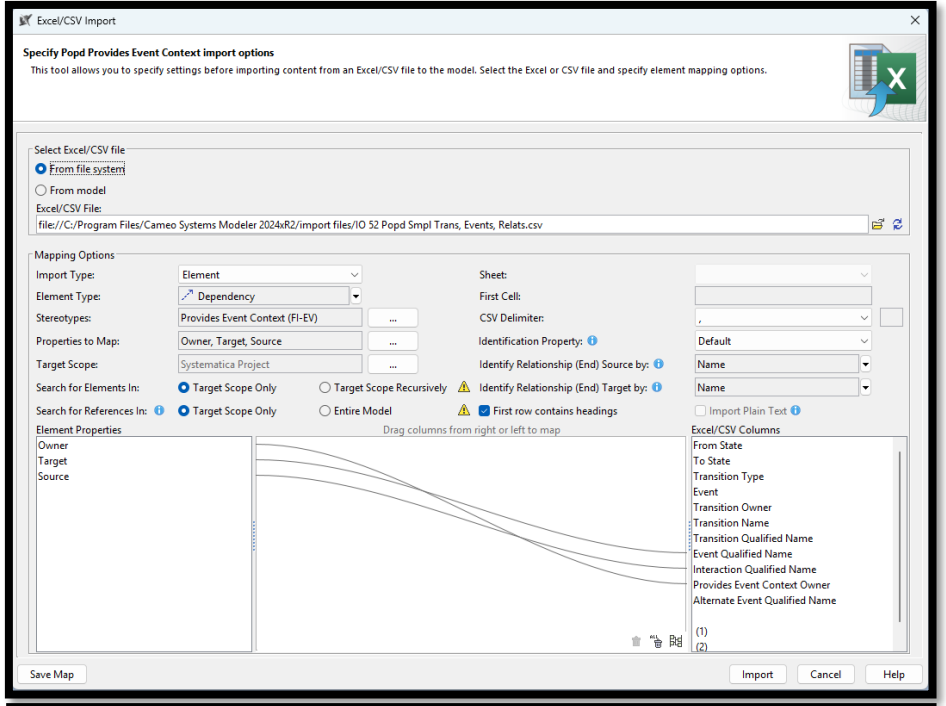


Popd FJ Events

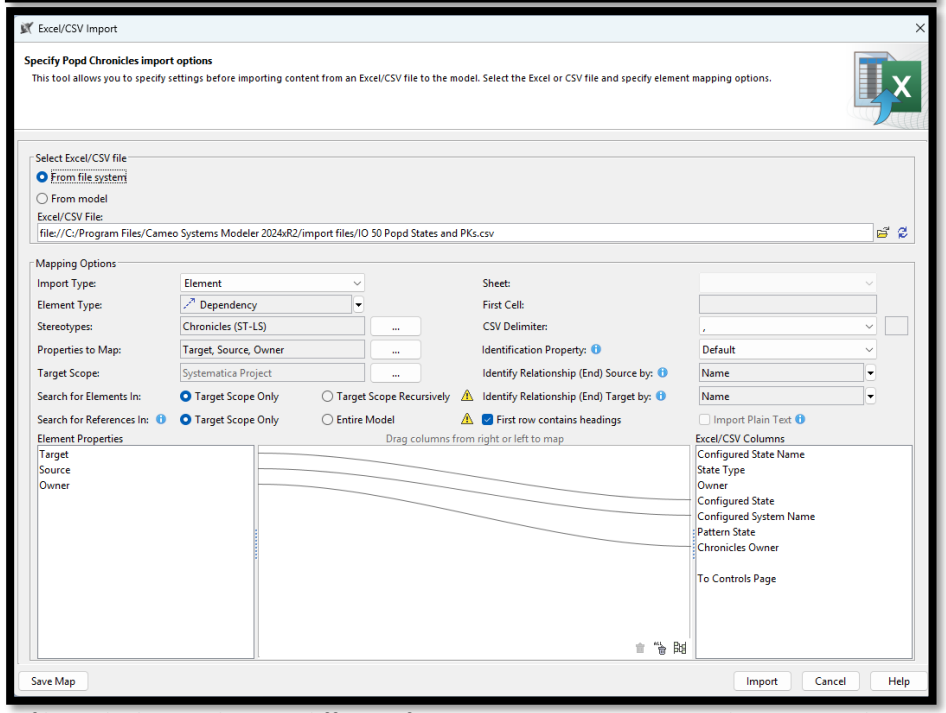


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Provides Event Context



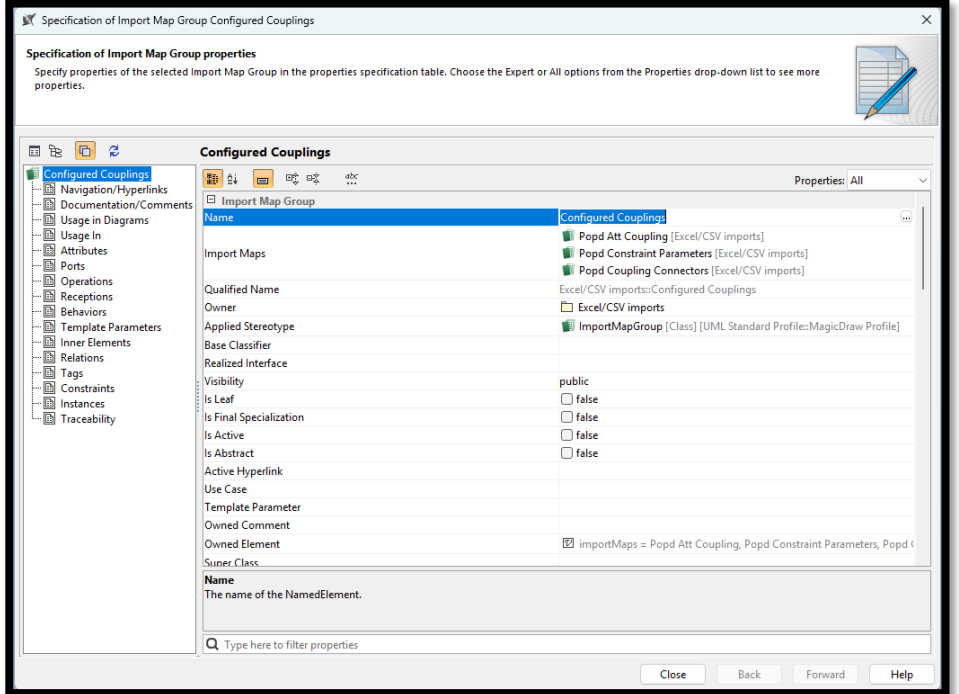
Popd Chronicles

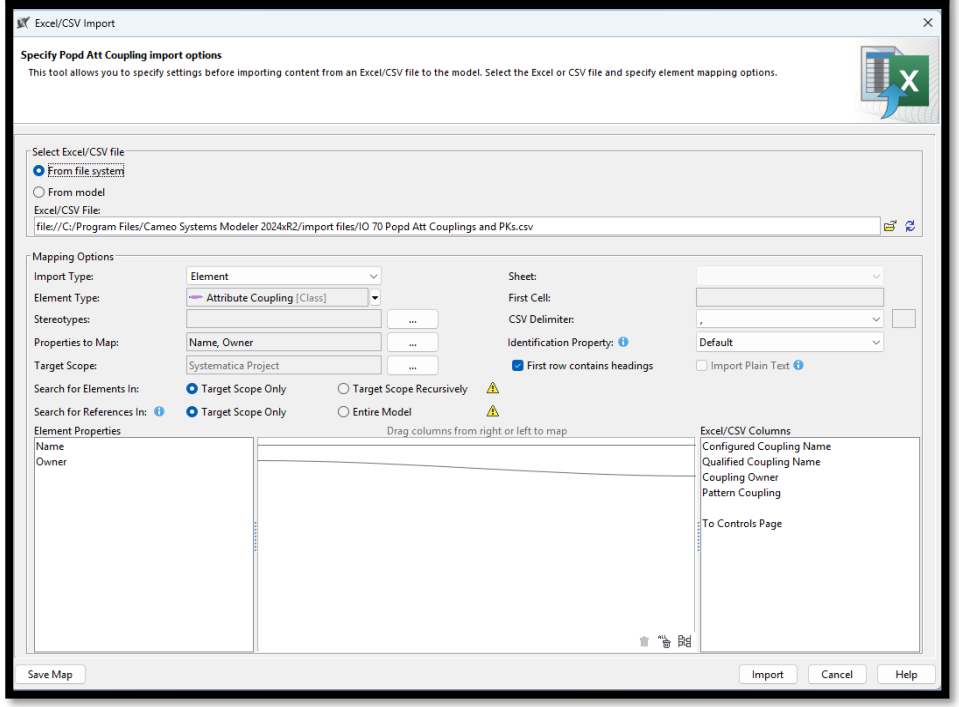


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Map Group: Configured Couplings

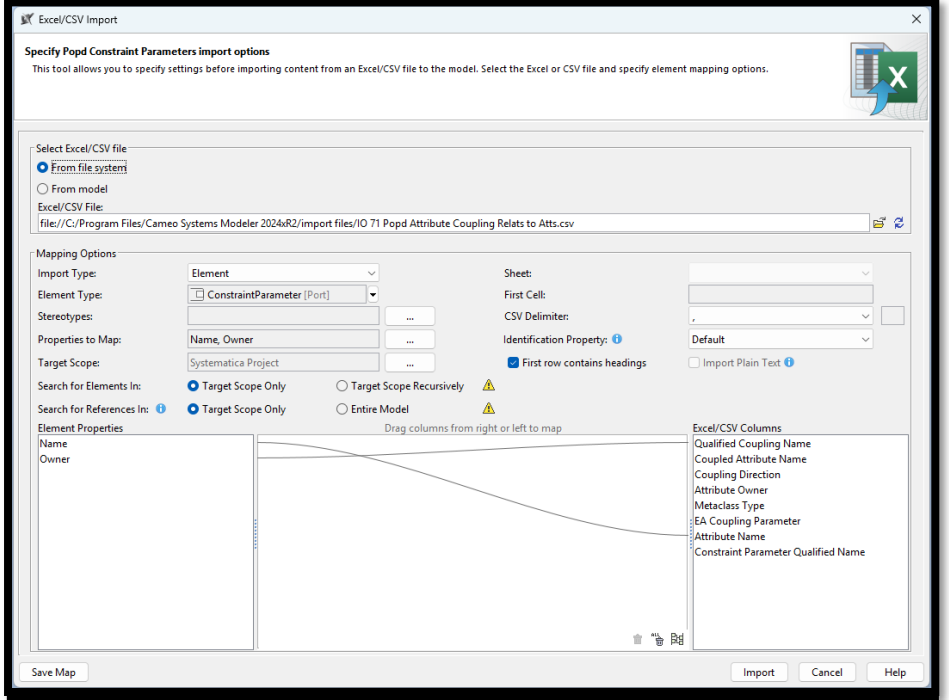
Popd Att Coupling



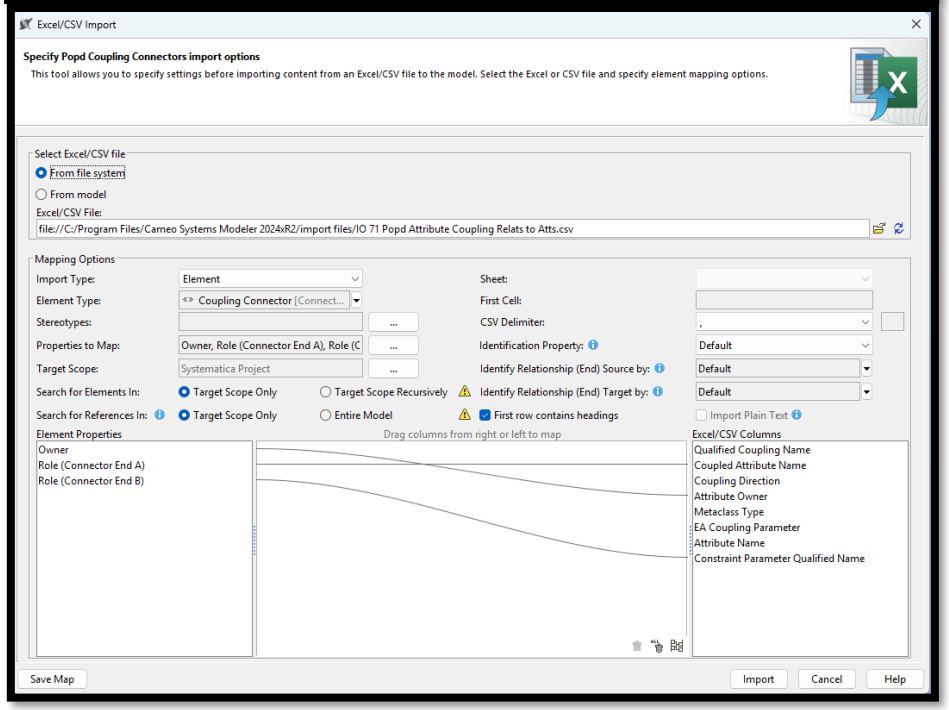


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Constraint Parameters

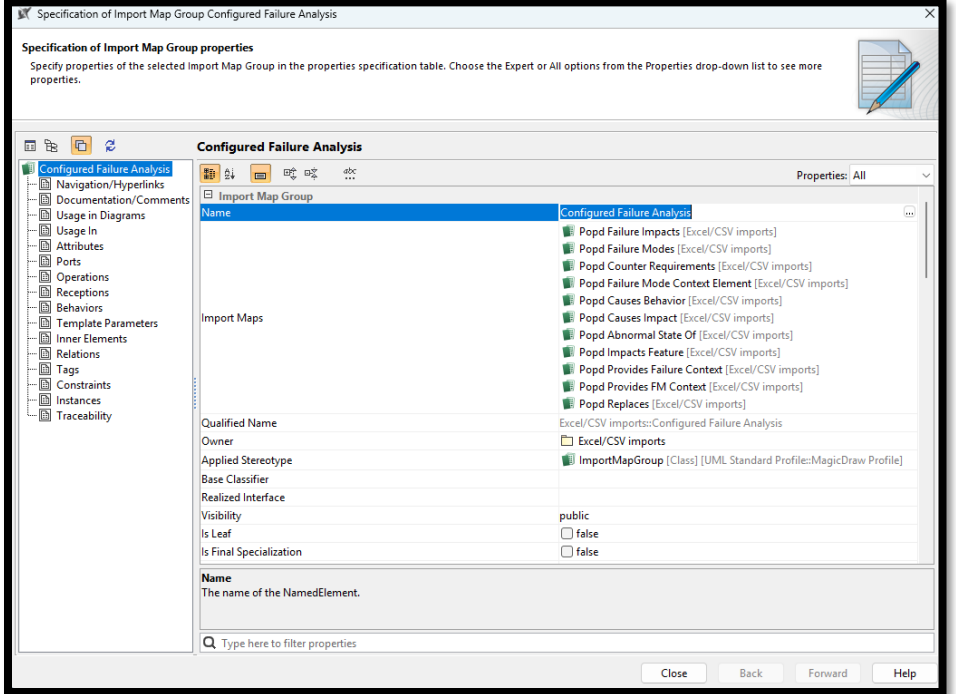


Popd Coupling Connectors

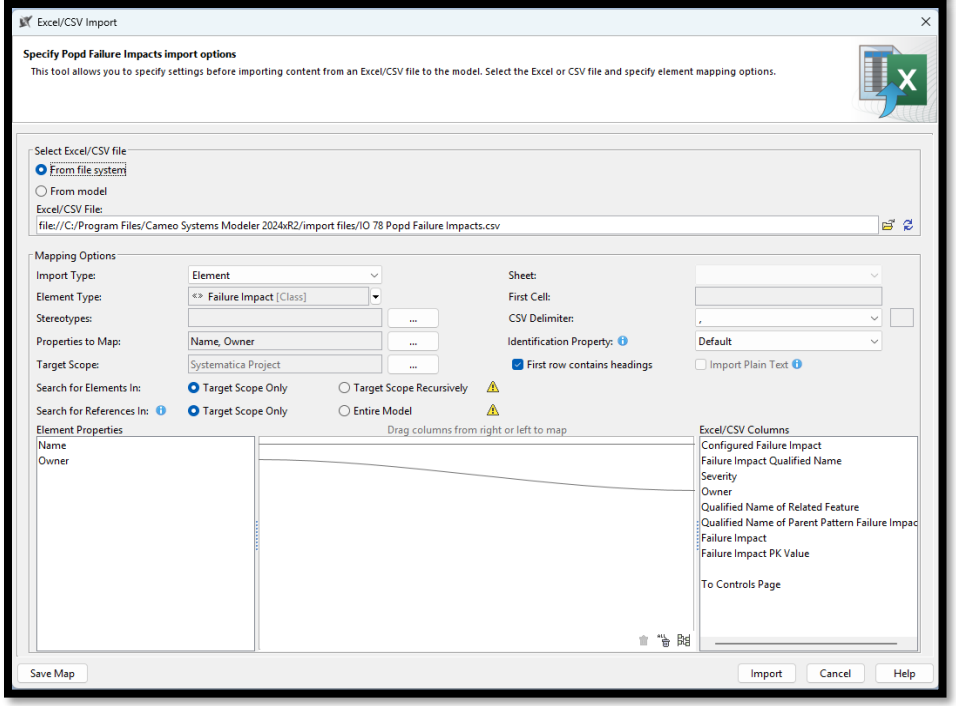


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Map Group: Configured Failure Analysis

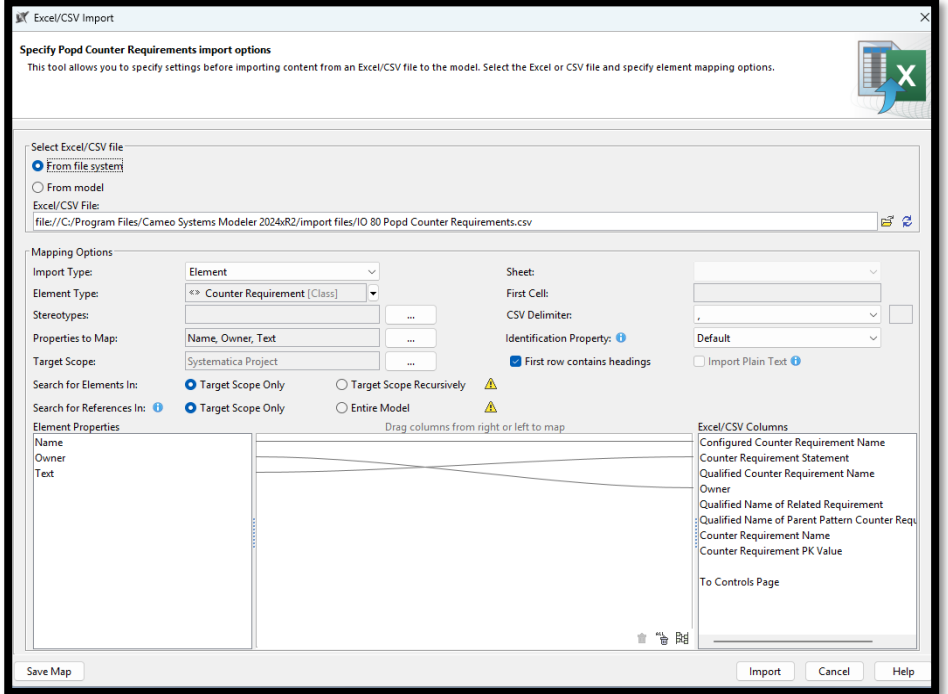


Popd Failure Impacts

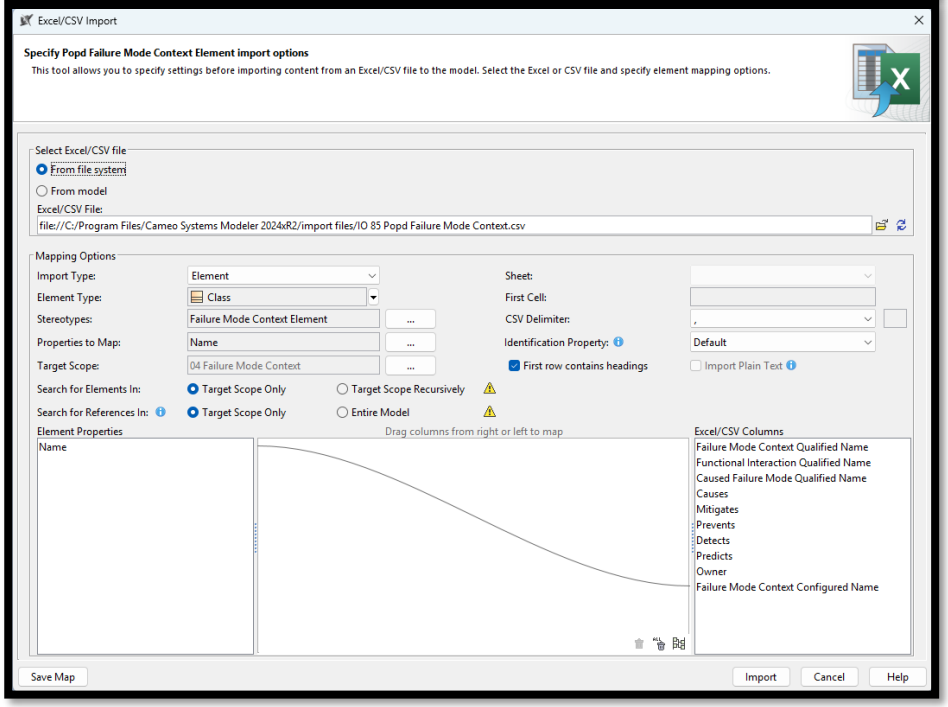


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Counter Requirements

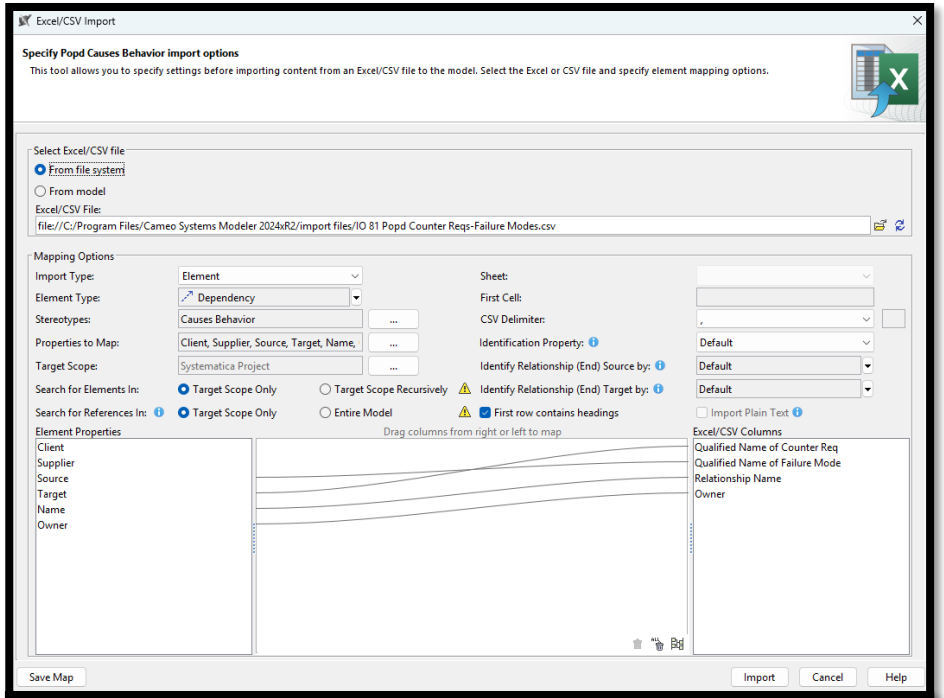


Popd Failure Mode Context Element

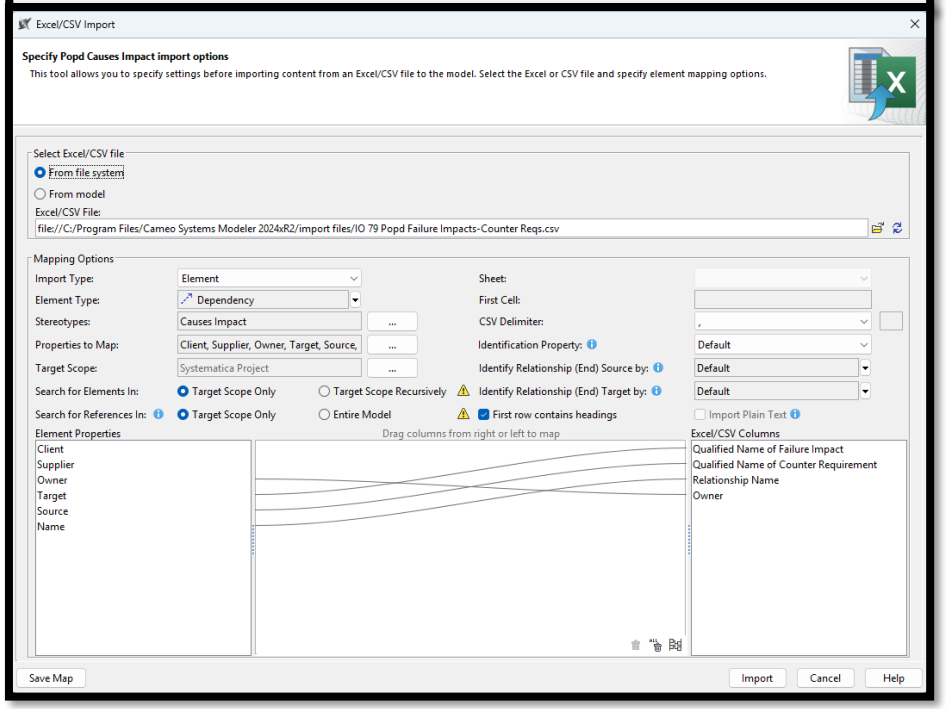


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Causes Behavior

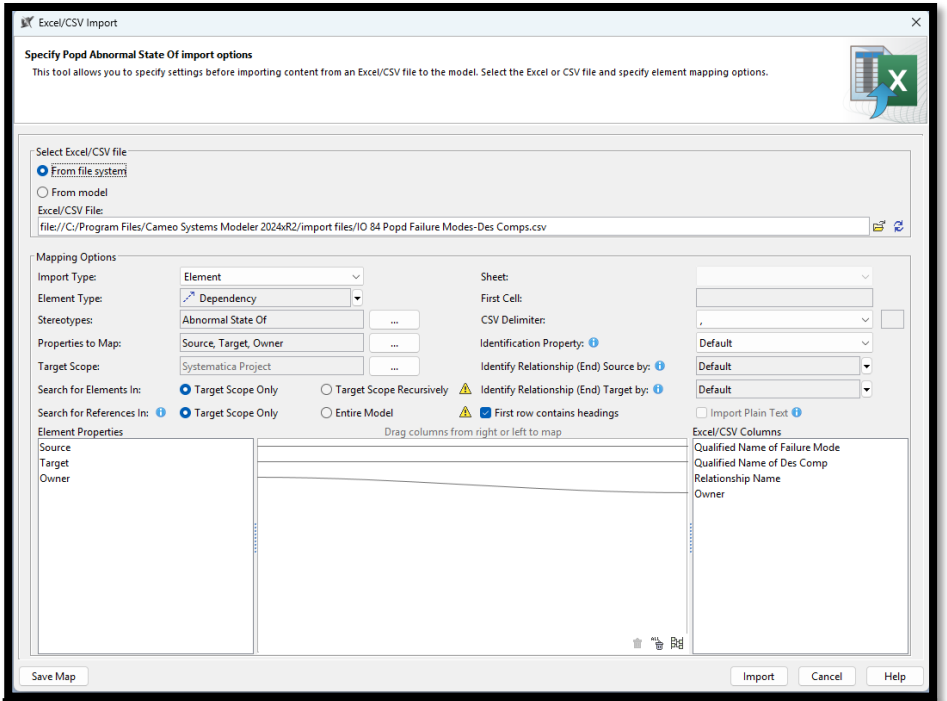


Popd Causes Impact

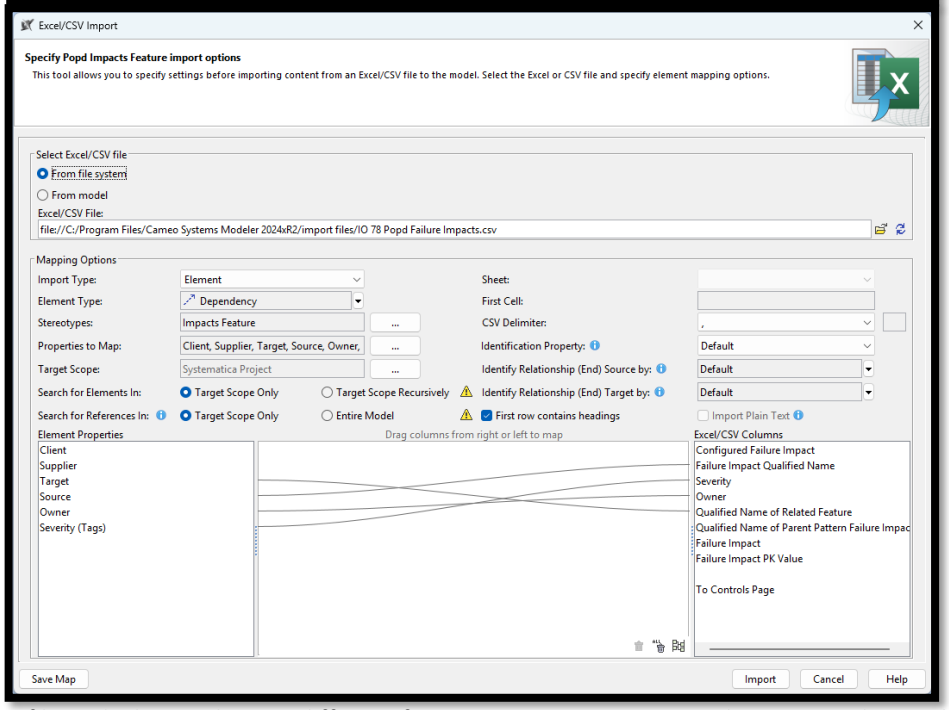


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Abnormal State Of

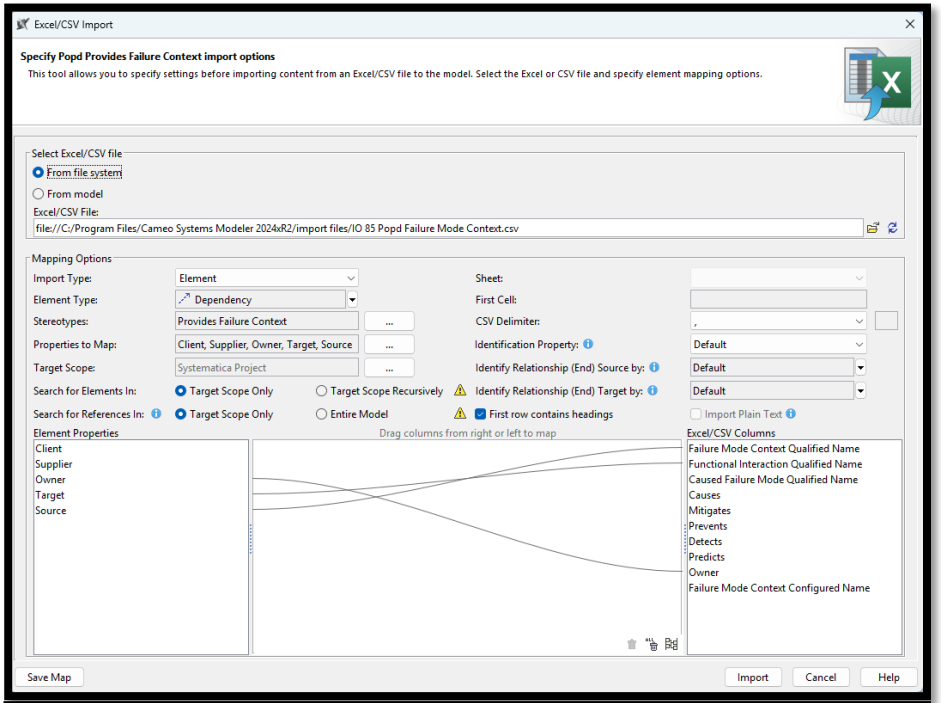


Popd Impacts Feature

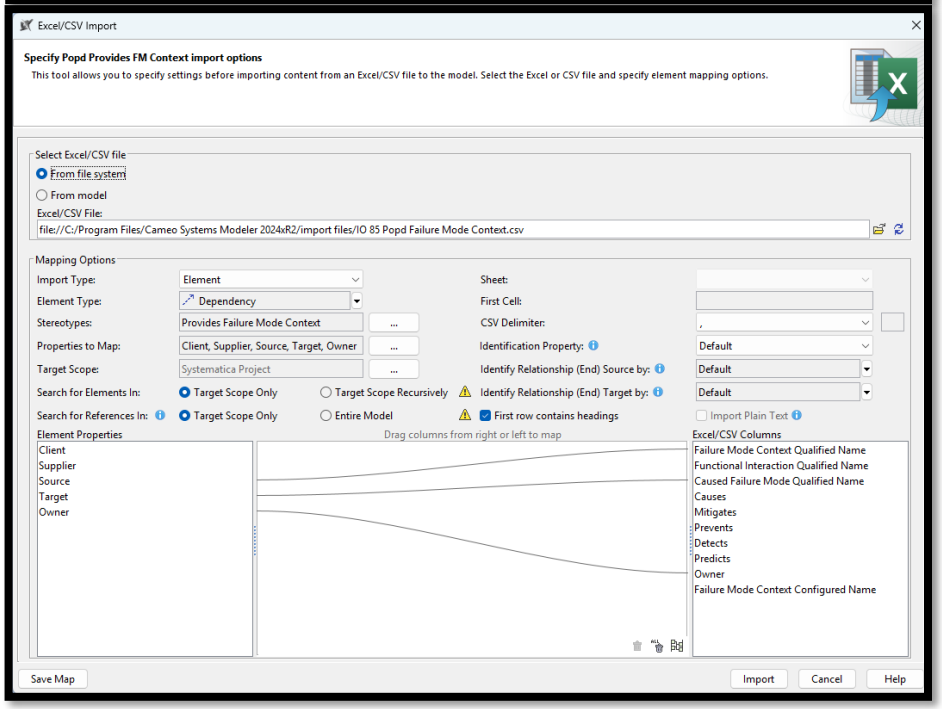


***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Provides Failure Context

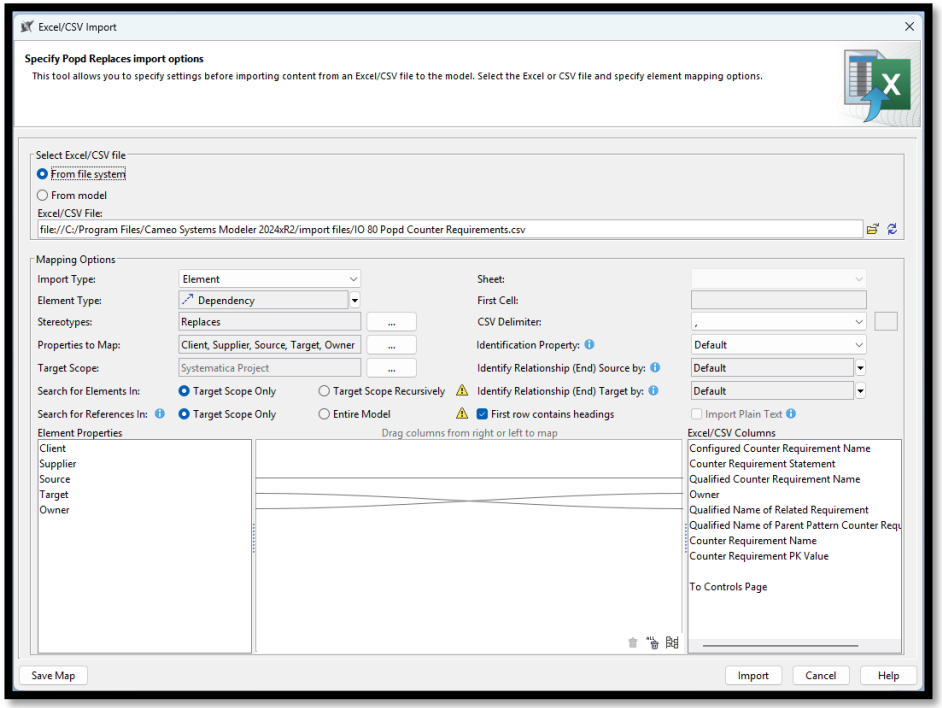


Popd Provides FM Context



***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.

Popd Replaces



***If the destination file path shown above is different from yours, see Section 2 guidance on replacing that string.