

A Modelling Language and Tools Agnostic Model-based Systems Engineering Strategy in Rolls-Royce



Mole Li

Principal Engineer

28/11/2022



Outline

Overview of Modelling Language Agnostic Strategy

- Challenges of Modelling Language Agnostic Strategy
- Universal Meta-model/Modelling Method
- Project/Domain Specific Modelling Method
- Agile Engineering Practices

Overview of Modelling Tools Agnostic Strategy

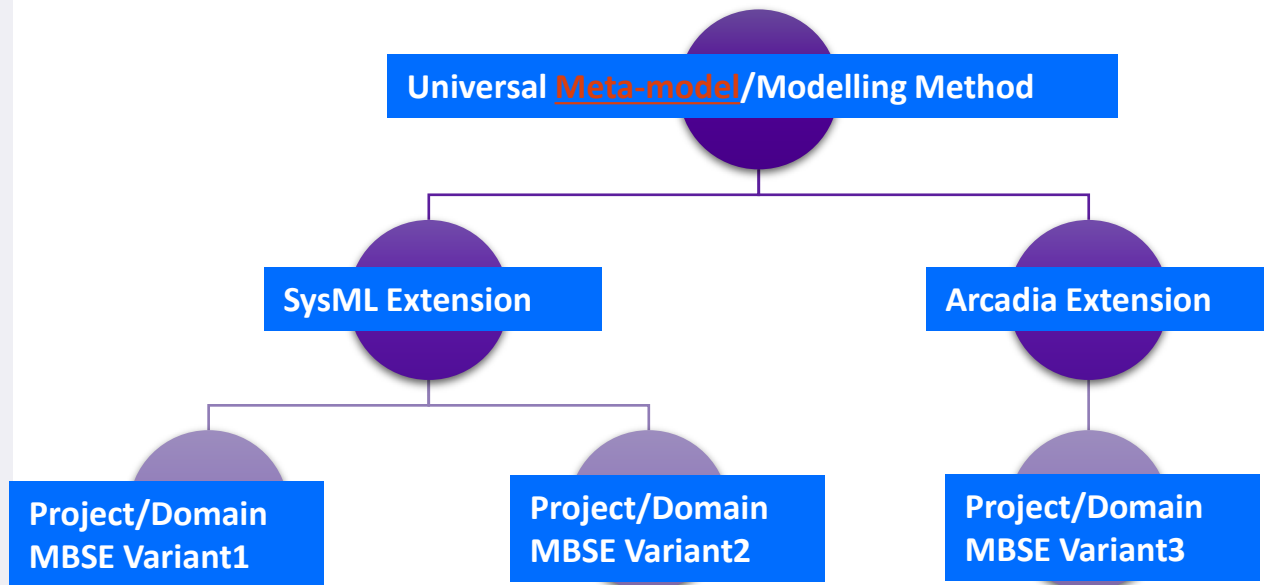
- Why XMI does not work
- Ecosystem Backend Tool: Using Universal Meta-model to Recognise Project/Domain Specific Model
- Ecosystem Backend Tool: Document Generation, Model Checking, Metrics Report, Requirement Management, Simulation
- Ecosystem Backend Tool: Enabling and Optimise Change Control and Configure Management

Model-based Systems Engineering Future



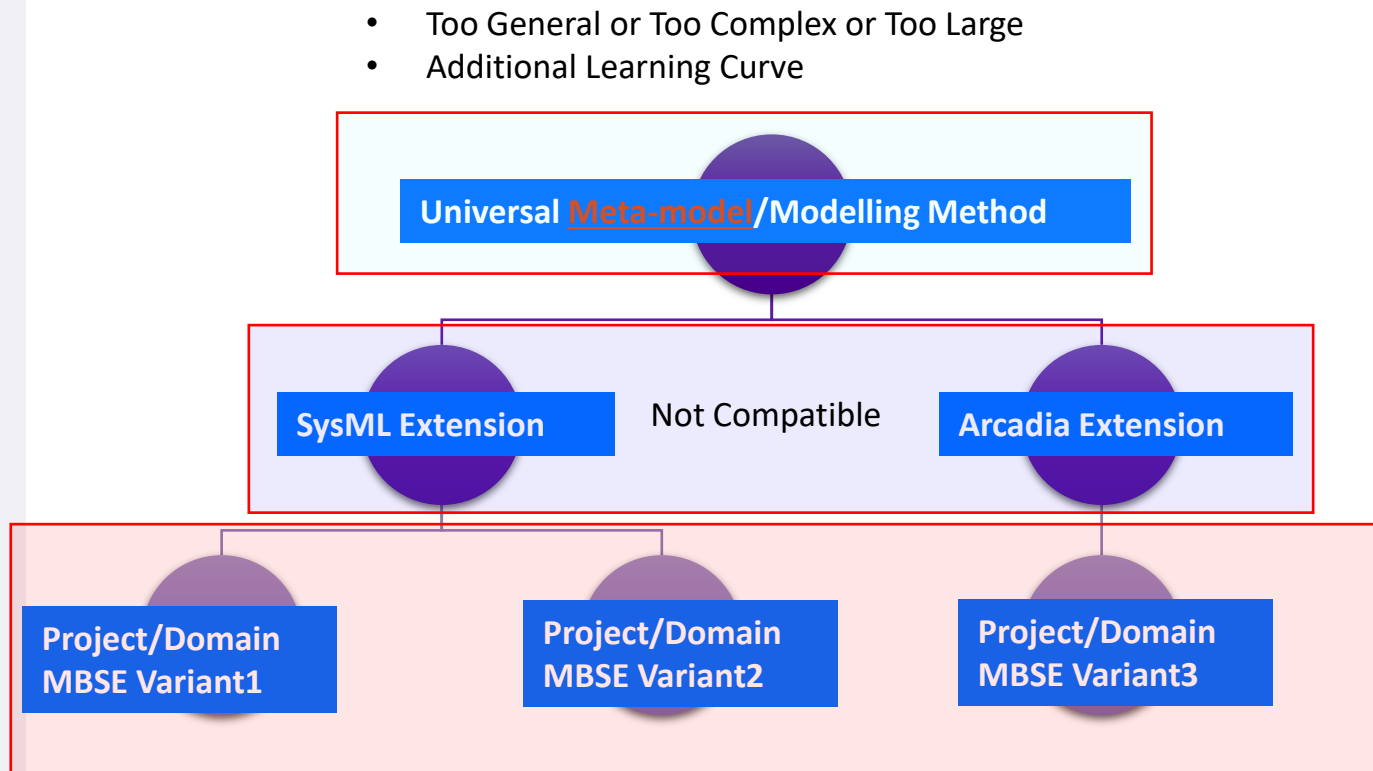
Overview:

Modelling Language
Agnostic Strategy



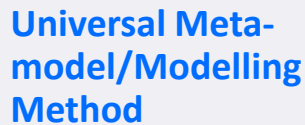


Challenges of Modelling Language Agnostic Strategy



Reality for most of large-scale organizations

- Domain/Context/Political Difference
- Tool Set & Tool Environment Difference



- Formalised (Formal Method)
- No Duplicated Concept
- Modelling Standard
- Auditing Friendly

The diagram illustrates the following classes and their relationships:

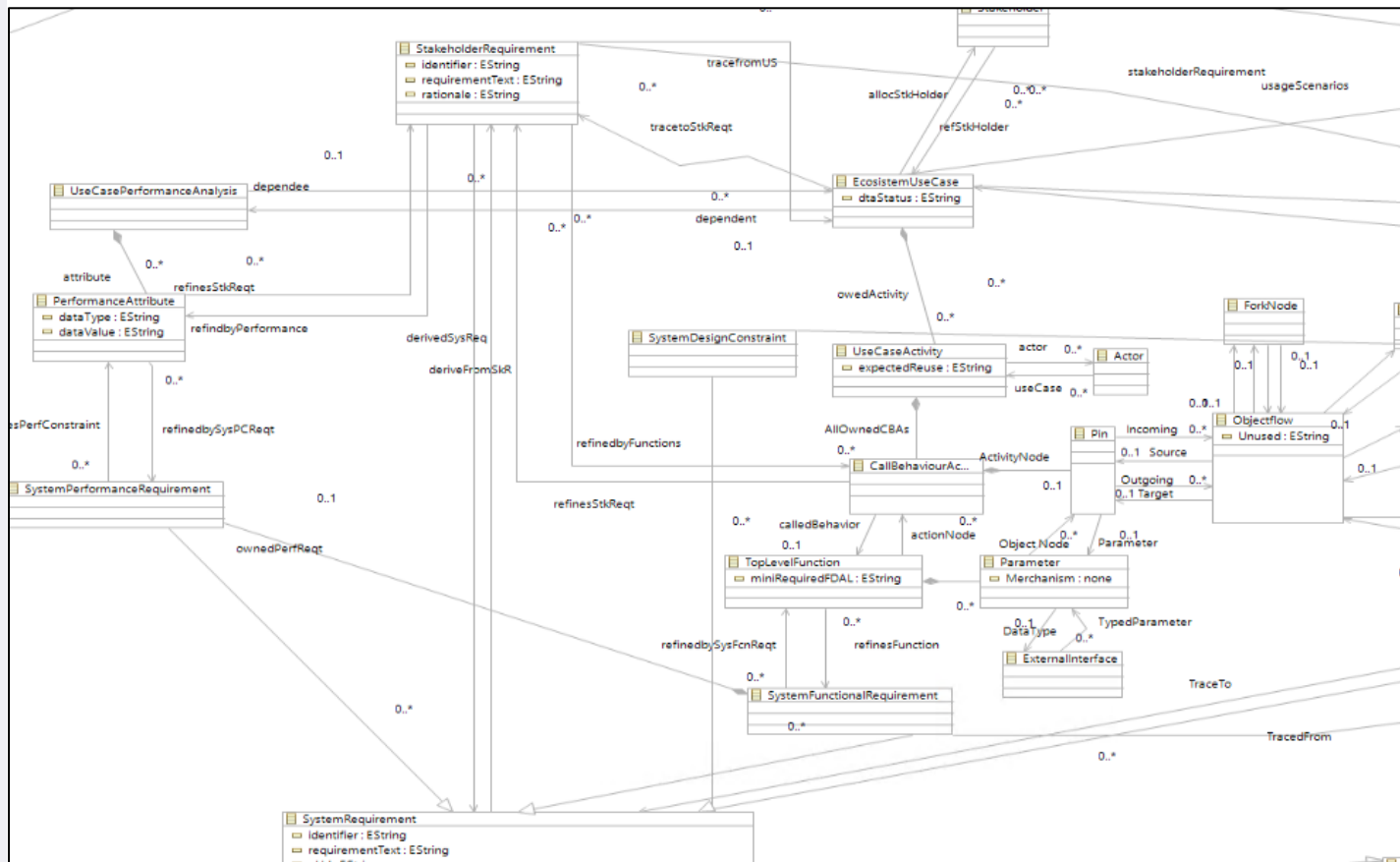
- StakeholderRequirement** (Attributes: identifier: EString, requirementText: EString, rationale: EString)
 - Generalized by: **StakeholderRequirement** (Attributes: identifier: EString, requirementText: EString, rationale: EString)
 - Associated with: **Stakeholder** (multiplicity 0..*) via role **stakeholder**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **usageScenario**
 - Associated with: **Function** (multiplicity 0..*) via role **function**
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **performanceConstraintGroup**
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **performanceConstraint**
 - Associated with: **SystemRequirement** (multiplicity 0..*) via role **systemRequirement**
 - Associated with: **Actor** (multiplicity 0..*) via role **actors**
 - Associated with: **Stakeholder** (multiplicity 0..*) via role **refStkHolder**
 - Associated with: **ExchangeType** (multiplicity 0..*) via role **exchangeTypes**
- PerformanceConstraintGroup** (Attributes: ownPerformanceConstraint, refPerfConstraint, refPCGroup)
 - Generalized by: **PerformanceConstraintGroup** (Attributes: ownPerformanceConstraint, refPerfConstraint, refPCGroup)
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **ownPerformanceConstraint**
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **refPerfConstraint**
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **refPCGroup**
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **tracedbyPerformance**
 - Associated with: **PerformanceConstraint** (multiplicity 0..*) via role **appliedToFCI**
- PerformanceConstraint** (Attributes: dataType: EString, dataValue: EString)
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **performanceConstraintGroup**
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **refPerfConstraint**
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **refPCGroup**
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **tracedbyPerformance**
 - Associated with: **PerformanceConstraintGroup** (multiplicity 0..*) via role **appliedToFCI**
- UsageScenario** (Attributes: dtaStatus: EString, expectedReuse: ES)
 - Associated with: **StakeholderRequirement** (multiplicity 0..*) via role **usageScenario**
 - Associated with: **Function** (multiplicity 0..*) via role **function**
 - Associated with: **Function** (multiplicity 0..*) via role **tracedbyFunction**
 - Associated with: **Function** (multiplicity 0..*) via role **appliedPer**
 - Associated with: **Function** (multiplicity 0..*) via role **parameters**
 - Associated with: **Function** (multiplicity 0..*) via role **outGoingInComing**
 - Associated with: **Function** (multiplicity 0..*) via role **source**
 - Associated with: **Function** (multiplicity 0..*) via role **target**
- Function** (Attributes: miniRequiredFDAL)
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **function**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **tracedbyFunction**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **appliedPer**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **parameters**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **outGoingInComing**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **source**
 - Associated with: **UsageScenario** (multiplicity 0..*) via role **target**
- Actor**
 - Associated with: **StakeholderRequirement** (multiplicity 0..*) via role **actors**
- Stakeholder**
 - Associated with: **StakeholderRequirement** (multiplicity 0..*) via role **refStkHolder**
- ExchangeType**
 - Associated with: **StakeholderRequirement** (multiplicity 0..*) via role **exchangeTypes**
- Parameter** (Attributes: direction: EString)
 - Associated with: **Function** (multiplicity 0..*) via role **parameters**
 - Associated with: **Function** (multiplicity 0..*) via role **typedParameter**
- ExchangeFlow**
 - Associated with: **Function** (multiplicity 0..*) via role **outGoingInComing**
 - Associated with: **Function** (multiplicity 0..*) via role **source**
 - Associated with: **Function** (multiplicity 0..*) via role **target**
- ConfiguredModellingInterface** (Attributes: frontendId: EString)
 - Associated with: **Function** (multiplicity 0..*) via role **target**



Project/Domain

MBSE Variant1

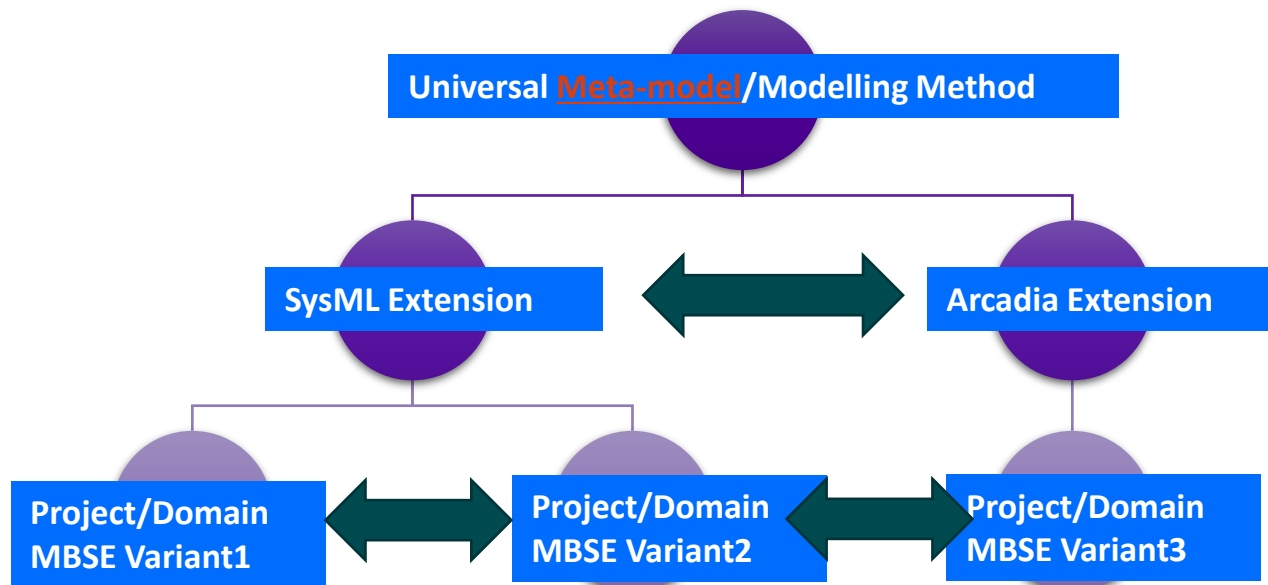
- Modelling Language: SysML + Stereotype
- Modelling Tool: Integrity Modeler
- User Guidance
- Project Additional Elements
- Tooling Support Profiles





Transformation Rule Documented

- Reduce Training Efforts (only learn delta part)
- Facilitates Agile Engineering Practices



Domain Class(Backend)	Artisan Class	SysML Implementation	Capella Class	Cameo Class
Stakeholder	Stakeholder	Block+stereotype	Stakeholder	Stakeholder
Stakeholder Requirement	Stakeholder Rqmt	Requirement + Stereotype	Requirement(VP)	Stakeholder Rqmt
Actor	Actor	Actor	Actor[System Component->Actor]	Actor
Usage Scenario	ECOSISem Use Case/Use Case Ac	Use Case + Stereotype/Activity + Stereotype	(System) Capability	ECOSISem Use Case/Use Case
Function Group			Functional Chain	
Function	Top Level Function/UC Call Behav	Activity + Stereotype/CBA	System Function	Top Level Function/UC Call
Exchange Flow	Object Flow	Data Flow	Functional Exchange	Object Flow
Parameter	Parameter	Parameter	Function Input/Output Port	Parameter
Exchange Type	External Interface	Interface Block + Stereotype	Exchange Item	External Interface
Interface	External Interface	Interface Block + Stereotype	Component Exchange	External Interface
PerformanceConstraintGroup	Use Case Performance Analysis	Block+Stereotype	Property Value Group	Use Case Performance Anal
Performance Constraint	Requirement Value	Block Property (of Performance Blk)	Property Value (Float)	Requirement Value
System Functional Requirement	System Requirement	Fcn Req + Stereotype	System Function (merge text)	System Requirement
System Performance requirement	System Requirement	Perf Req + Stereotype	Property Value (String)_Description	System Requirement
System Design Constraint	System Requirement	Design Const + Stereotype	Constraint	System Requirement
Function.refinedbySysmlReq			parent-child of System Functions	
Usage Scenario.refStkHolder	UC.referenced Stakeholders			UC.referenced Stakeholder
Stakeholder Requirement.alloctFromUS	traced from link StkRqt to UC		currently use outgoing links of Requirement	traced from link StkRqt to UC
Usage Scenario.alloctStkRqt	trace link from UC to StkRqt			trace link from UC to StkRqt
Stakeholder Requirement.tracedbyFunctions			currently use incoming links of Requirement(VP) at StkRqt	
Stakeholder Requirement.tracedbyPerformance			currently use incoming links of Requirement(VP) at StkRqt	
Stakeholder Requirement.tracedbyConstraint			currently use incoming links of Requirement(VP) at StkRqt	
Function.parameters			System Function.inputs	

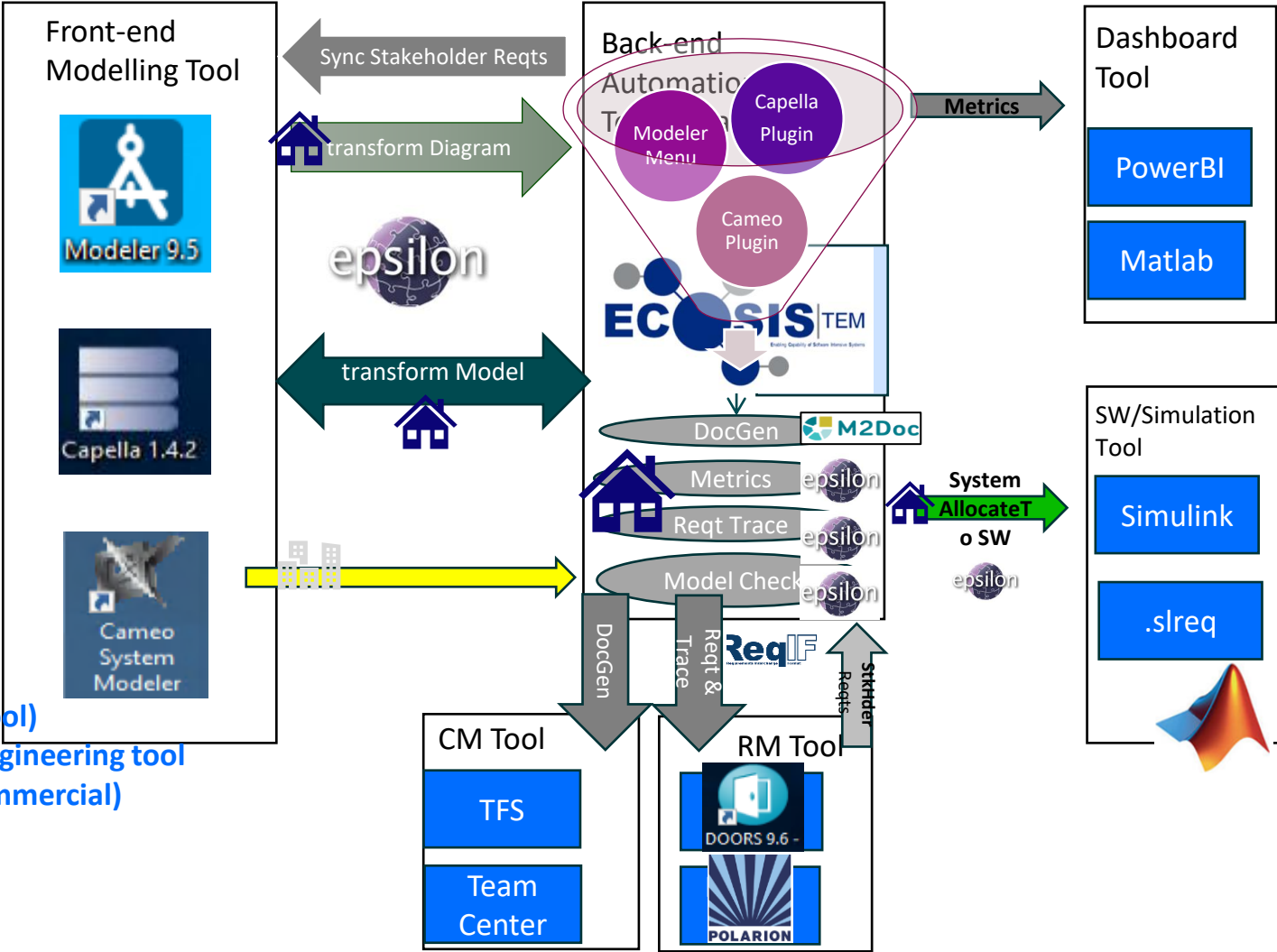


Back-end Service from Front-end

- RR In-house Dev based on Open-Source
- Commercial Off-The Shelf
- Out-source to Universities

Hide Back-end from users
Reduce training efforts (new tool)
Keep artefacts in its original engineering tool
Interconnection (externals, commercial)

Non-Confidential ©2022 Rolls-Royce
Not Subject to Export Control



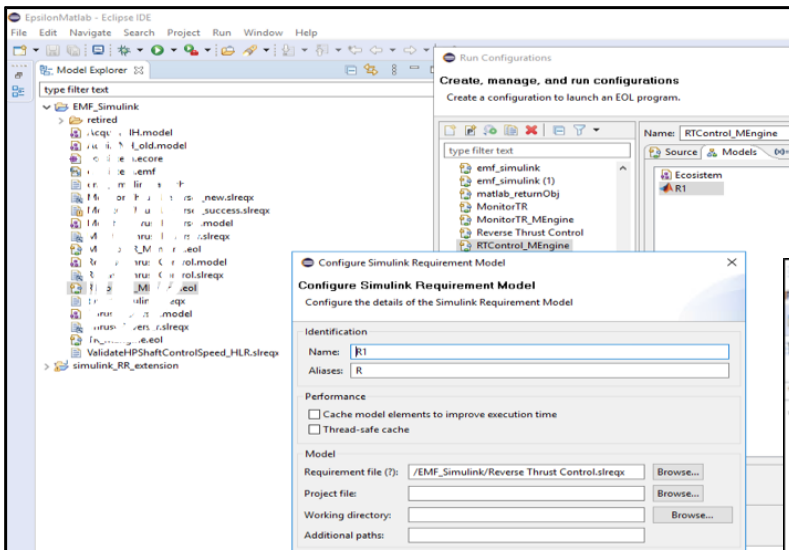


XMI Issue

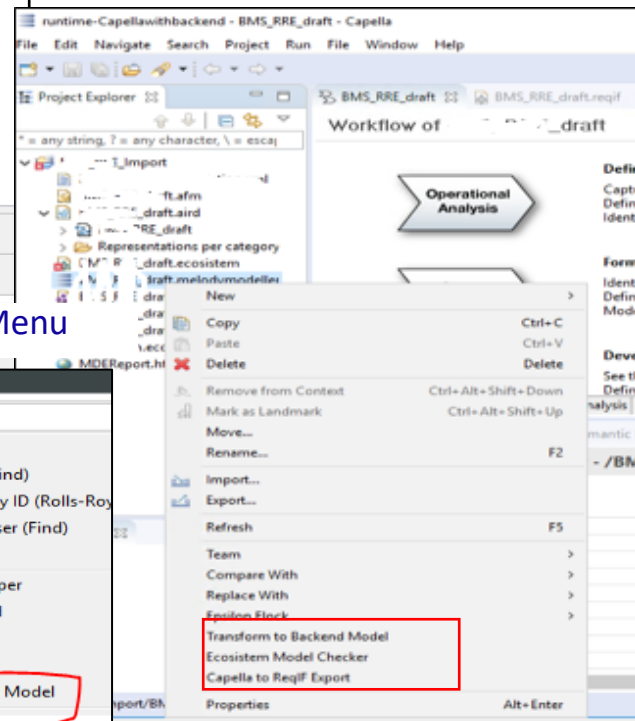
- Performance of XMI export from Artisan (and holistic model-wide nature) (around 8hrs)
- Import of XMI into other modelling tools is also quite slow
- Information lost (may due to heavy SysML Profile extension)
- Probably need to set up as overnight CI job, but then that limits usability (eg. long turn around time for users to do model checks etc)
- Reliance on external developer for future tool developments and releases



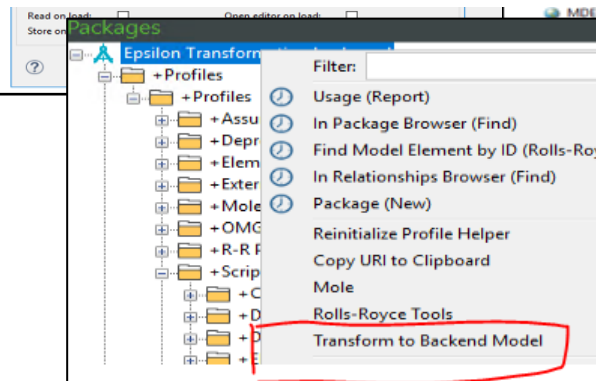
Original UI- trigger from Eclipse with Java project Settings



What does it mean to End-users?



Improved UI- trigger from Capella/IM as Context Menu





Transform to Backend Model via Context Menu

Using Universal Meta-model to Recognise Project/Domain Specific Model

1. Diagram Export during transformation consumes longer time (1-5min)
2. Transformation without diagram should be in few seconds
3. Based on Model Change to Run Transformation (vs Transformation Embedded Method)
4. Once Transformed, can keep being used for: DocGen, Model Check, RM Sync etc.

The screenshot shows the Capella IDE interface. The Project Explorer on the left displays the project structure, including files like 'draft.validation.xml', 'draft.afm', 'draft.aird', 'draft', 'Representations per category', 'draft.ecosystem', 'draft.melodymodeller', 'draft.model', 'draft.model.bak', 'draft.reqif', 'ecosystem.ecore', and 'MDEReport.html'. The main editor shows the 'Deployment' diagram with various function blocks. The Properties panel on the right shows the details of the selected 'Function Store Energy' block.

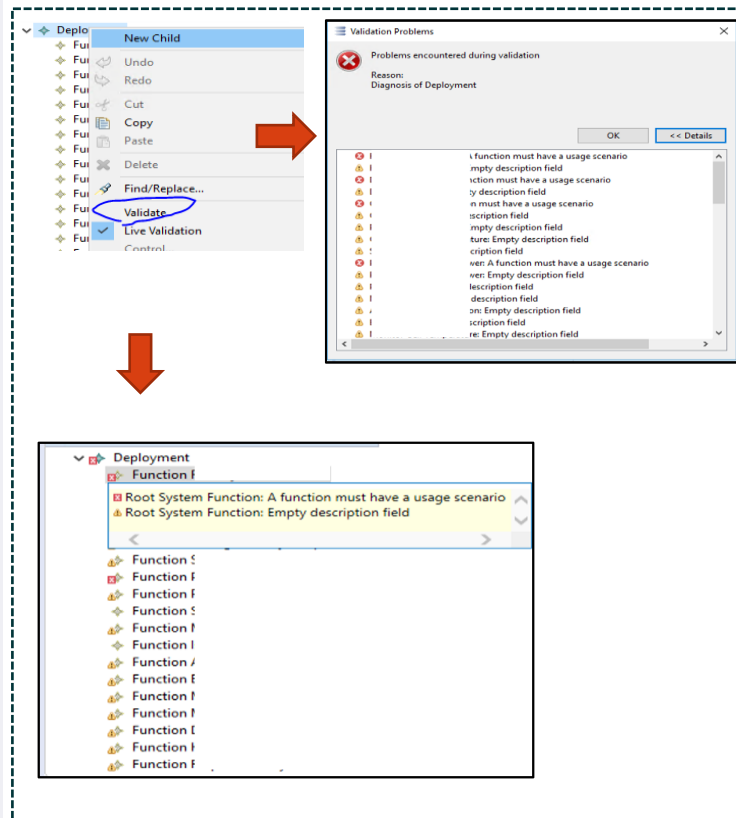
Property	Value
Artisan Identifier	
Baseline Png File Path	file:H:\CapellaImageTemp\Store_Energy.png
Description	
Function Group	Function Group Charge Battery, Function Group
Live Png File Path	
Name	Store Energy
Usage Scenarios	Usage Scenario Balance Cells, Usage Scenario Co



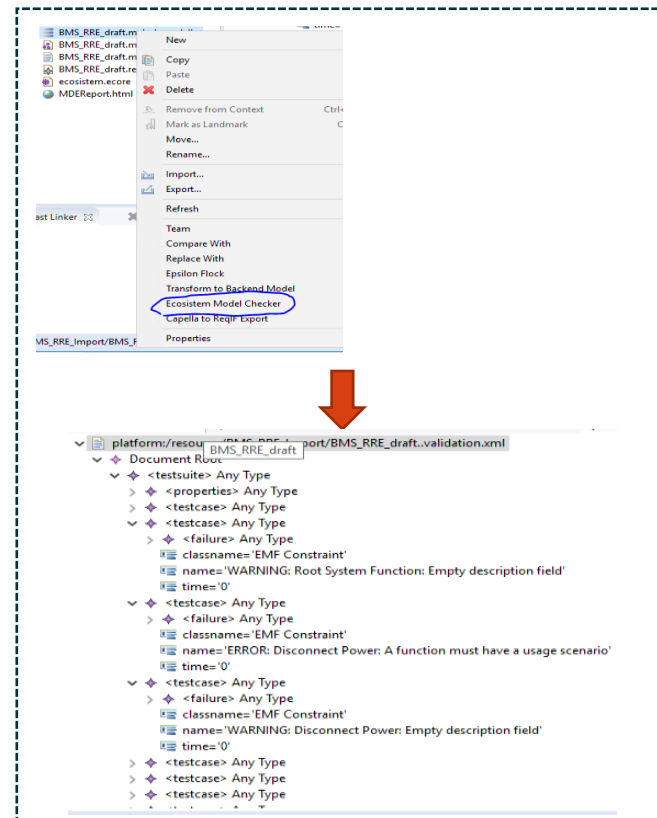
1. Precondition: Transformed Back-end Model Exists
2. Option to show results on Back-end Model
3. Option to show results as XML export (TFS Pull Request/PowerBI)

Menu for Capella Model Check

Show Model Check Results on Back-end Model



XML Export





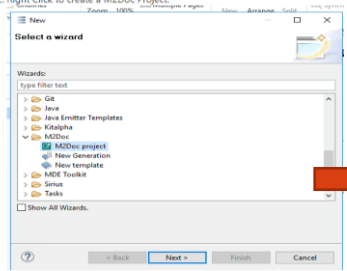
1. Precondition: Transformed Back-end Model Exists
2. Challenge is enabling call extended Epsilon via Capella context menu rather than M2Doc side

DocGen via M2Doc

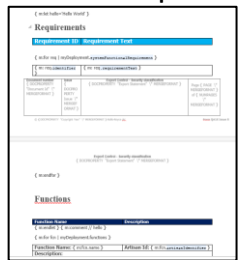
The same process to use M2Doc

Generated Document

2. Right Click to create a M2Doc Project:



Load Template



File Explorer showing project structure:

- BMS_RRE_Import
 - BMS_RRE_draft.validation.xml
 - BMS_RRE_draft.afm
 - BMS_RRE_draft.asd
 - BMS_RRE_draft.ecosistem
 - BMS_RRE_draft.melodymodeller
 - BMS_RRE_draft.model
 - BMS_RRE_draft.model.bak
 - BMS_RRE_draft.reqif
 - BMS_RRE_draft.result.docx
 - Capella_DocGen.conf
 - ecosistem.ecore
 - MDEReport.html
 - Capella_DocGen
 - ArtisanModel2.docx
 - Capella_DocGen.docx

Context menu for Capella_DocGen.docx:

- Open
- Open With...
- Copy
- Paste
- Delete
- Remove from Context
- Mark as Landmark
- Move...
- Rename...
- Import...
- Export...
- Refresh
- Team
- Compare With...
- Replace With...
- Epsilon Flow
- Generate Documentation
- Validate Documentation Template
- Edit Documentation Configurations
- Transform to Backend Model
- Ecosystem Model Checker
- Capella to Reqif Export
- Properties

Technical Report

Table of Contents

- Table of Contents
- Template variables
- Generation examples
- Requirements
- Requirement ID: Requirement Test

Template variables

Generation examples

Requirements

Requirement ID: Requirement Test

Functions

Function Name: Description

Function Name: TEST System to Control

Function Name: Change battery

Function Name: TEST System to Control

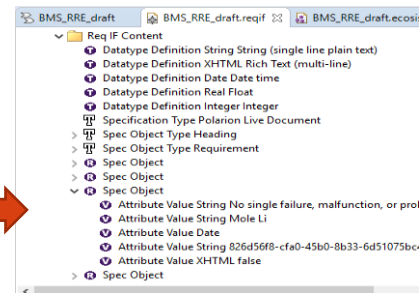


1. Precondition: Transformed Back-end Model Exists
2. Manually Generate an Empty Polarion/DOORS Document (only needed first time/for a separate doc)
(Polarion/DOORS Unique ID cannot be created outside and enables Polarion target document modification rather than duplicated)

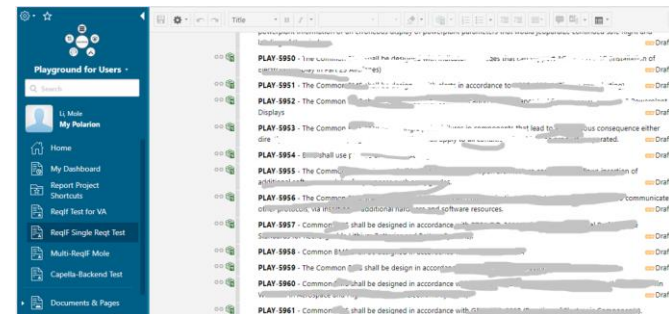
Menu for Capella to Polarion Connection via ReqIF

Create Polarion ReqIF Empty File

ReqIF can be checked in Capella



ReqIF can be imported to Polarion
& Update original Document



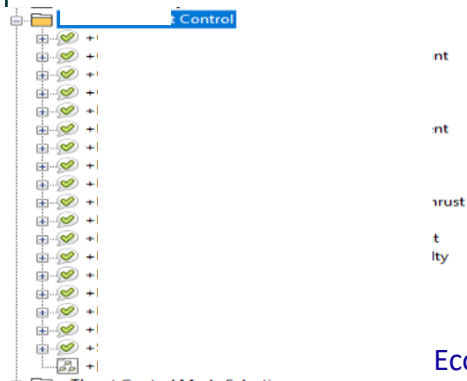


Transformation for simulation and software development

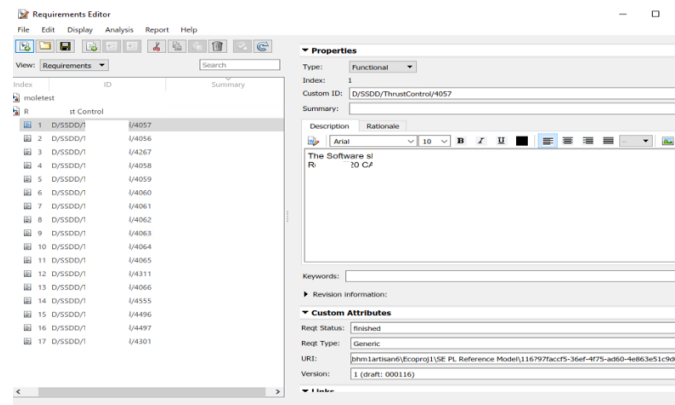
Key Technology Evolvement

1. hierarchical requirements with Text and Rationale transformation
2. Automated custom attributes creation/modification/duplication detection
3. objects/value exchange (especially custom attributes) between EMF/Java and Matlab

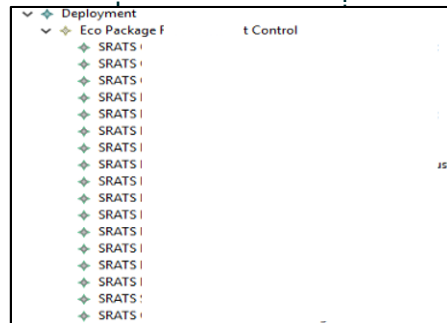
PTC Modeller



Simulink Requirement



Ecosystem Back-end

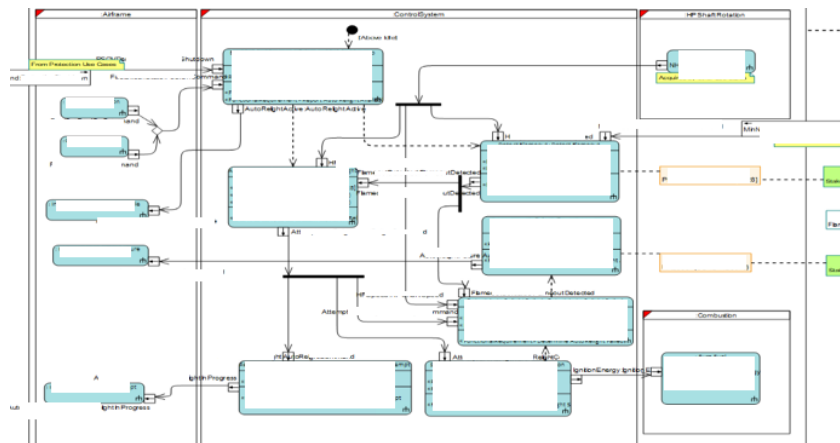




Transformation for Reuse:

PTC Modeller - Capella Transformation via Backend

Non-Confidential ©2022 Rolls-Royce
Not Subject to Export Control



Note:

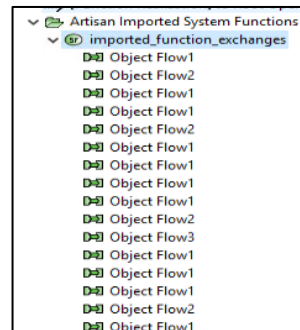
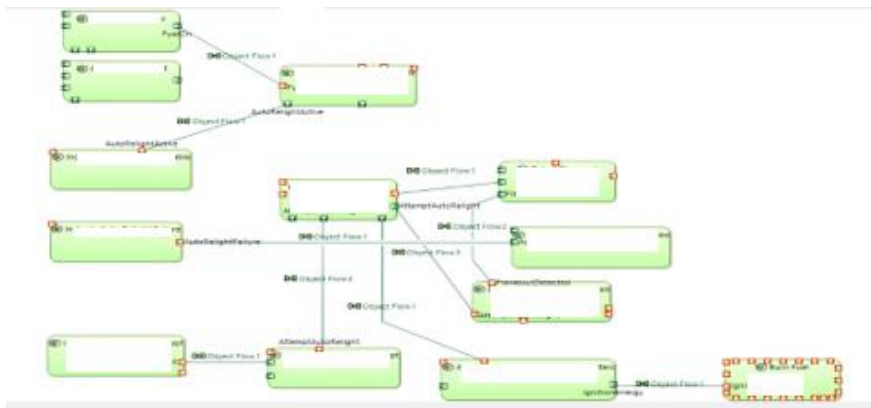
1. Only Object Flow transformed
no Control Flow

2. Join/Fork/Decision/Merge
nodes are simplified

3. Noticeable name differenced
due to System Function is named
by SSRD rather than Top Level
Function

4. Omitted Context Functions

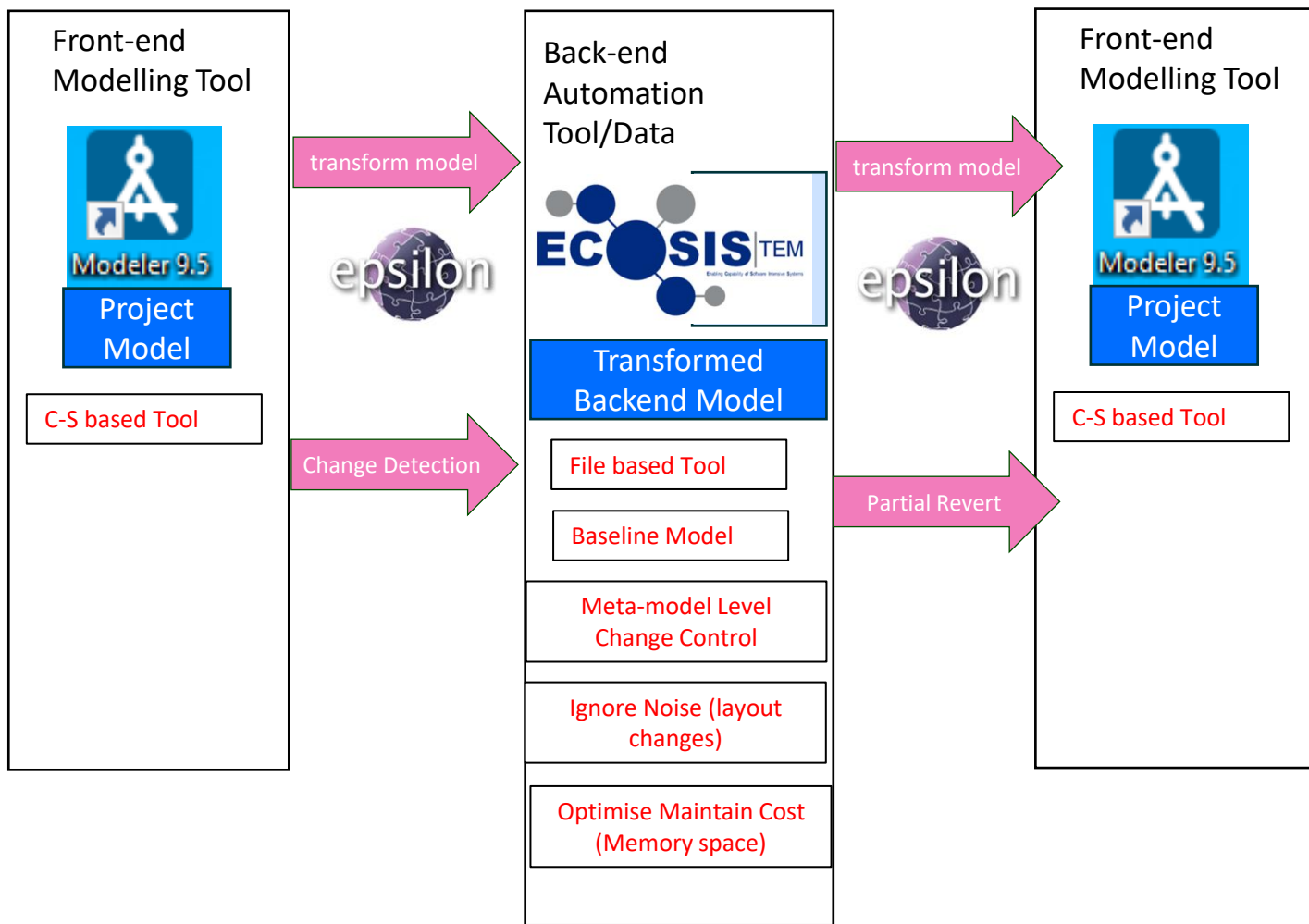
5. Functional Exchange contained
by grouping SF





Enabling and Optimise Change Control and Configure Management:

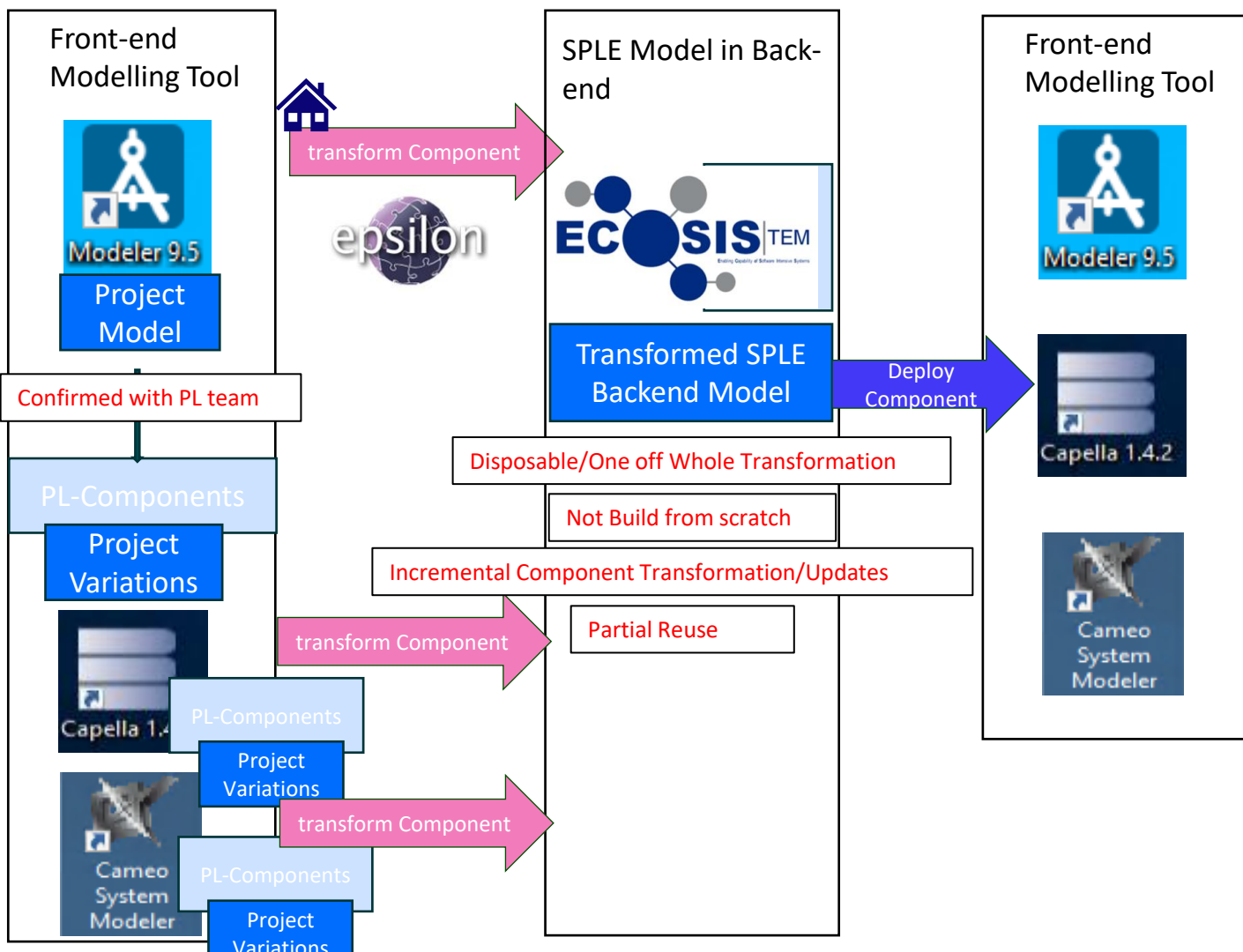
- **Component-based Transformation**





SPLE

- Component-based Transformation
- Cross Tool SPLE Deployment
- Configurable Library Updates





Thank You

Non-Confidential ©2022 Rolls-Royce
Not Subject to Export Control