PARAMETRICO E TRADES

2025 – Preparado por Prof. Dr. Christopher Shneider Cerqueira

WEEK	CLASS ACTIVITY	REF	INDIVIDUAL	W	GROUP	W
1 28/Jul	Course Structure and Initial Definitions Systems Engineering Review	[illegible]	IA-01 - Reading and [illegible] Questions (10)	10%		0%
2 04/Aug	Classical Systems Engineering Diagrams (IDEF-0/N2/eFFBD/DFD)	[4] *papers	IA-02	0%	GA-02 - Prepare a representation of your system using classical Diagrams	50%
3 11/Aug	Transition from Legacy to MBSE MBSE Methodologies MBSE Languages	[5] [7] *[illegible]	IA-03 - Reading and [illegible] Questions (10)	10%		0%
4 18/Aug	OPM - Basic	[6]	IA-04 - Exercises	10%		0%
5 25/Aug	OPM - Extended	[6]	IA-05 - Exercises	10%		0%
6 01/Sep	OPM - Group Presentation		IA-06	0%	G6 - Prepare a presentation of your system using OPM	50%
7 08/Sep	SysML Introduction (bdd/ibd)	[7]	IA-07 - Exercises	10%		0%
8 15/Sep	P1 - Conceptual Questions and Case	[1] [2] [3] [4] [6]	IA-08 - Questions and a mini-case	50%	GA-08 -	
				100%		100%

WEEK	CLASS ACTIVITY	REF	INDIVIDUAL	W	GROUP	W
9 29/Sep	SysML (act/stm)	[7]		10%	GA-09 -	0%
10 06/Oct	SysML (seq/uc)	[7]	IA-10 - Exercises	10%	GA-10 -	0%
11 13/Oct	Simulation on SysML		IA-11 -	0%	GA-11 -	0%
12 20/Oct 21/Oct	SysML (pkg/req)	[7]	IA-12 - Exercises	10%	GA-12 -	0%
13 27/Oct	Arcadia process applied into the SysML	[5]	IA-13 -	0%	GA-13 -	0%
14 03/Nov	Some System Analysis on SysML SysML V2 Perspectives	[8]	IA-14 -	0%	GA-14 -	0%
15 10/Nov	SysML Group Presentation Course Ending		IA-15 -	0%	GA-15 - Prepare a presentation of your system using SysML	100%
16 17/Nov	P2 - Conceptual Questions and Case	[5] [7]	IA-16 - Questions and a mini-case	70%	GA-16 -	
				100%		100%
EXAM 24/Nov 08/Dec	**If necessary: Writing an article (min 6pgs / max 10pgs) reporting the case of their group in the SIGE standard.**					**100%**

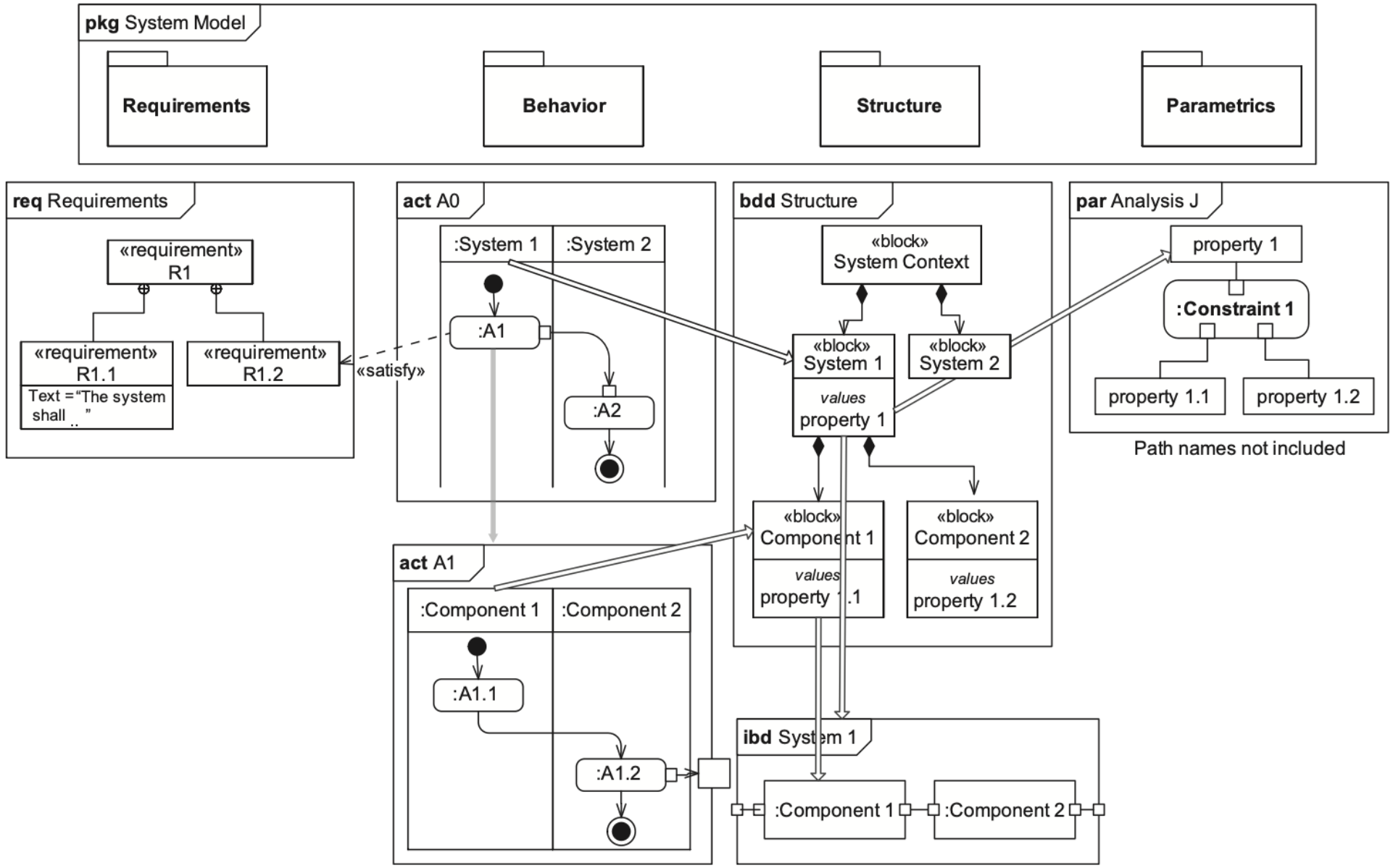


FIGURE 3.3

Simplified diagrams highlighting some of the language features for each kind of diagram in SysML-Lite.

Podemos iniciar com uma visão geral do contexto do sistema

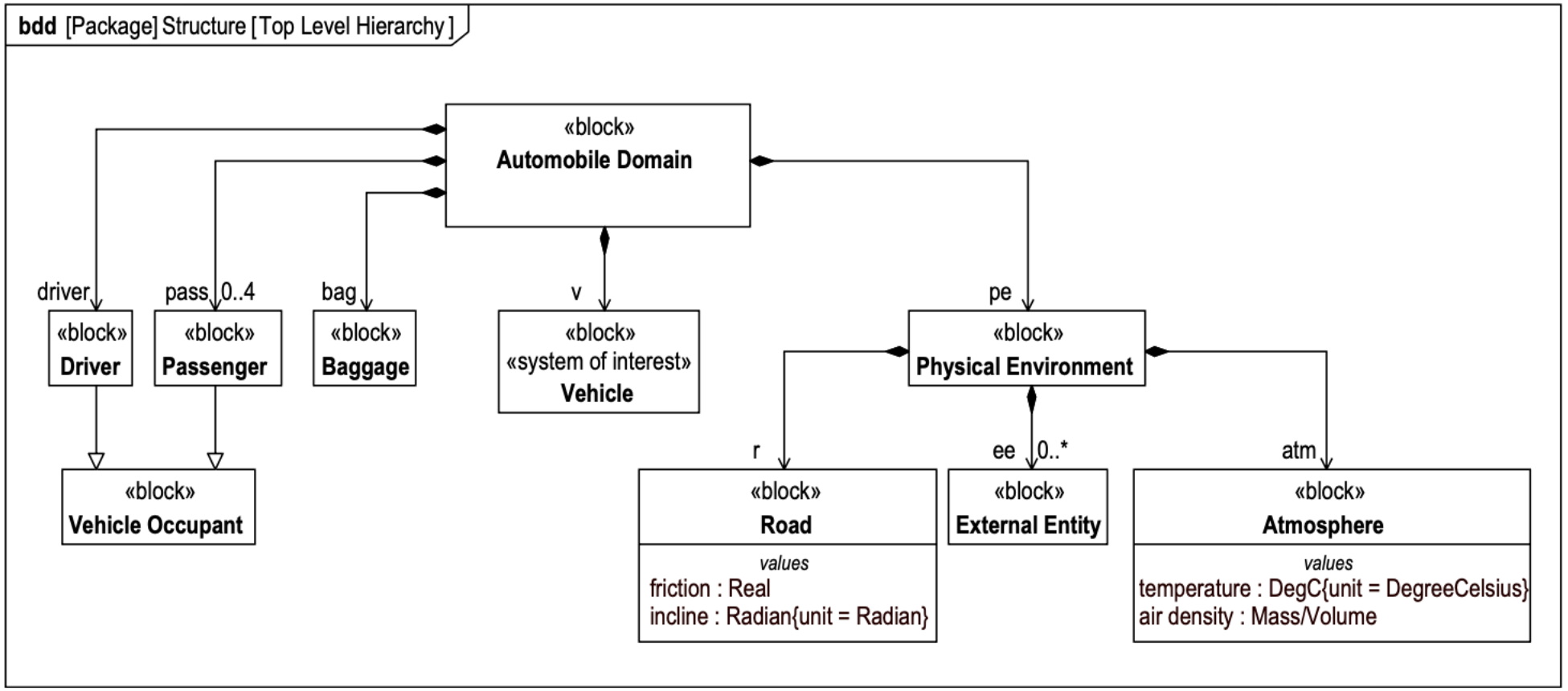


FIGURE 4.3

Block definition diagram of the *Automobile Domain* showing the *Vehicle* as the *system of interest*, along with the *Vehicle Occupants* and the Environment. Selected value properties for the *Road* and *Atmosphere* are also shown.

Estudar a interconexão em alto nível

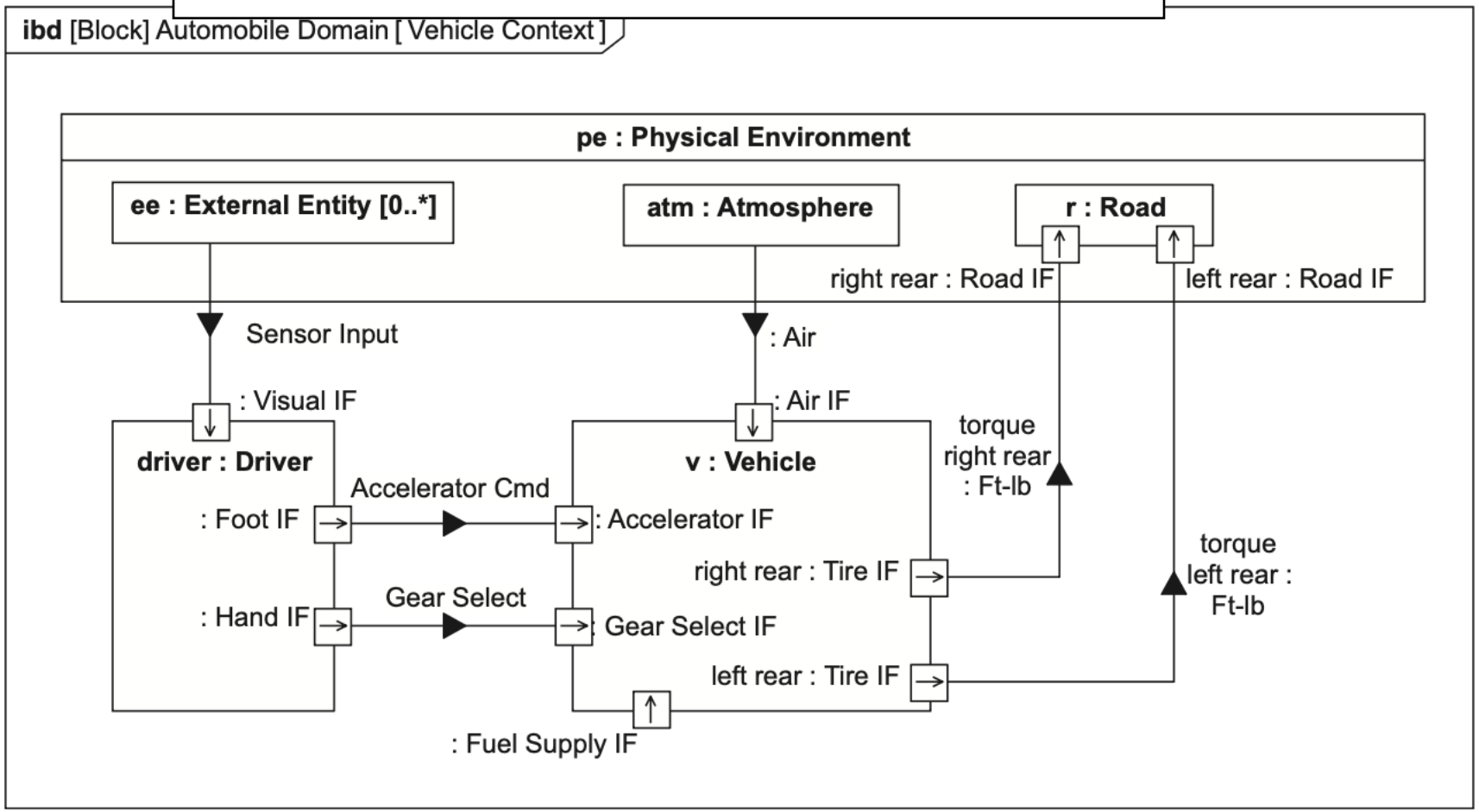


FIGURE 4.9

The internal block diagram for the *Automobile Domain* describes the *Vehicle Context*, which shows the *Vehicle* and its external interfaces with the *Driver* and the *Physical Environment* that were defined in Figure 4.3.

Descrever a sequencia funcional dos elementos principais

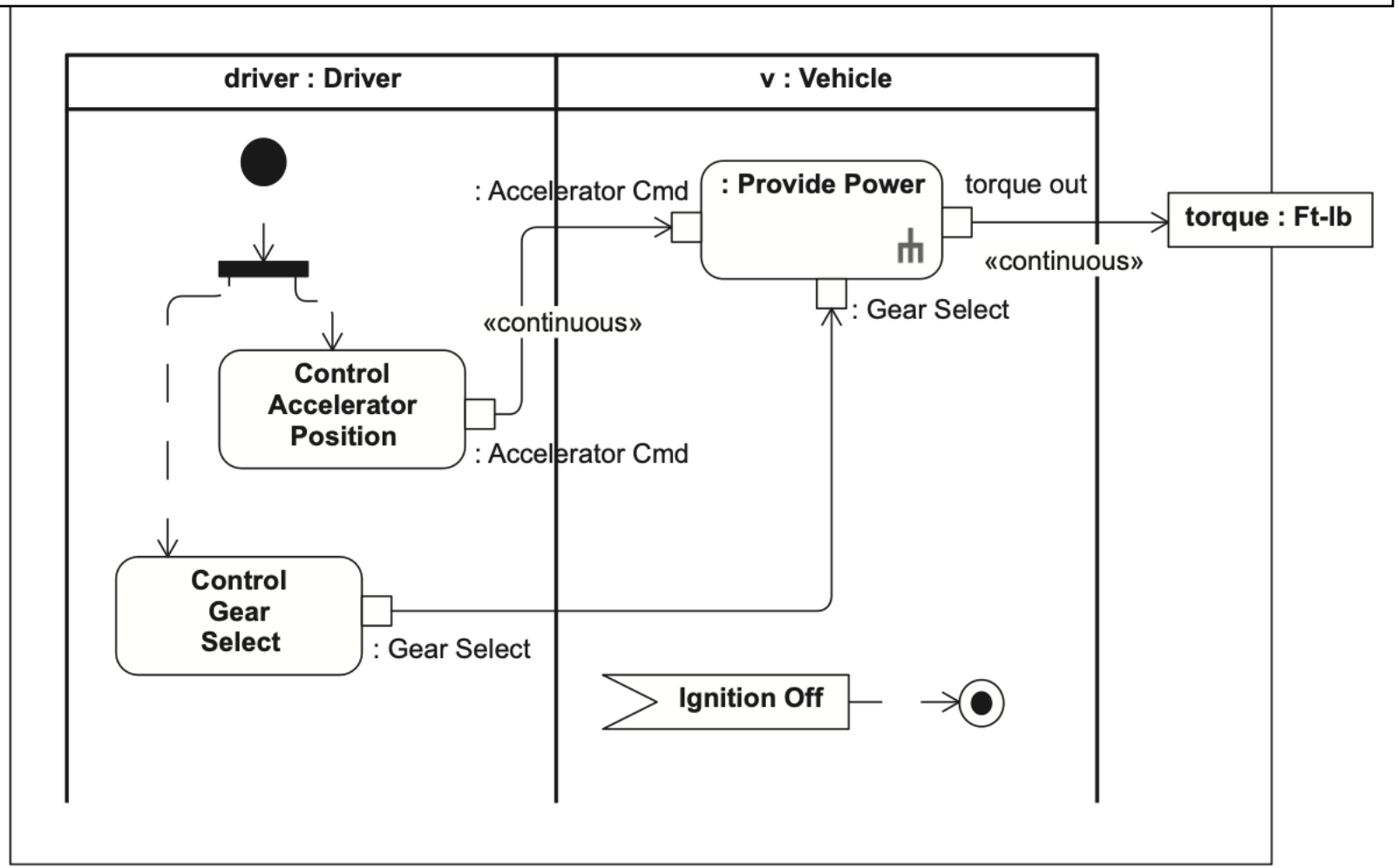
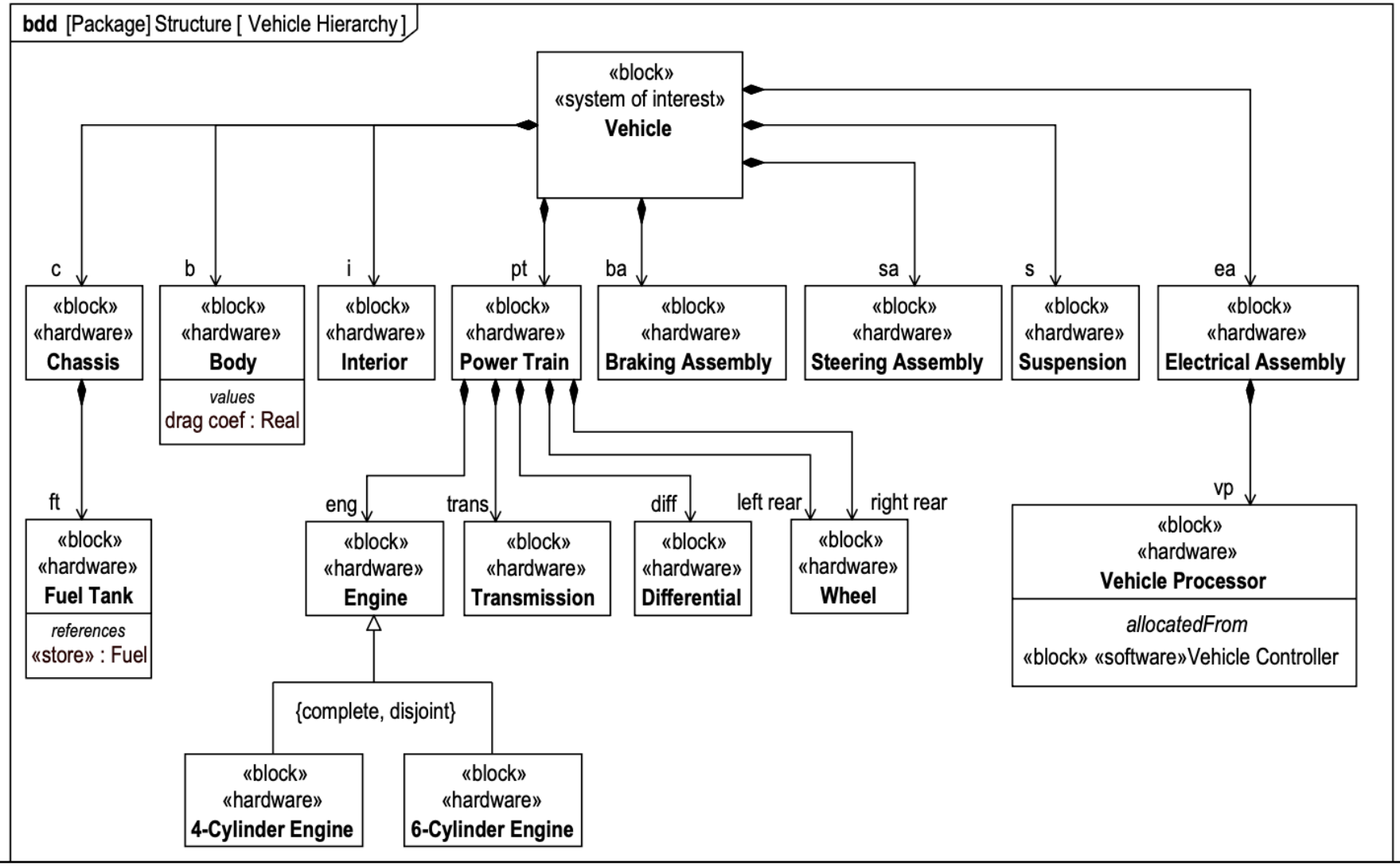


FIGURE 4.7

Activity diagram allocated from the *Control Neutral*, *Forward*, and *Reverse Power* interaction uses that are referenced in the *Drive Vehicle* sequence diagram in Figure 4.5. It shows the continuous *Accelerator Cmd* input and the *Gear Select* input from the *Driver* to the *Provide Power* action that the *Vehicle* must perform.



mas.... depois de um tempo é desejável fazer a decomposição do SuD

A block definition diagram of the *Vehicle Hierarchy* that shows the *Vehicle* and its components. The *Power Train* is further decomposed into its components, and the *Vehicle Processor* includes the *Vehicle Controller* software.

E as relações internas dos SuD

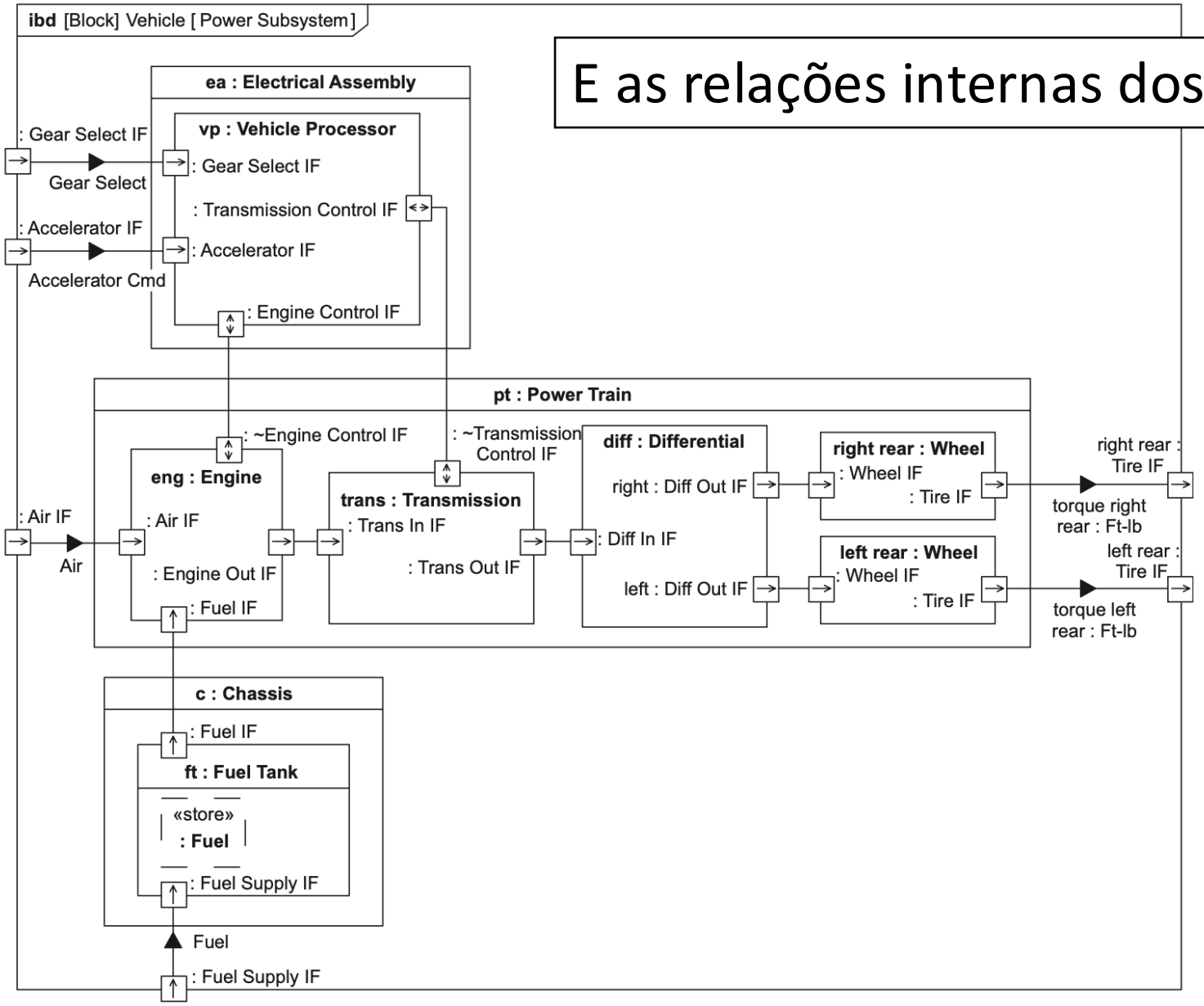
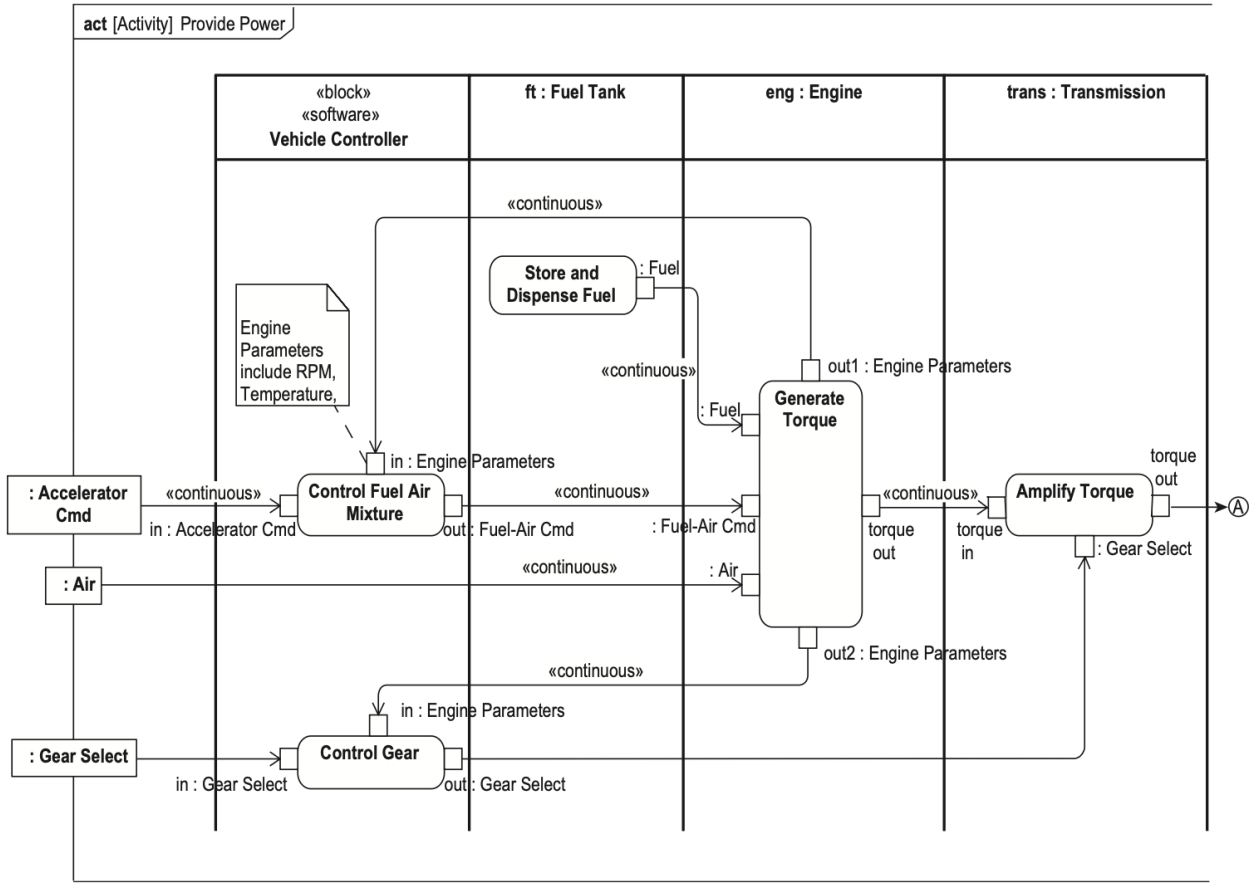


FIGURE 4.12

The internal block diagram for the *Power Subsystem* shows how the parts of the *Vehicle* that *Provide Power* are interconnected. The parts interact as specified by the activity diagram in Figure 4.11.

e

E descrever a arquitetura funcional no nível de subsistemas...



GURE 4.11 Cont'd

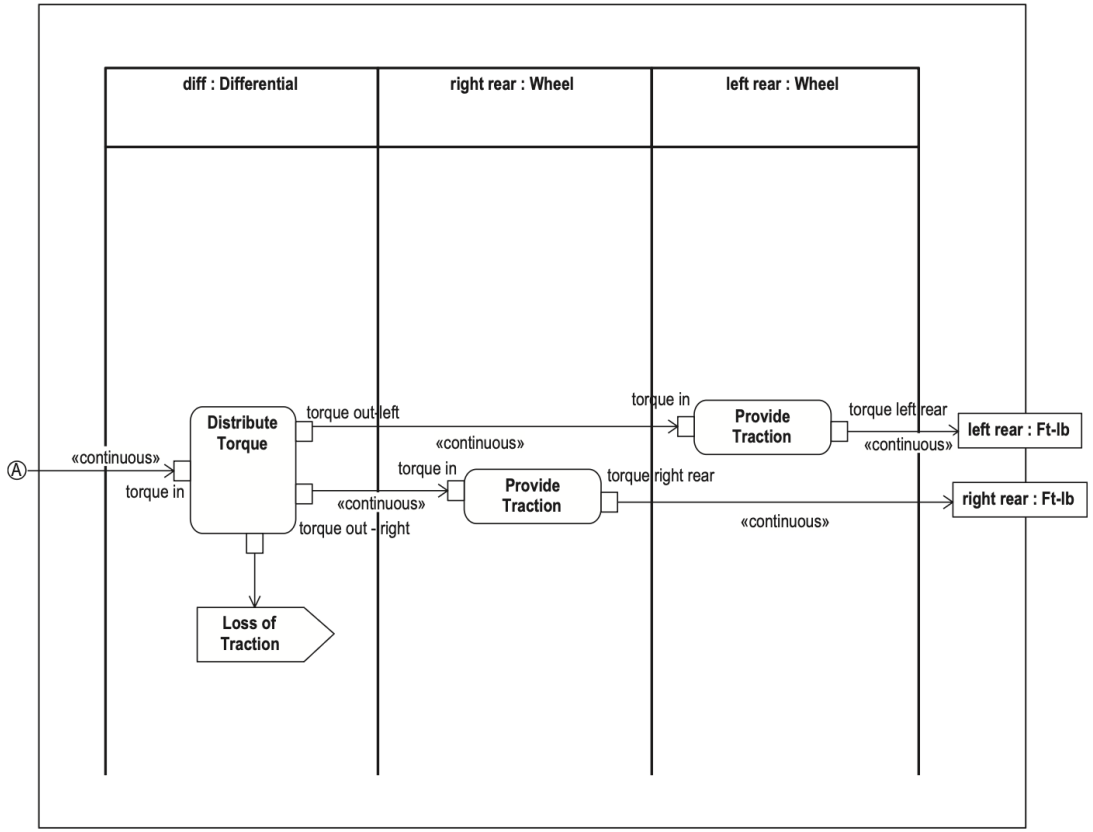


FIGURE 4.11

The activity diagram for *Provide Power* shows how the *Vehicle* components generate the torque to move the vehicle. This activity diagram realizes the *Provide Power* action in Figure 4.7 with activity partitions that correspond to the components in Figure 4.10.

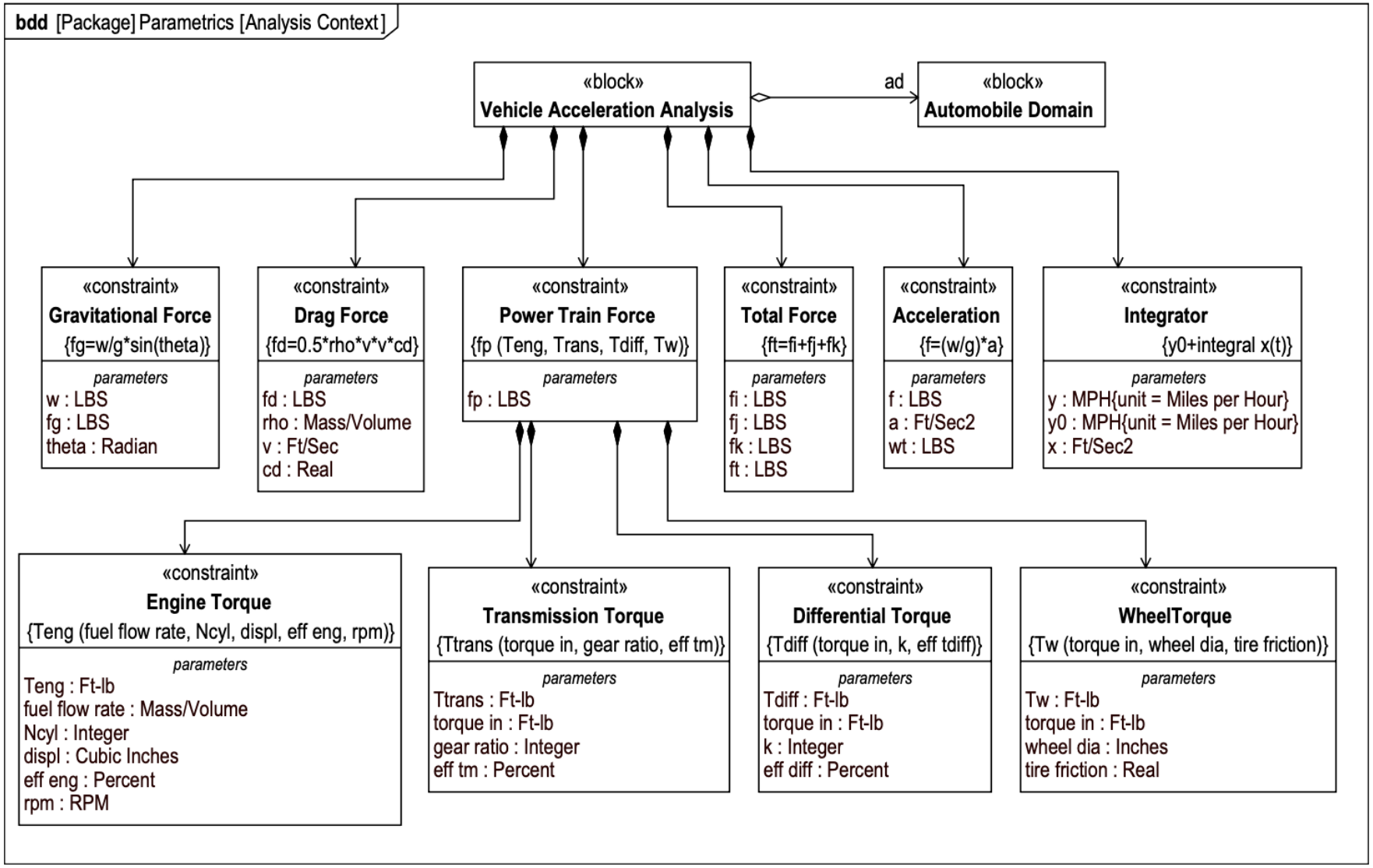


FIGURE 4.13

mas.... agora... gostaria de descrever os parâmetros do meu SuD

Automobile Domain block from Figure 4.3 is referenced since it is the subject of the analysis.

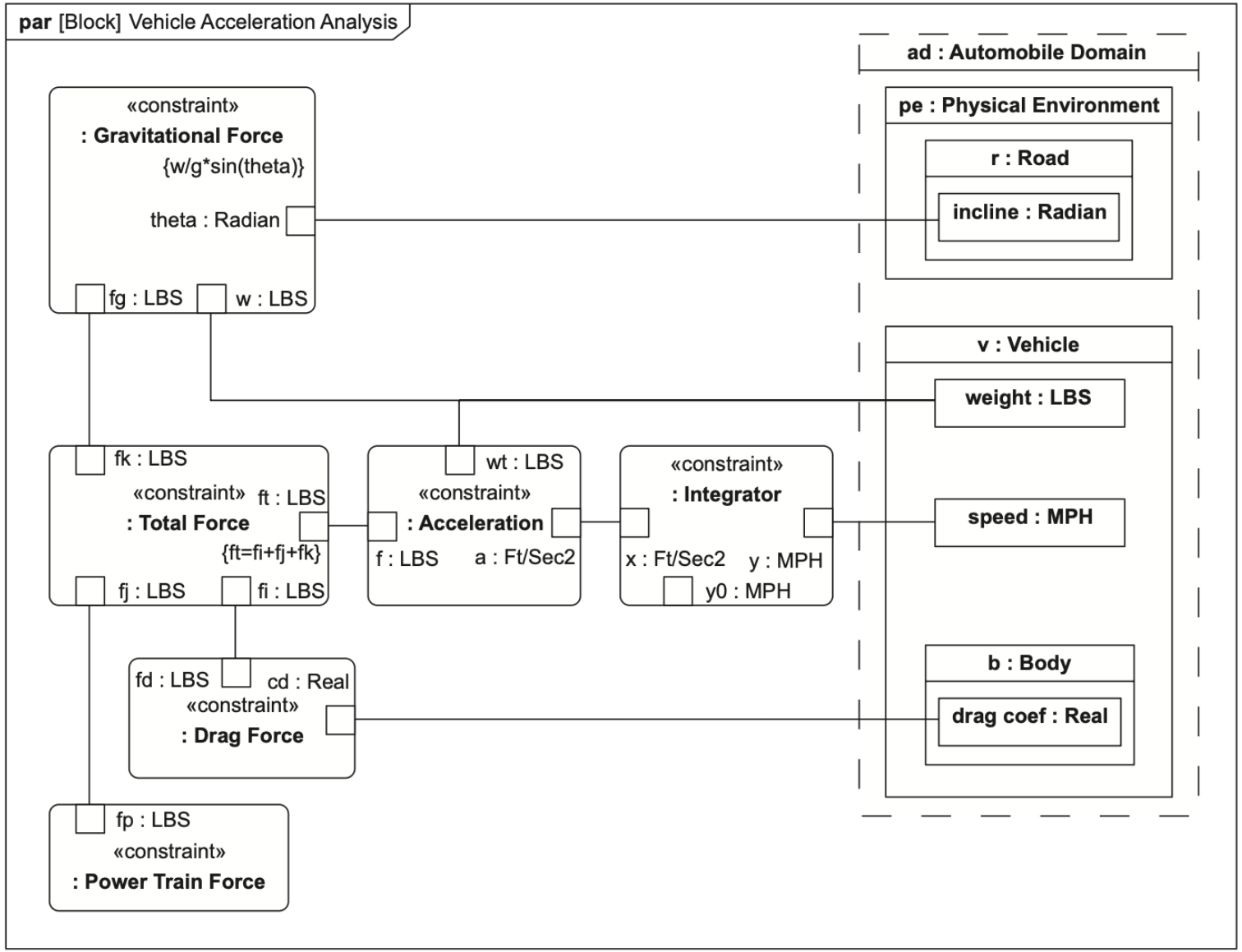


FIGURE 4.14

E descrever todo o equacionamento do SuD (no nível desejado)

Bem... vamos caminhar nos diagramas:

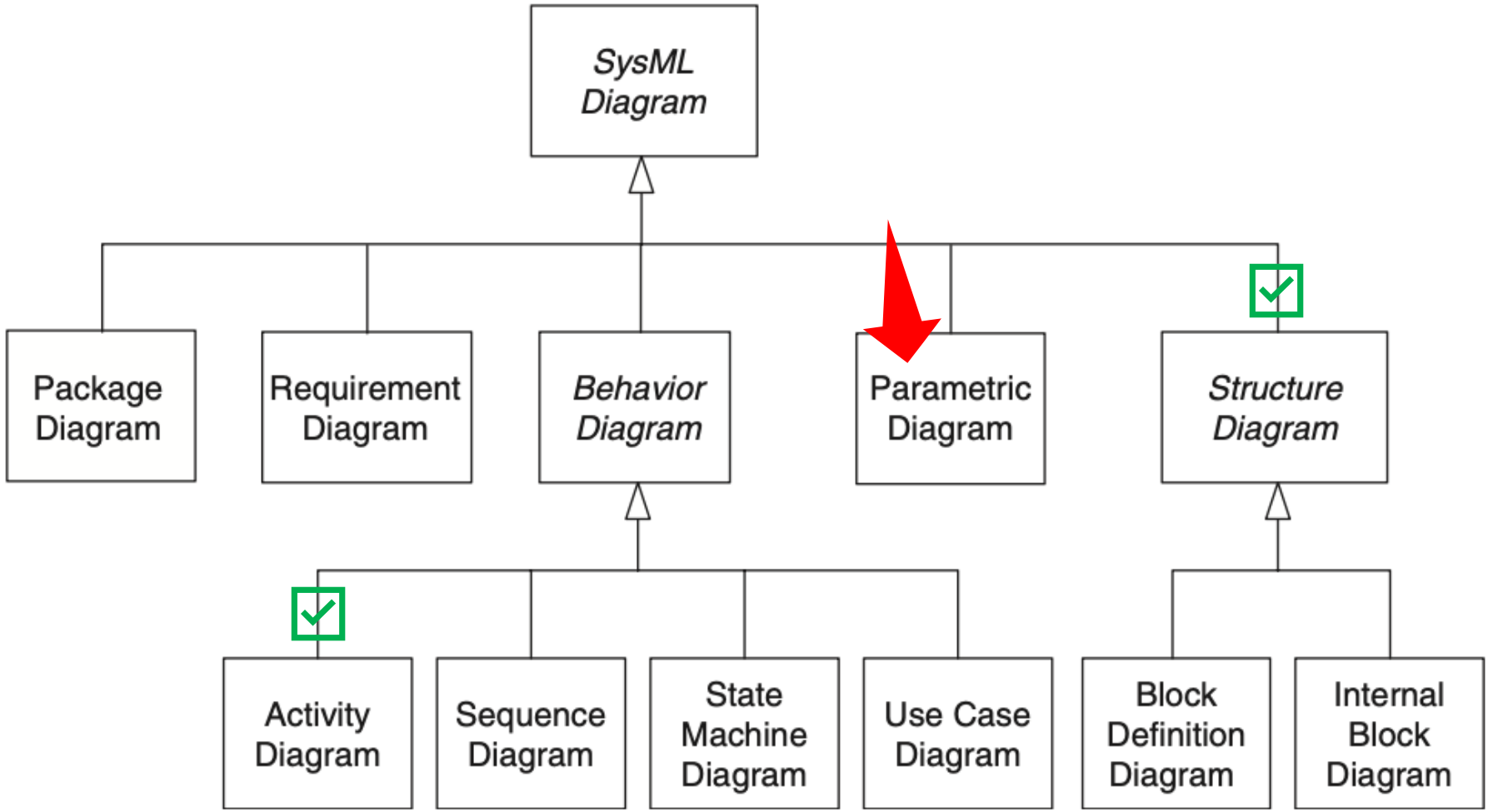


FIGURE 3.1

SysML diagram taxonomy.

Modelagem Paramétrica de Restrições (*Constraints*)

Capítulo 8

Introdução

- Um esforço de projeto típico **inclui a necessidade de realizar tipos diferentes de análises** de engenharia, como análise de budget, análise de sensibilidade e otimização de projeto.
- Pode incluir a análise de desempenho, confiabilidade, custo e propriedades do sistema em consideração.
- O SysML suporta esse tipo de análise por meio do uso de modelos paramétricos.

Intro: Parametros são propriedades/restrições

- Os modelos paramétricos restringem as propriedades de um sistema, **que podem ser avaliadas por uma ferramenta de análise apropriada**.
- As restrições **são expressas como equações**, com os **parâmetros das equações sendo vinculados às propriedades do sistema** que está sendo analisado.

Intro: **Constraint Block**

- O SysML introduz o **bloco de restrição** para dar suporte à construção de modelos paramétricos.
- Um bloco de restrição é um **tipo especial de bloco** usado para **definir equações** para que possam ser reutilizadas e interconectadas.
- Os blocos de restrição têm dois recursos principais: um **conjunto de parâmetros e uma expressão**.
  - Que restringe esses parâmetros.
- Os blocos de restrição seguem o **padrão de definição e uso que se aplica a blocos** e partes.

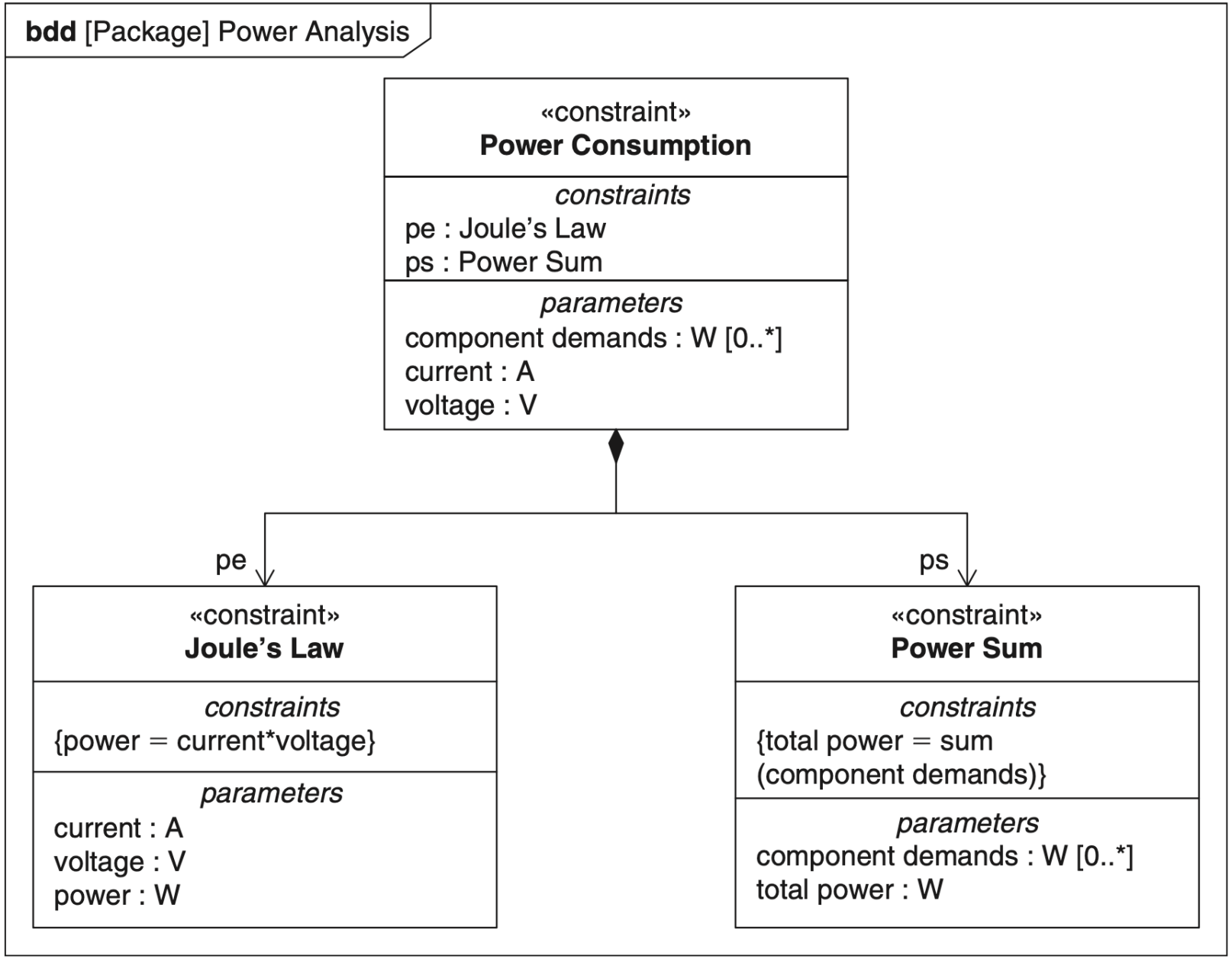


FIGURE 8.1

Example block definition diagram with constraint blocks.

Parametric Diagram

- Os diagramas paramétricos são usados para criar sistemas de equações que podem restringir as propriedades dos blocos.
- O cabeçalho completo para um diagrama paramétrico é o seguinte:

- *par [model element kind] model element name [diagram name]*

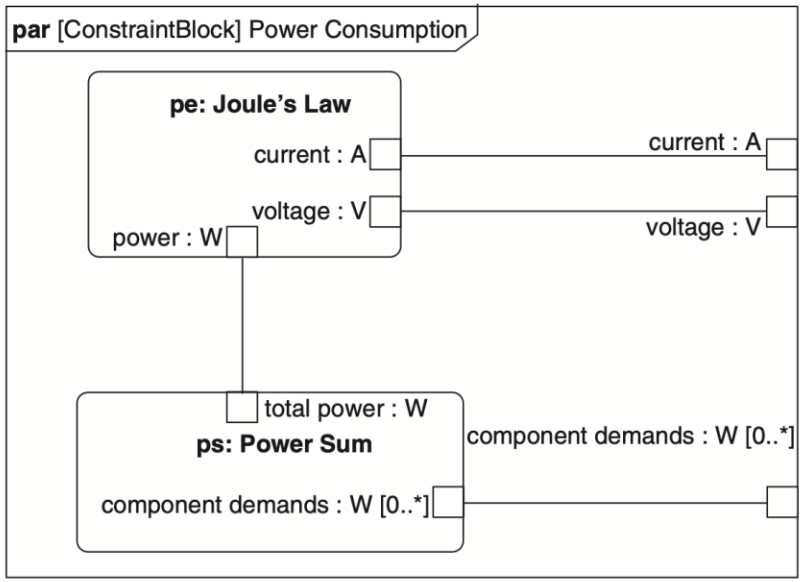


FIGURE 8.2

A parametric diagram used to construct systems of equations.

Conteúdo de uma restrição (constraints)

- O SysML inclui um **mecanismo genérico para expressar restrições** em um sistema como *expressões de texto* que podem ser aplicadas a qualquer elemento de modelo.
- SysML **não fornece uma linguagem** de restrição interna.
- A **definição de restrição deve incluir a linguagem** usada para permitir que a restrição seja avaliada.
  - *Uma restrição também pode ser mostrada como um símbolo de nota anexado ao(s) elemento(s) do modelo que ela restringe, com o texto da restrição mostrado no corpo da nota.*

Blocos... Reusando restrições

- O SysML também inclui um bloco de restrição que estende o conceito de restrição genérica.
- Um bloco de restrição **encapsula uma restrição para permitir que ela seja definida uma vez e usada em diferentes contextos**, semelhante à maneira como as partes representam os usos de blocos em diferentes contextos.
- O conceito equivalente à parte é chamado de propriedade de restrição.
  - A **expressão de restrição** de um bloco de restrição **pode ser qualquer expressão matemática e pode ter uma dependência explícita do tempo**. Além da expressão de restrição,
  - Um bloco de restrição define um conjunto de **parâmetros de restrição** — um tipo especial de propriedade usada na expressão de restrição. Os parâmetros de restrição podem ser vinculados a outros parâmetros e às propriedades dos blocos.

- Um bloco de restrição é definido em um diagrama de definição de bloco (bdd)
- O compartimento de nome do bloco de restrição inclui a palavra-chave «constraint» acima do nome para diferenciá-lo de outros elementos em um bdd.
- A expressão de restrição é definida no compartimento de restrições do bloco de restrição e os parâmetros de restrição são definidos no compartimento de parâmetros usando o seguinte formato:

parameter name: type[multiplicity]

Usando composição

- Os modeladores podem **compor blocos de restrição complexos** a partir de outros blocos de restrição.
- Nesse caso, o bloco de restrição composto **descreve uma equação que vincula os parâmetros de suas restrições filho**.
- Isso permite que equações complexas sejam definidas **reutilizando equações mais simples**.

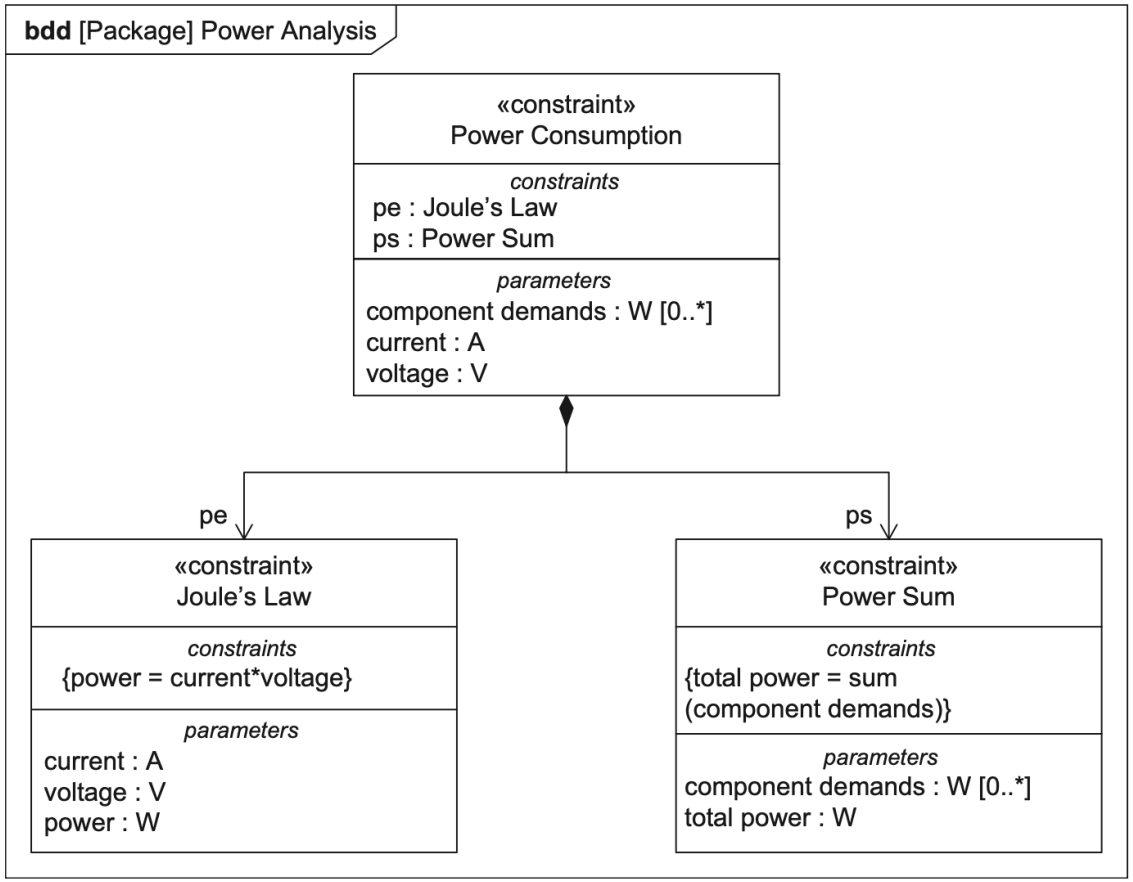


FIGURE 8.5

A hierarchy of constraints on a block definition diagram.

O diagrama de parâmetros exibe a visão “interna” da relação entre restrições/parâmetros

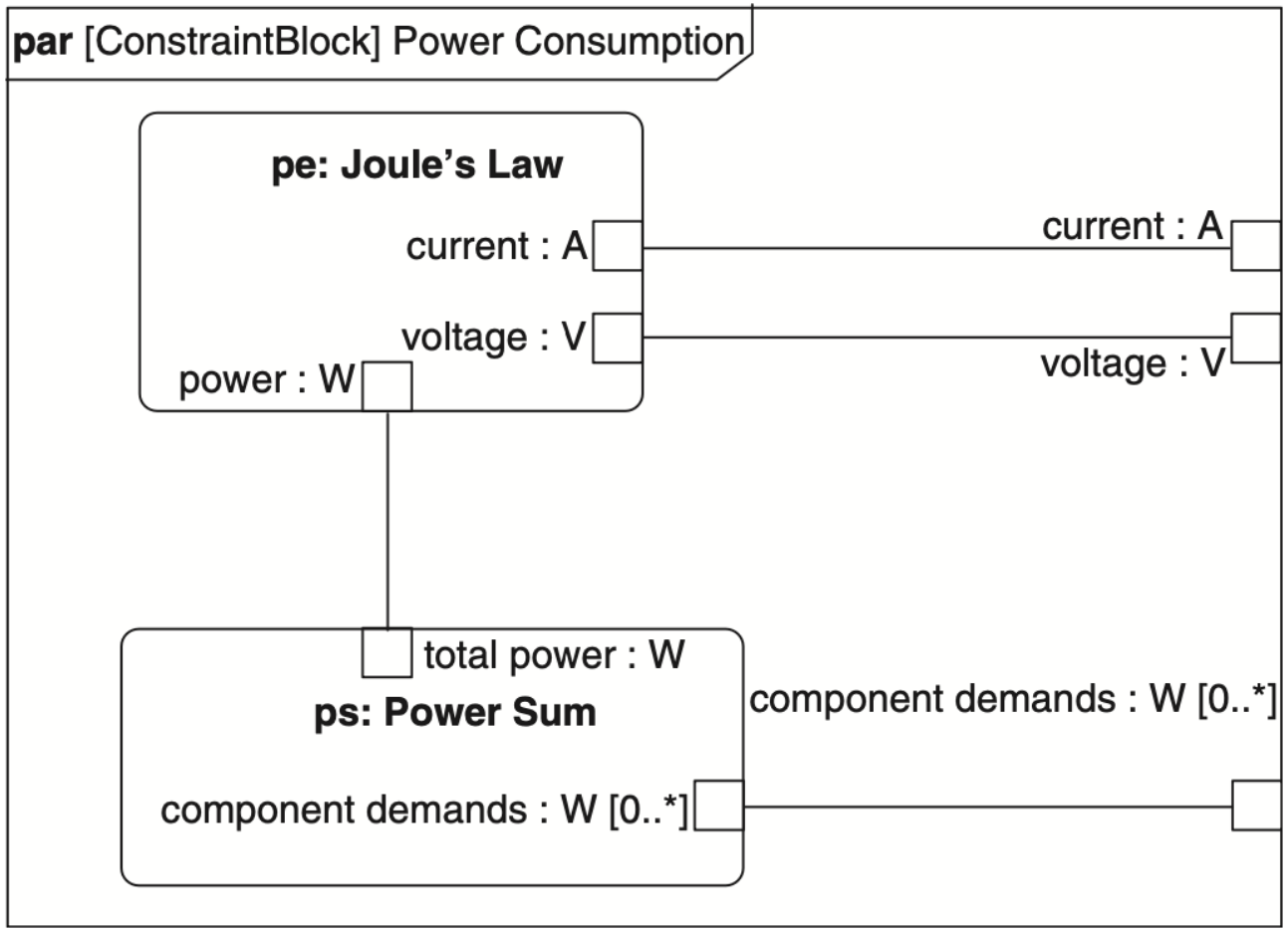


FIGURE 8.6

Internal details of the power consumption equation using a parametric diagram.

Interconexão igual ao IBD

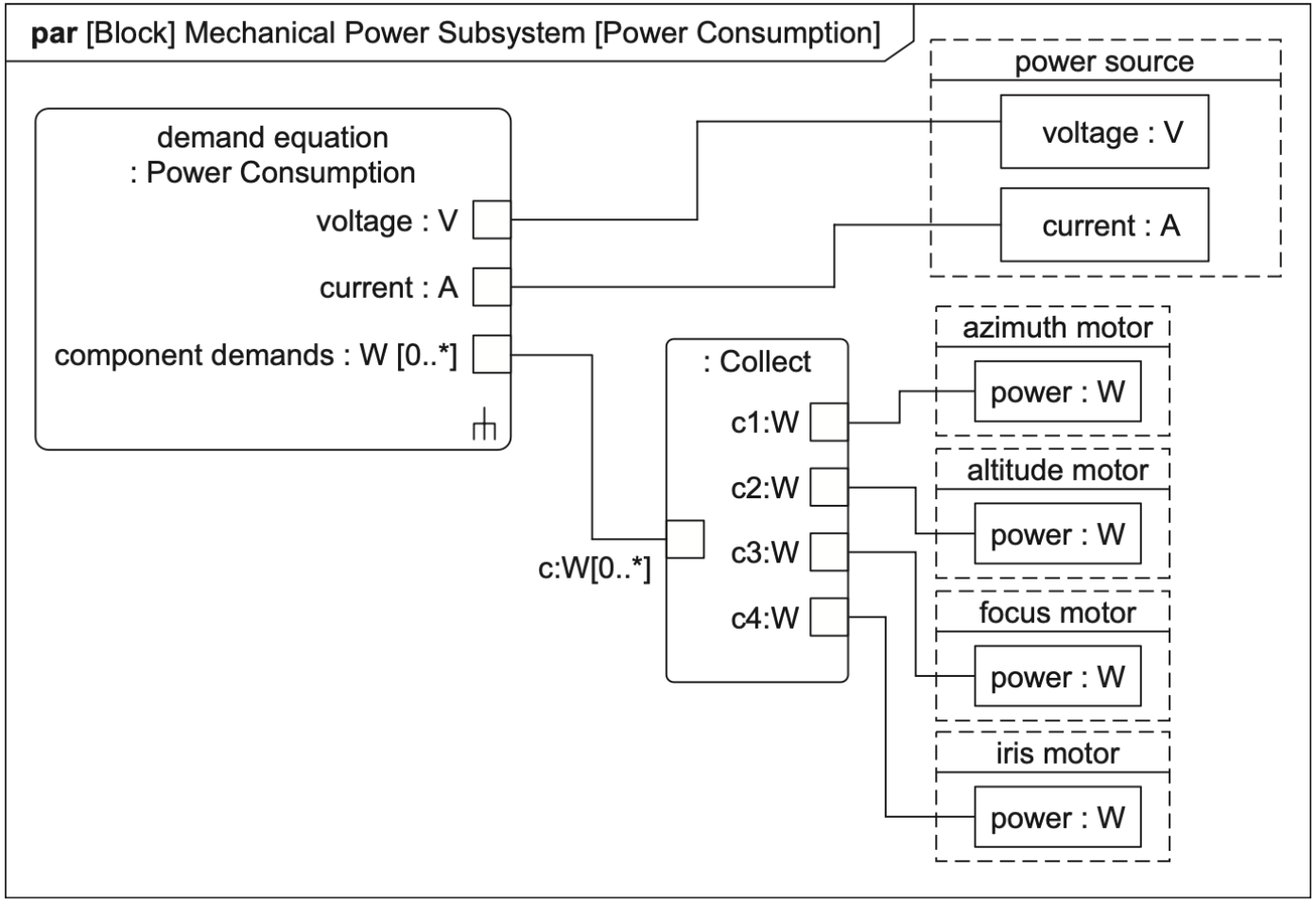


FIGURE 8.7

Binding constraints to properties on a parametric diagram.

Apontando valores

- Para permitir que uma ferramenta de análise avalie blocos contendo propriedades de restrição, pelo menos algumas das propriedades de valor do bloco em análise precisam ter valores específicos definidos.
- Muitas vezes, esses valores são fornecidos durante a análise por meio da interface da ferramenta de análise, mas também podem ser especificados usando uma configuração de bloco.
  - *Isso é feito criando uma especialização do bloco com os valores iniciais necessários ou usando uma especificação de instância para descrever uma instância do bloco*.

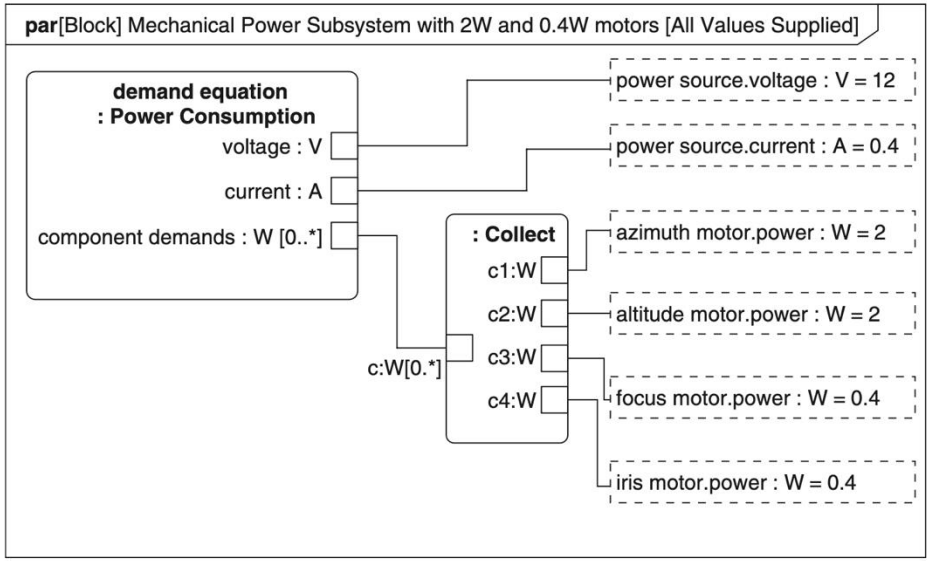


FIGURE 8.8

Describing a specific analysis configuration.

Descrevendo propriedades temporais

- Uma propriedade de valor é frequentemente uma **propriedade variável no tempo** que pode ser restringida por equações diferenciais ordinárias com derivadas de tempo ou outras equações dependentes do tempo:
  - *tratar o tempo como implícito na expressão*
  - *incluir uma propriedade de tempo separada que represente explicitamente o tempo nas equações de restrição*

Tempo em uma expressão

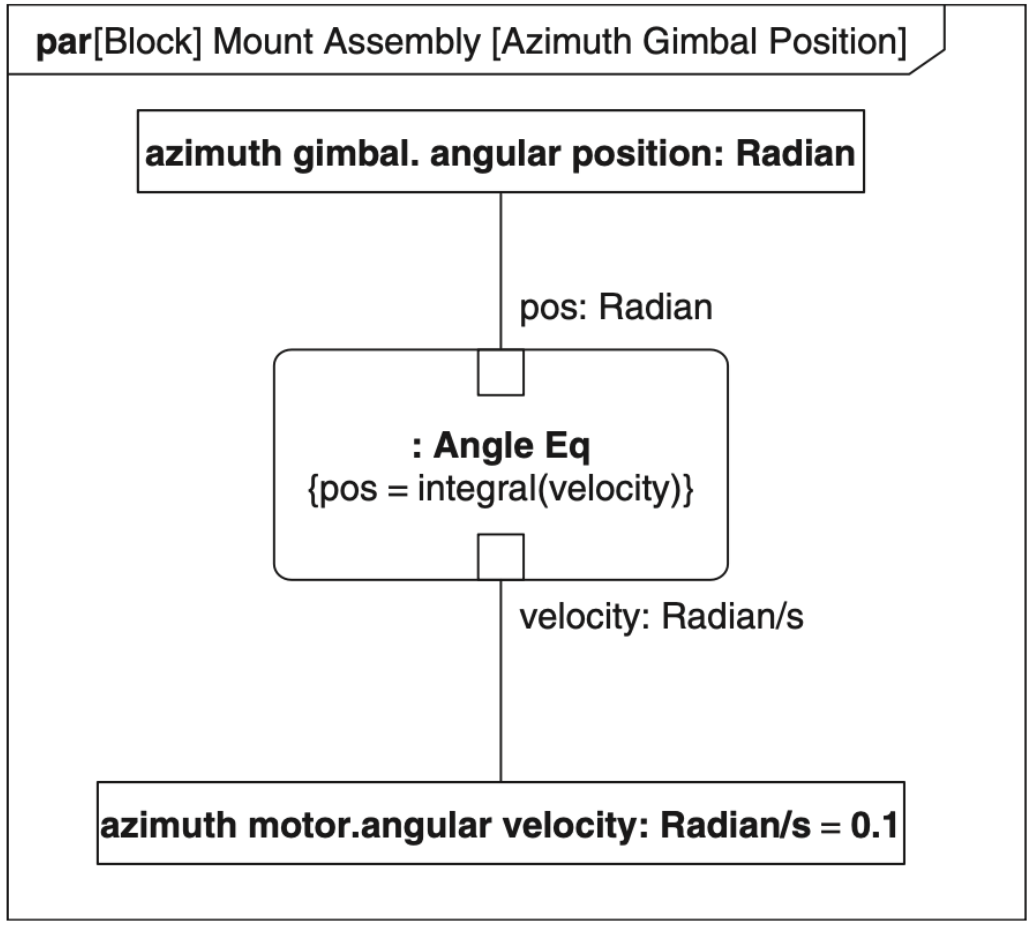


FIGURE 8.9

Using a time-dependent constraint.

Tempo explicito entrando nos blocos

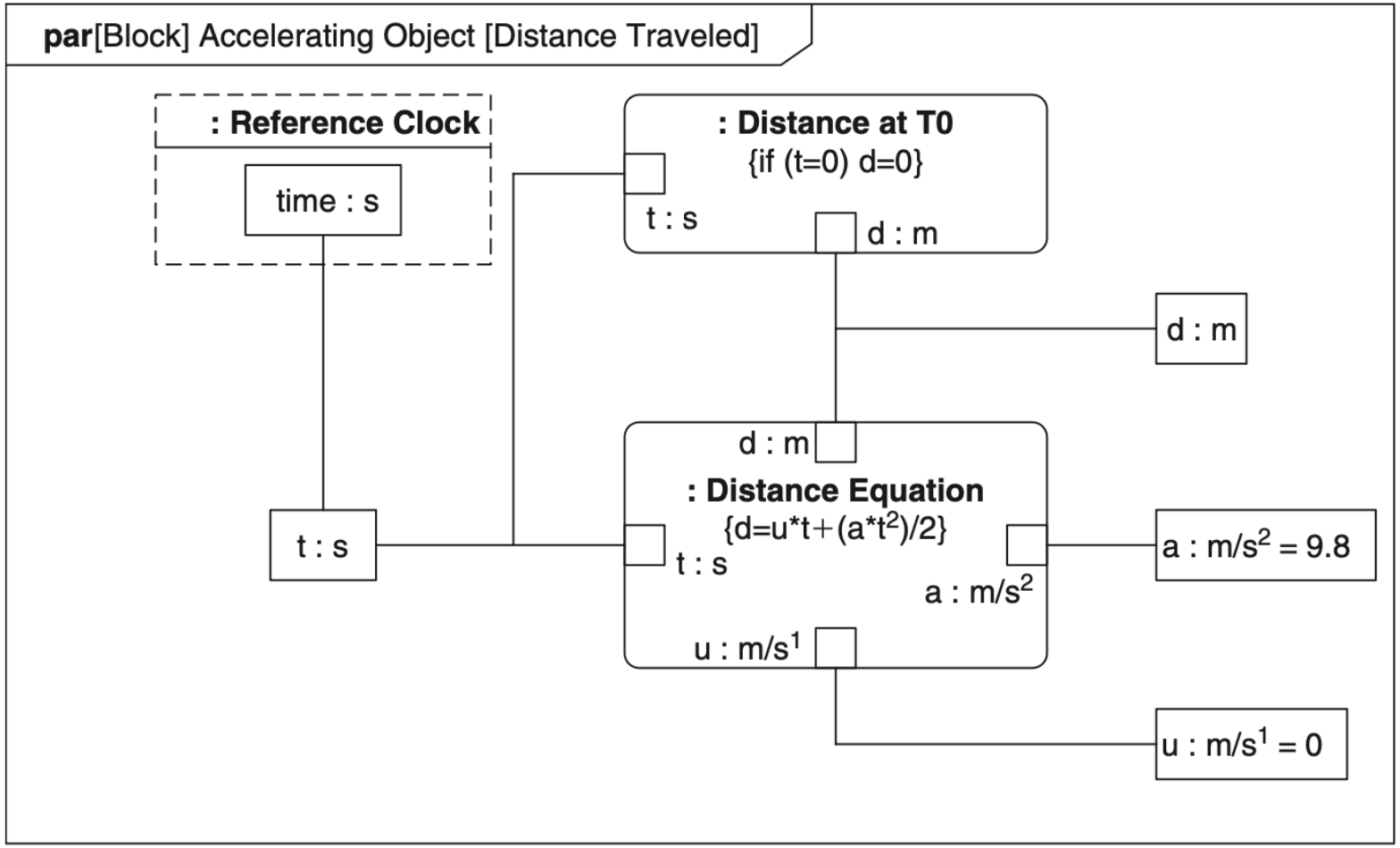


FIGURE 8.10

Explicitly representing time in a parametric diagram.

Usando um restrição para propriedades associadas à um fluxo.

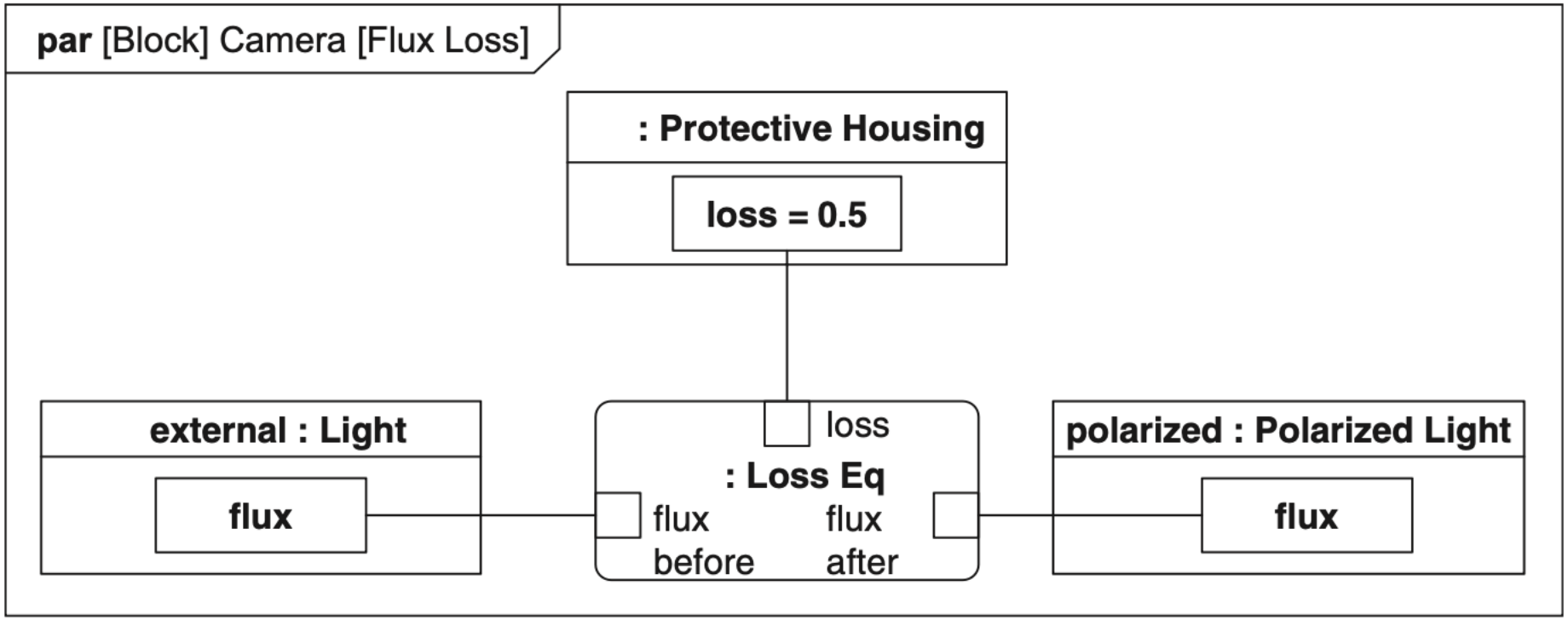


FIGURE 8.11

Constraining item flows.

Parâmetros e dados no diagrama de atividades

Parâmetros das atividades

- Os parâmetros de uma atividade podem ser **agrupados em conjuntos de parâmetros**, que devem ter apenas parâmetros de entrada ou saída como membros.

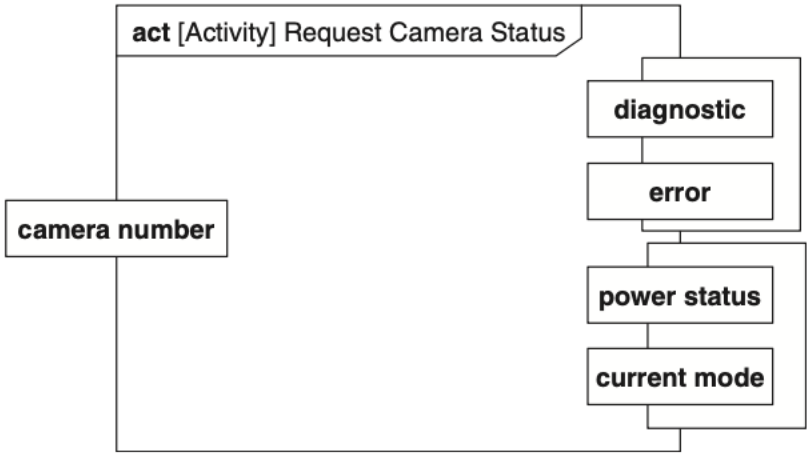


FIGURE 9.9

An activity with parameter sets.

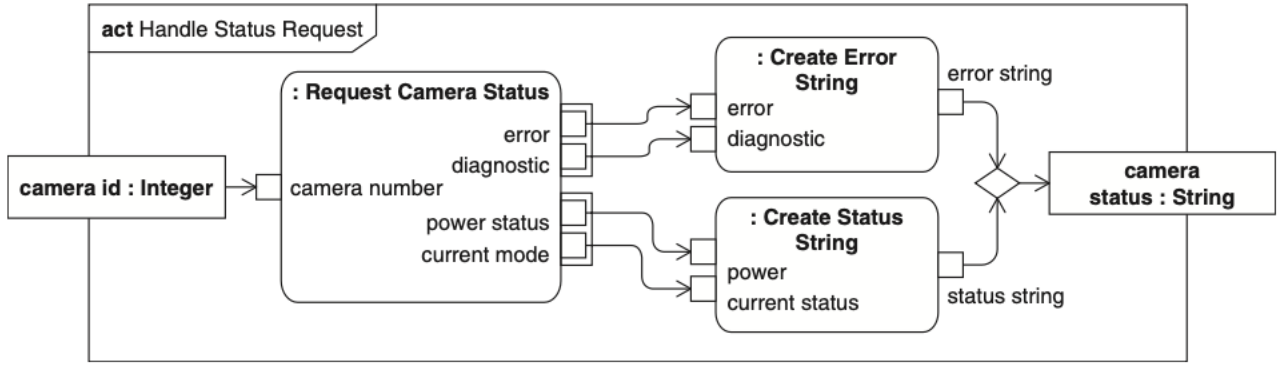


FIGURE 9.10

Invoking an activity with parameters sets.

Buffers e Data Stores

- Um **central buffer node** fornece um armazenamento para tokens de objeto fora de pinos e nós de parâmetro. Os tokens fluem para um nó de buffer central e são armazenados lá até que fluam novamente.
- Às vezes, as atividades exigem que os mesmos tokens de objeto sejam armazenados para acesso por várias ações durante a execução. Um tipo de nó de objeto chamado **data store node** pode ser usado para isso.
  - Ao contrário de um nó de buffer central, um nó de armazenamento de dados fornece uma cópia de um token armazenado em vez do original.

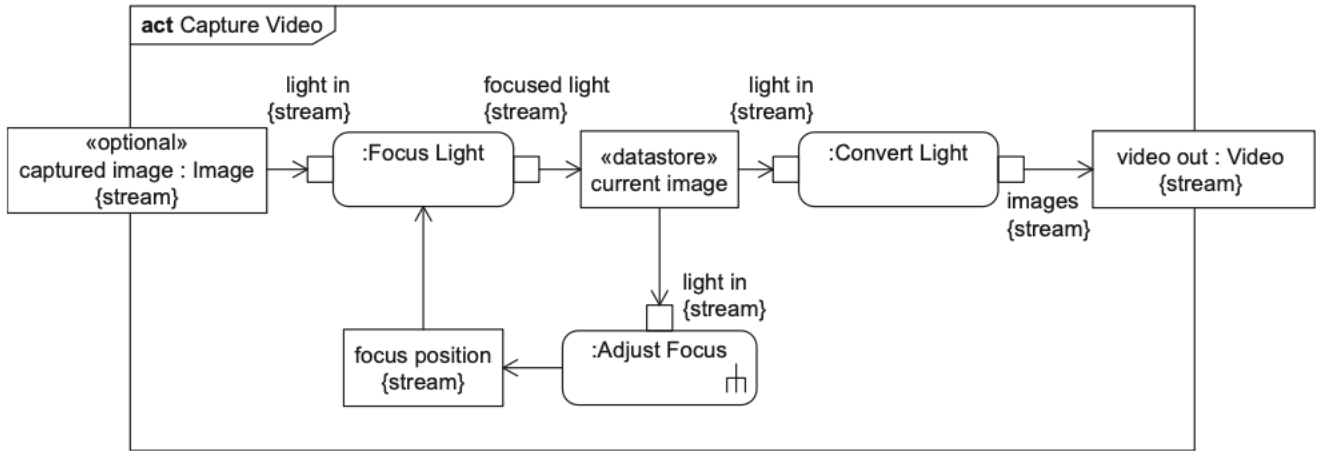


FIGURE 9.11

Using a data store node to capture incoming light.

Analysis Context

A modeler can create an analysis context

- What often occurs, however, is that the constraints on block properties vary depending on the analysis requirements.
- An analysis context is modeled as a block with associations to the block being analyzed (i.e., subject of the analysis), the chosen analysis model, and any intermediate transformations.
- By convention, the block being analyzed is referenced by the analysis context block because there may be many different analysis contexts for the block being analyzed.
- A white diamond symbol or a simple association with no end adornment is used to represent a reference from the analysis context block to the subject of the analysis.

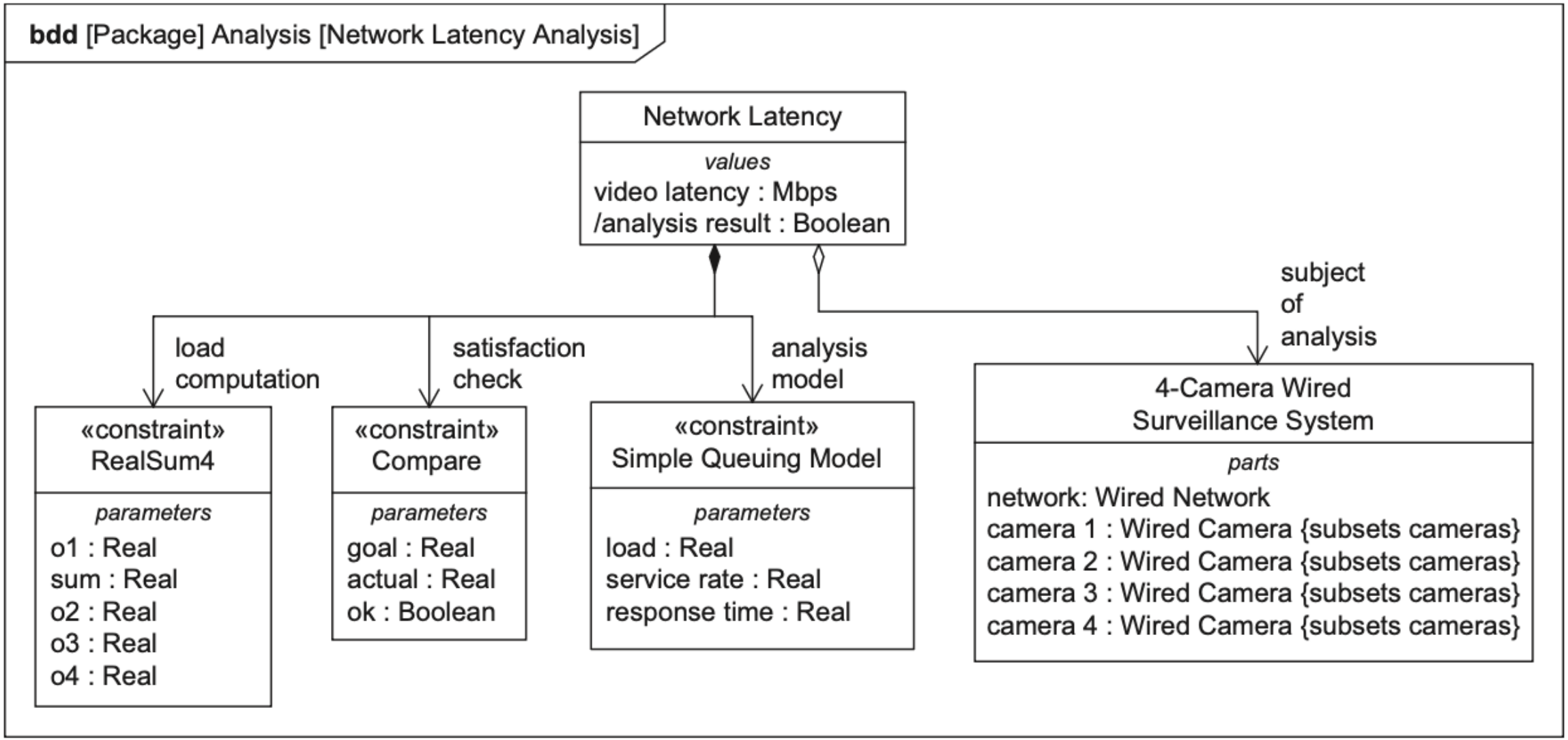


FIGURE 8.12

An analysis context shown on a bdd (constraint equations not shown).

Binding values

- The parameters of the analysis model are bound to the properties of the subject of analysis.
- The loads on the system from all four cameras in the subject of analysis are summed to establish the total load using load computation. The network bandwidth of the subject of analysis is used to establish the service rate for the analysis model.
- The response time, calculated using analysis model, is then compared to the required video latency using satisfaction check.

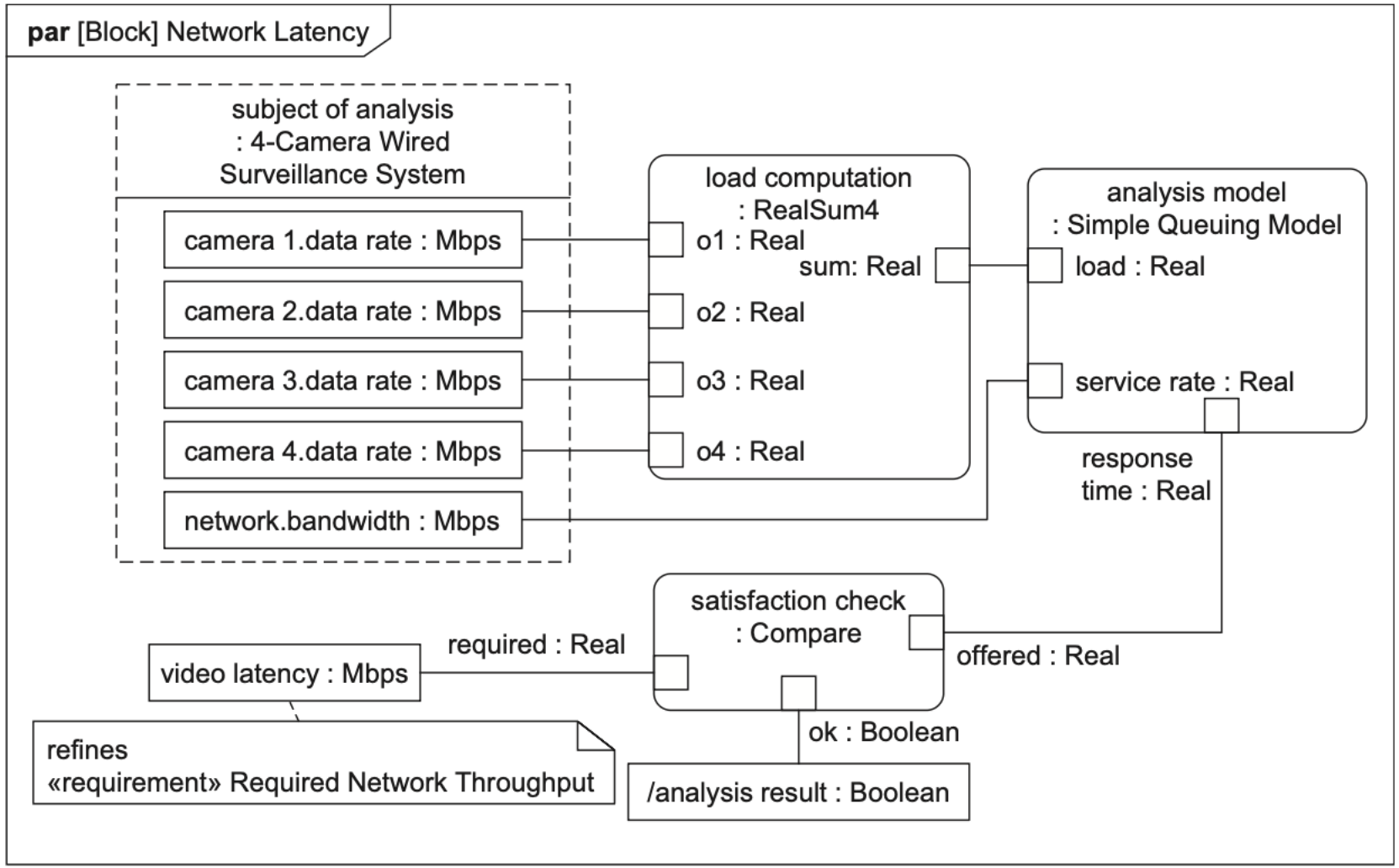


FIGURE 8.13

Binding values in an analysis context.

Trade

- A common use of constraint blocks is to support trade studies. A trade study is used to compare a number of alternative solutions to see whether and how well they satisfy a particular set of criteria.
- Each solution is characterized by a set of measures of effectiveness (often abbreviated “moes”) that correspond to the evaluation criteria and have a calculated value or value distribution.
- The moes for a given solution are then evaluated using an objective function (often called a cost function or utility function), and the results for each alternative are compared to select a preferred solution.

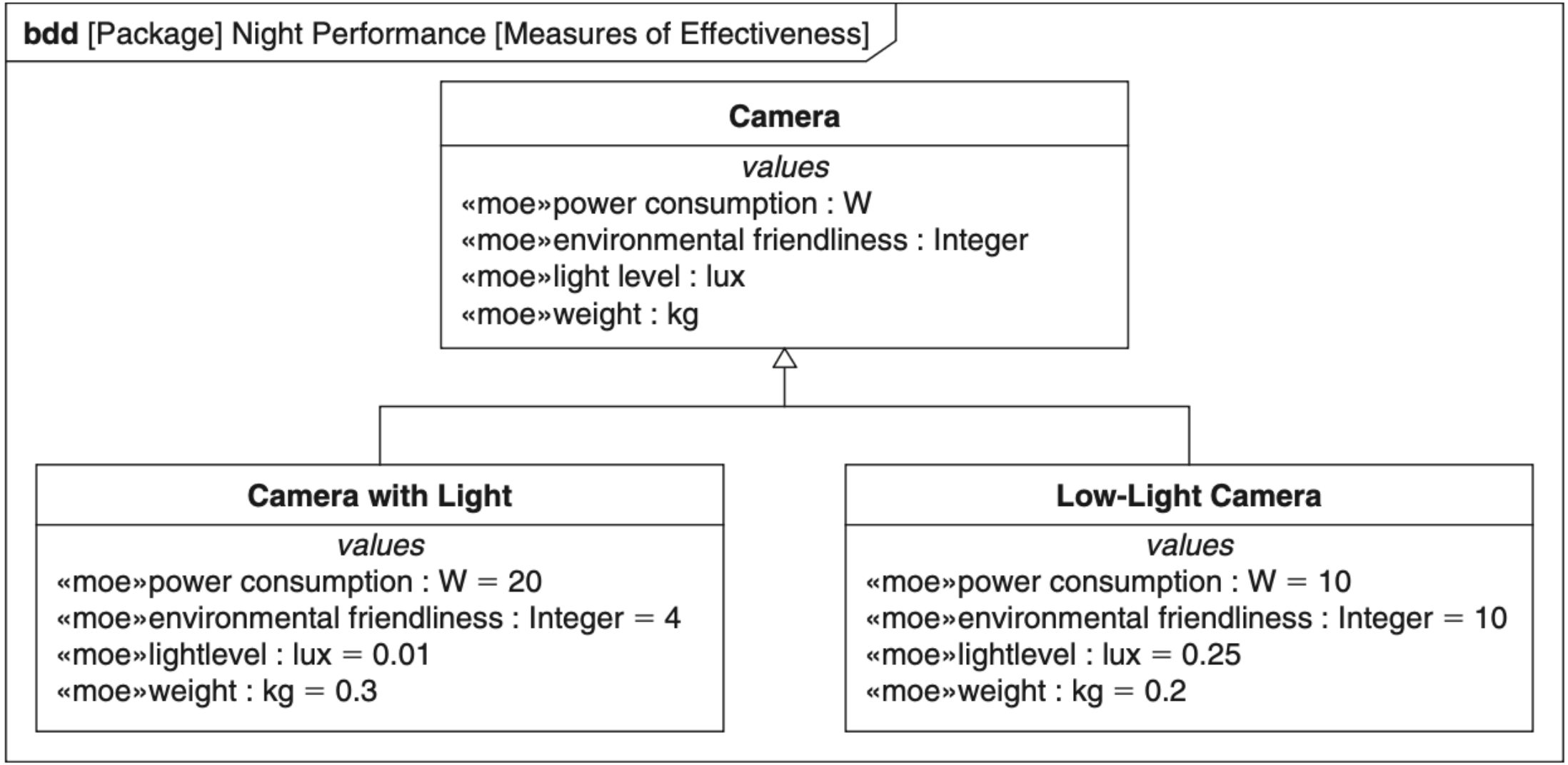


FIGURE 8.14

Two variants of a camera for handling low-light conditions.

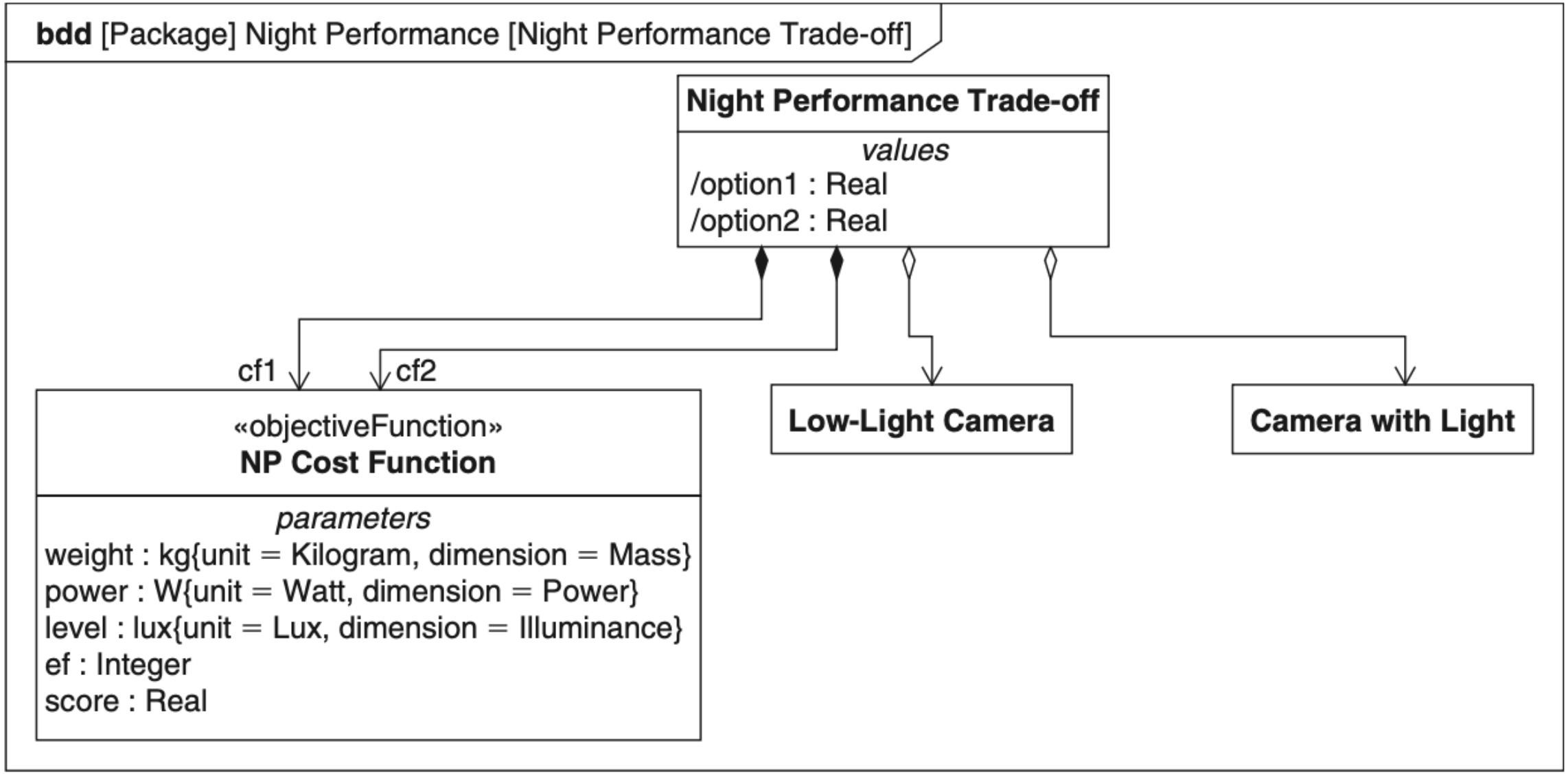


FIGURE 8.15

A trade study represented as an analysis context.

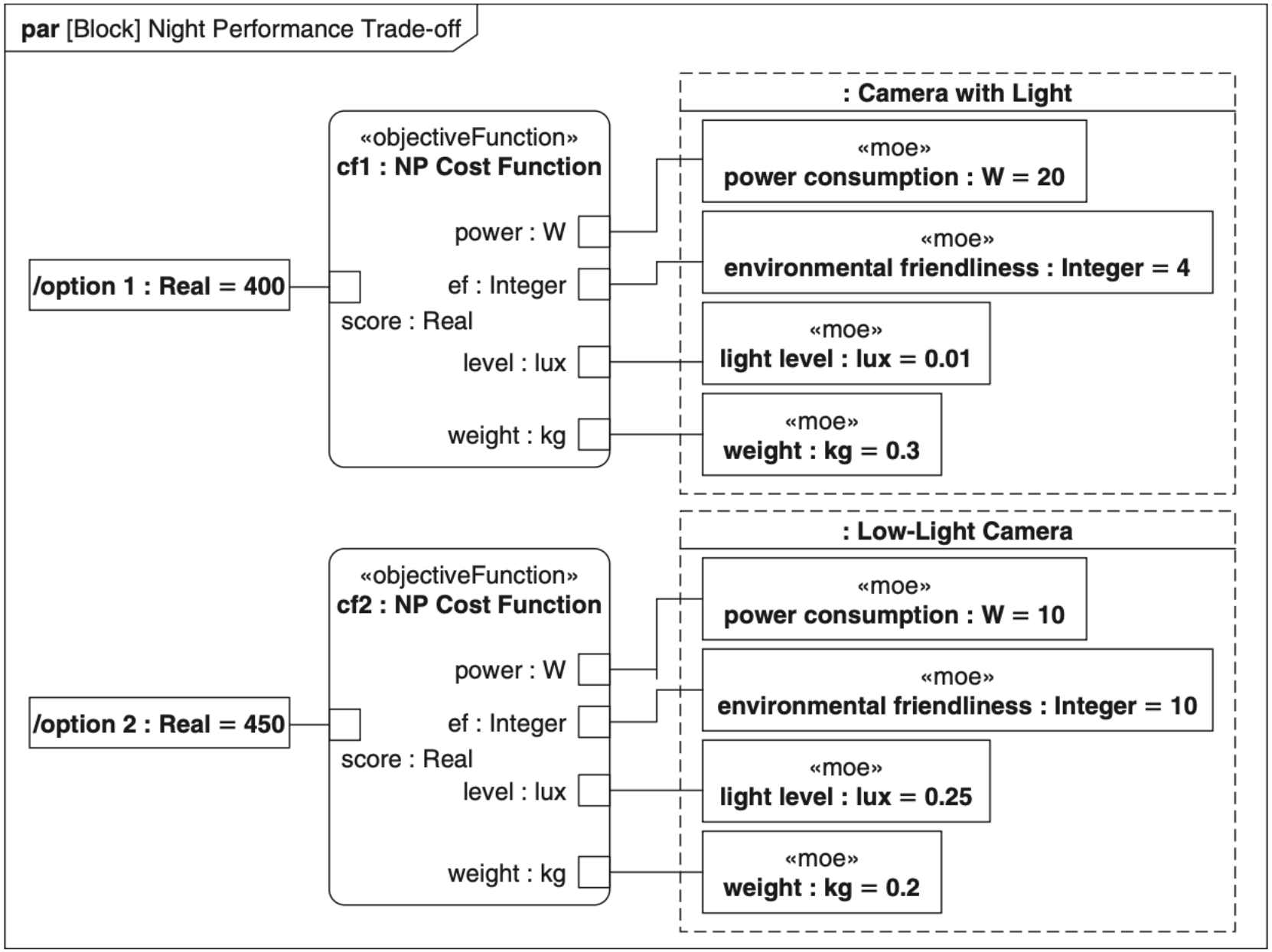


FIGURE 8.16

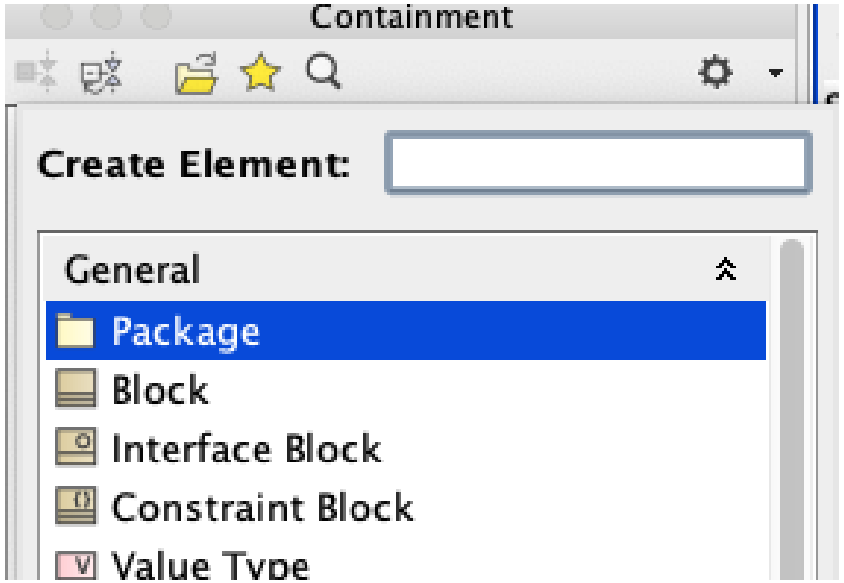
Trade-off results between the two low-light camera variants.

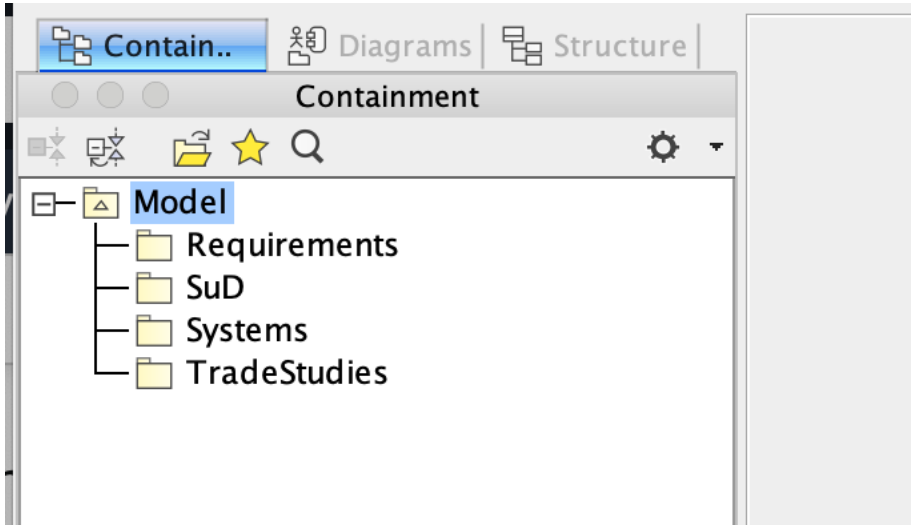
What is a trade study!?

Step by step…

0) Create the packages of what you want to do a trade study

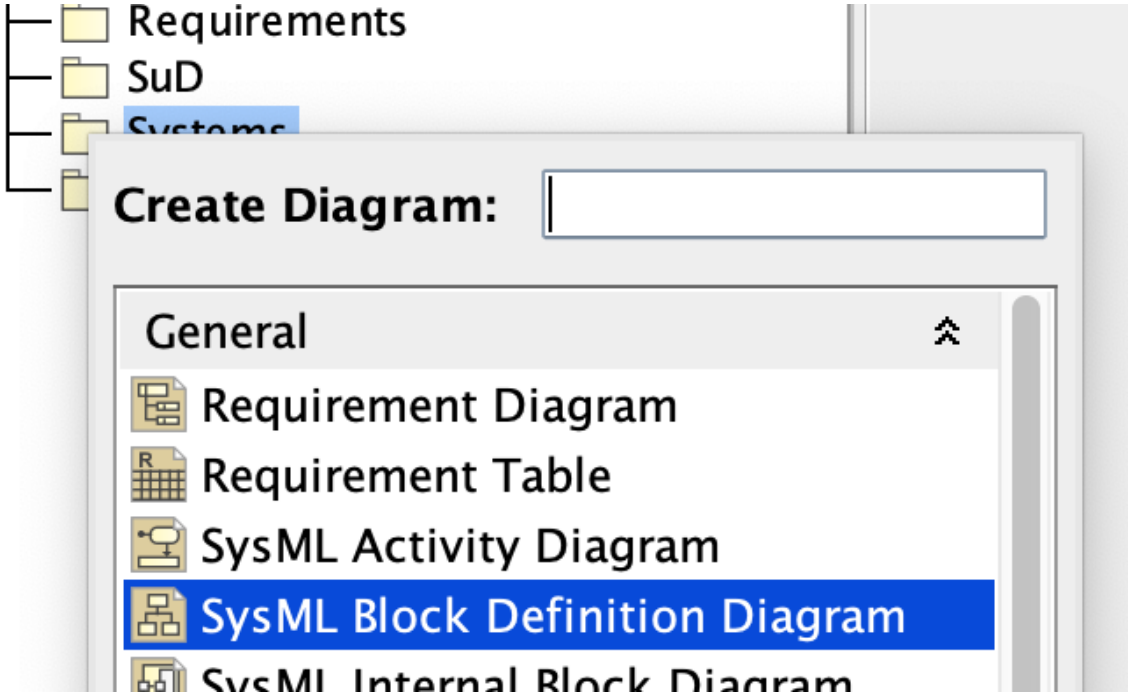
- First Create the basic packages
  - Click on model, and create two packages:
    - **Requirements** and
    - **SuD** and
    - **Systems** and
    - **TradeStudies**

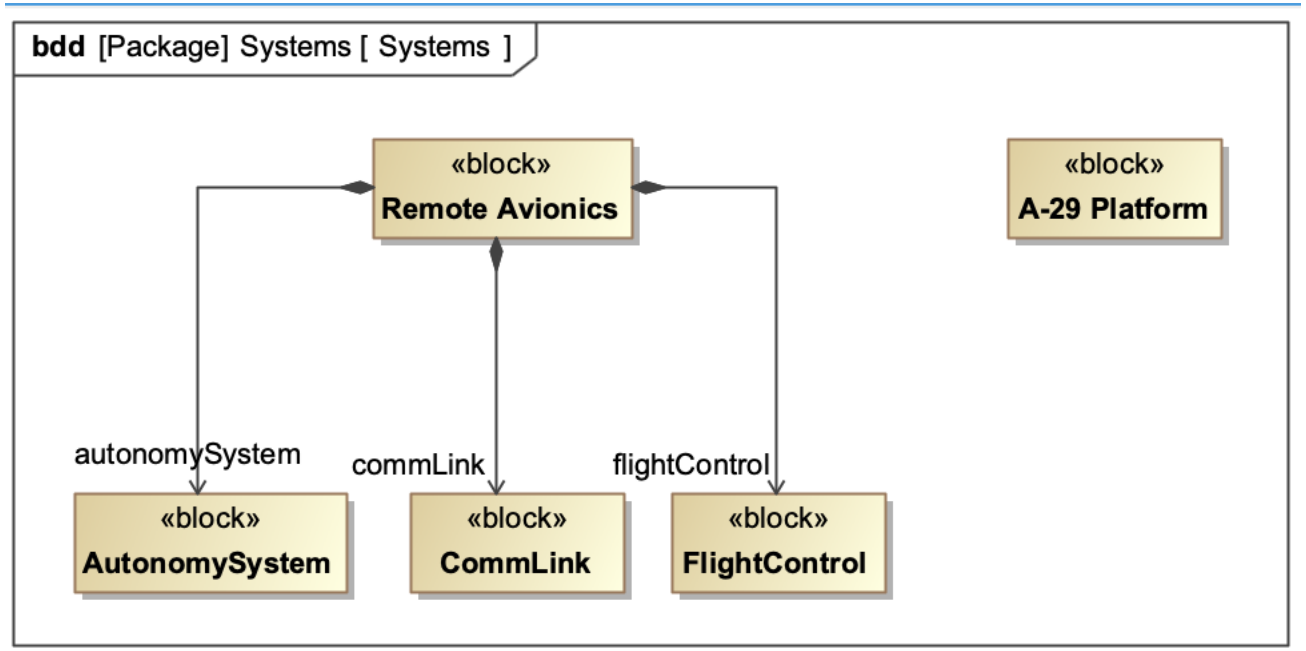




1) Create the architecture that you want to do a trade study

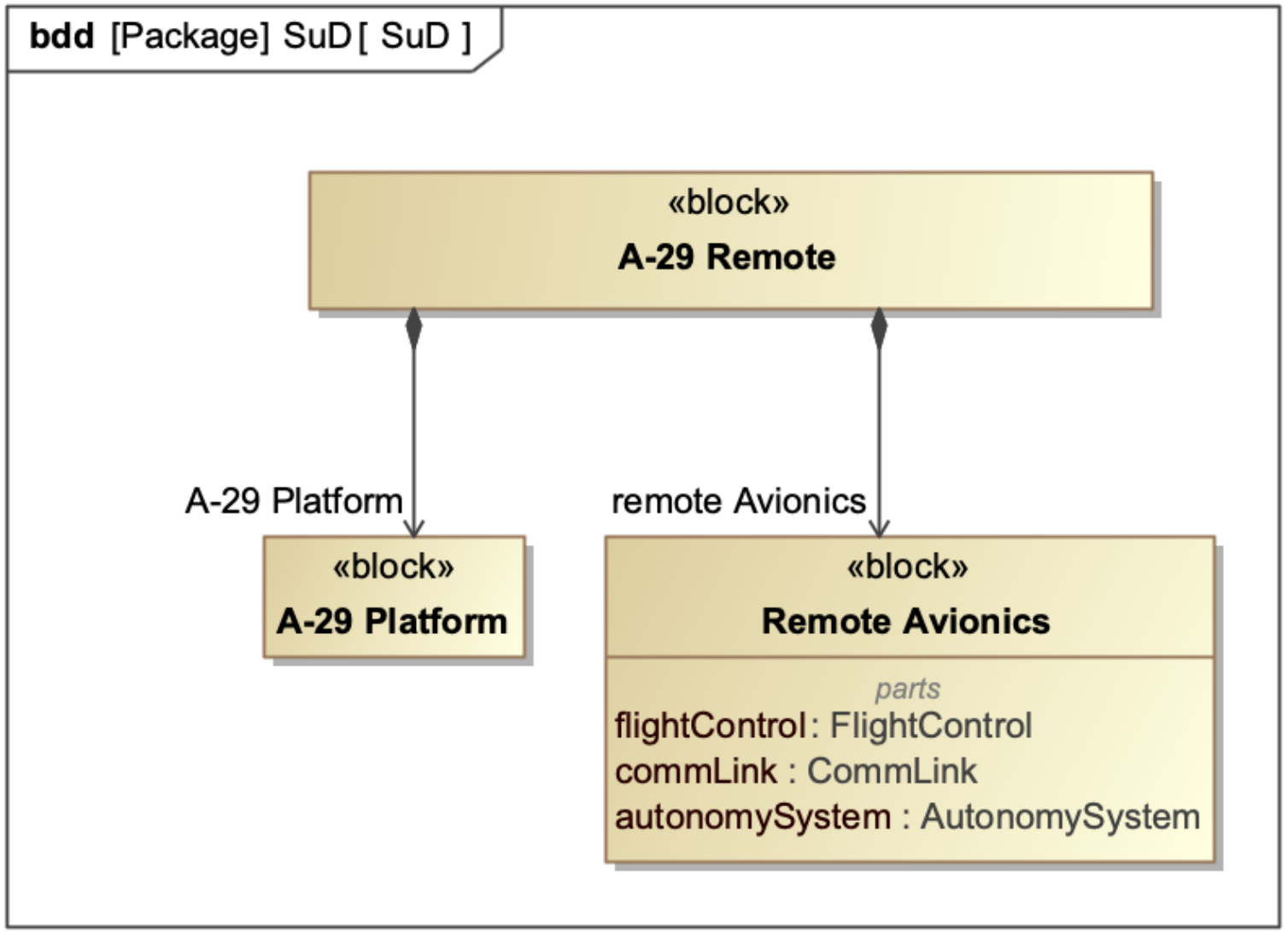
- Create the “systems” and a BDD



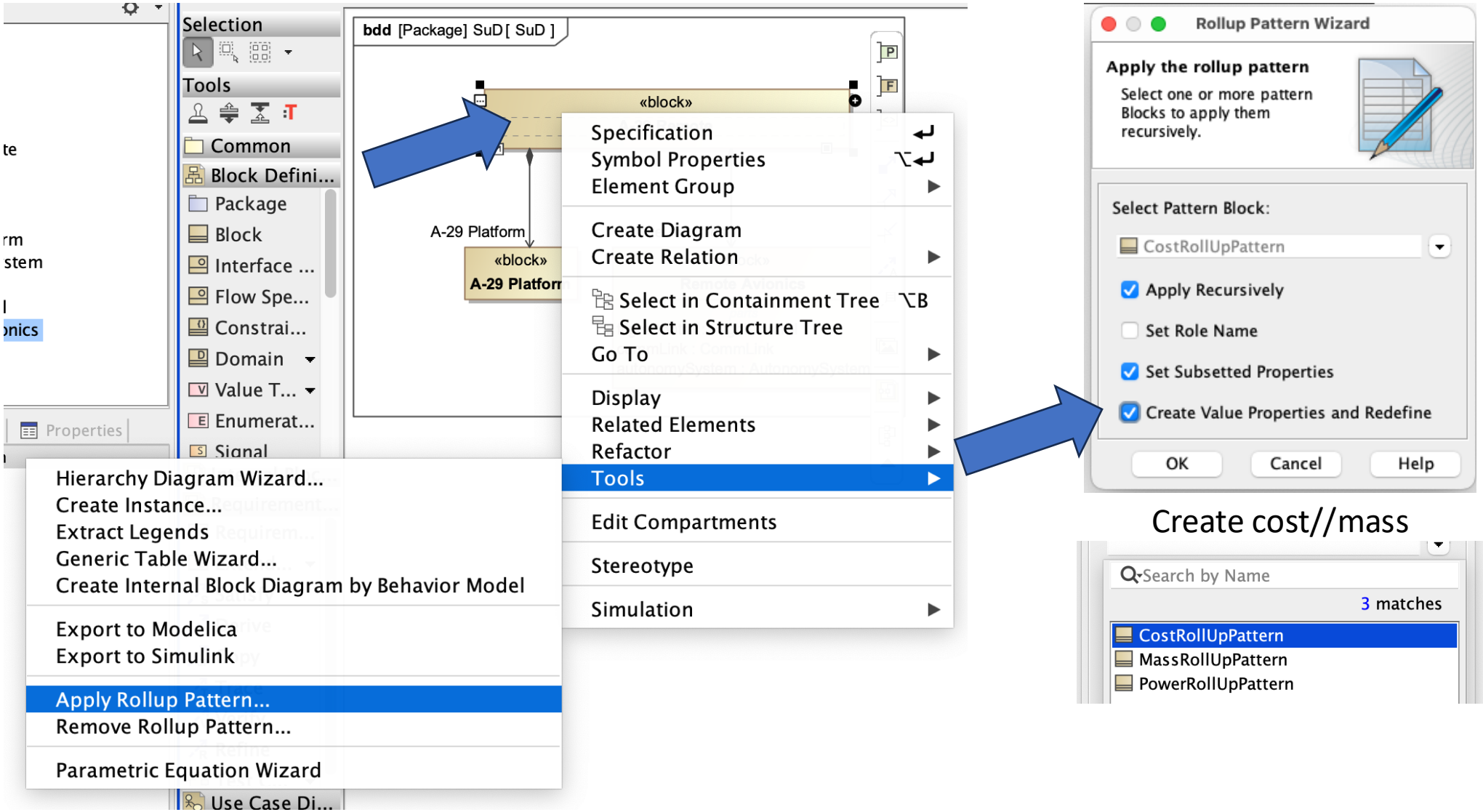


1) Create the architecture that you want to do a trade study

- Do the same w/ the SuD and add the compositions of the SuD System (A-29 Remote) w/ the other existing elements

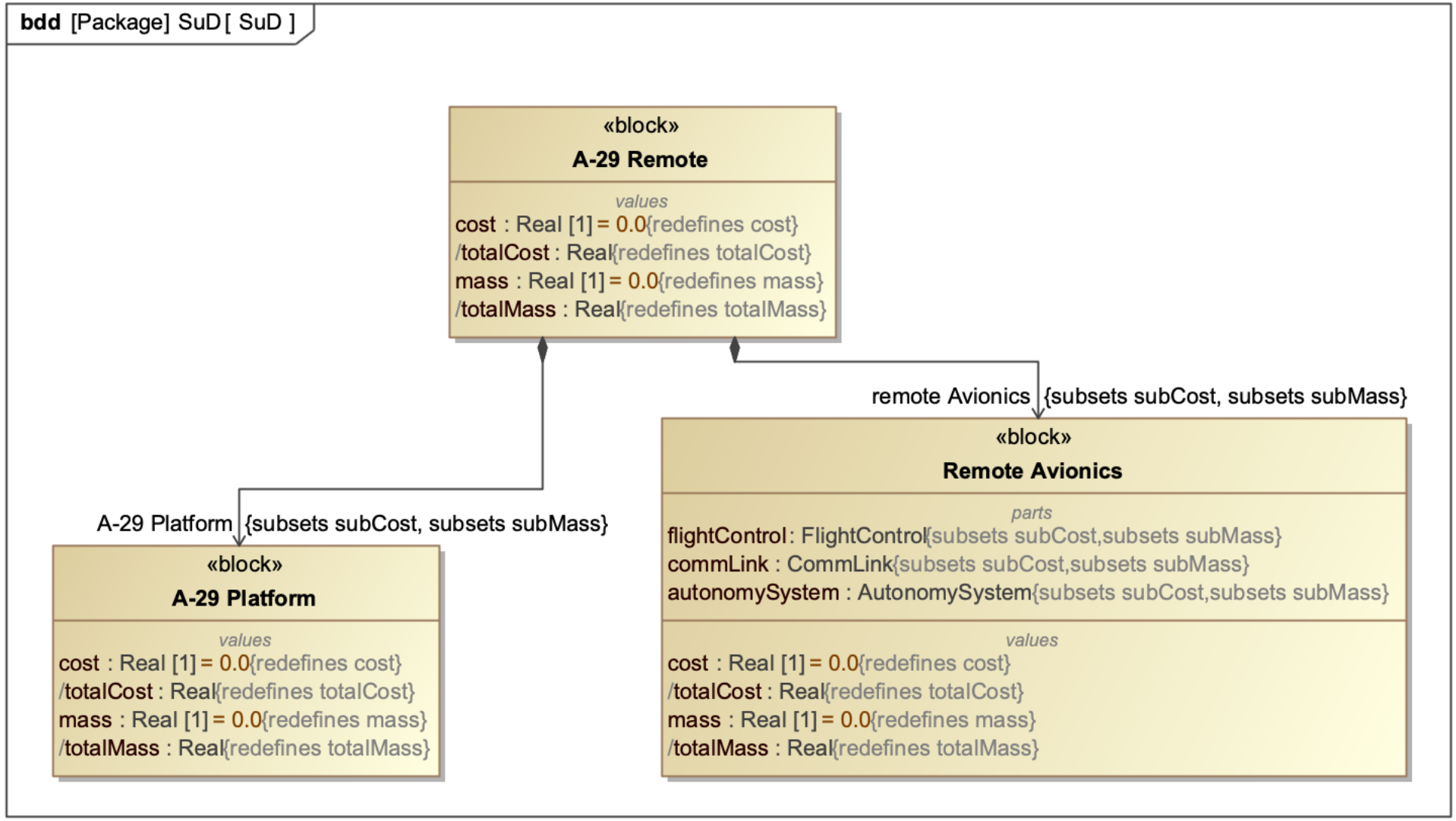


2) Create value alternatives (roll-up patterns)

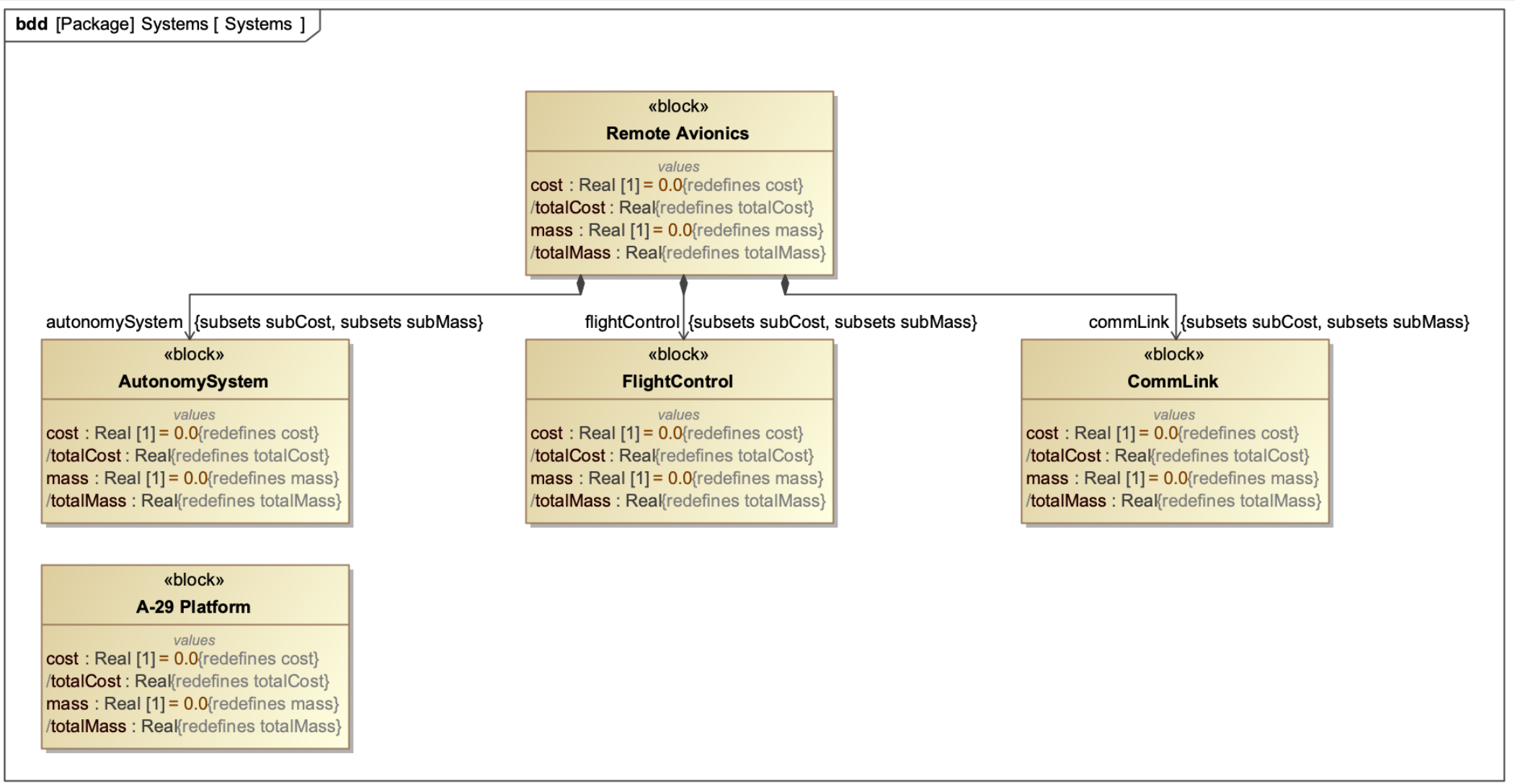


Create cost//mass

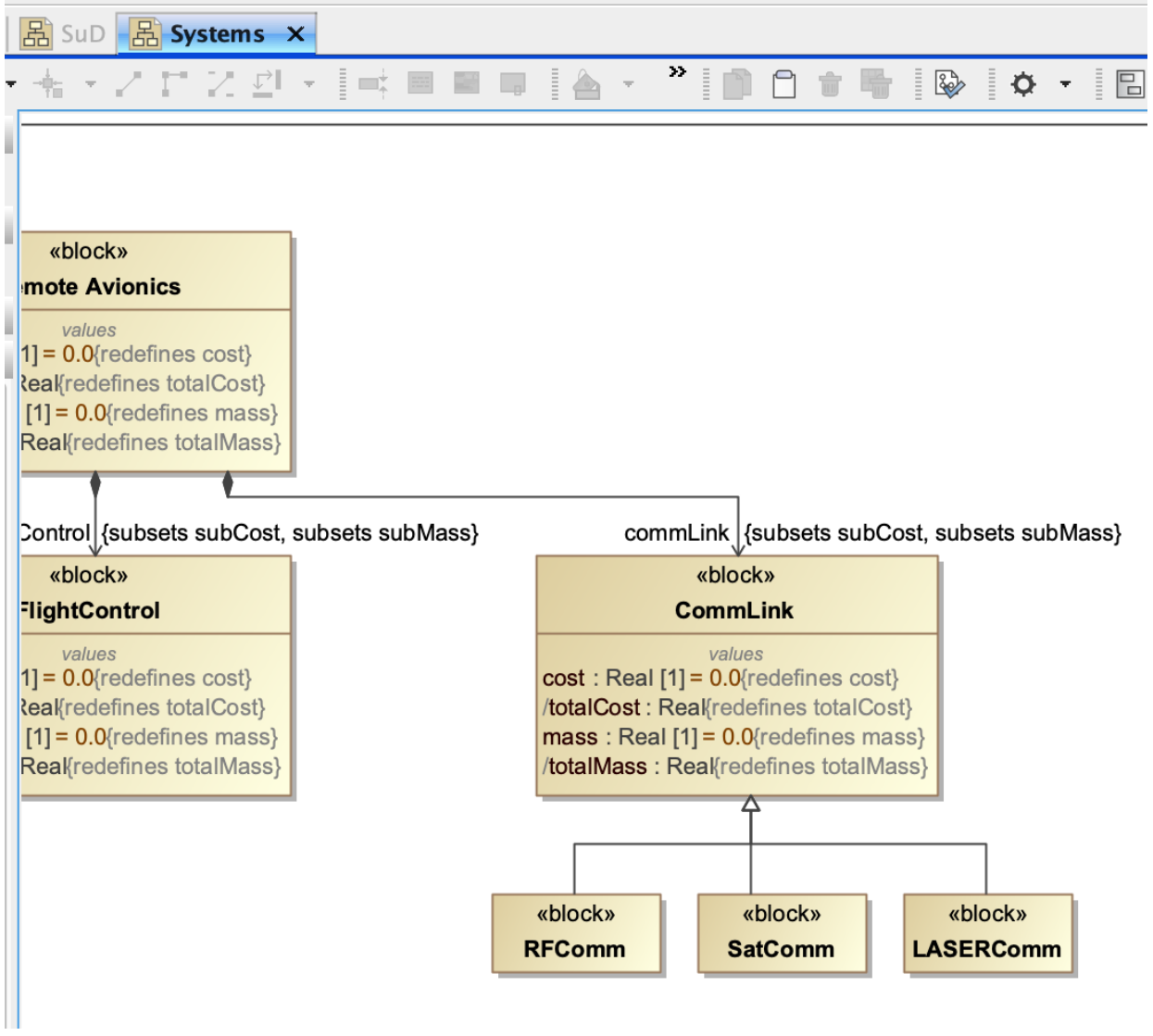
Result of the rollup pattern



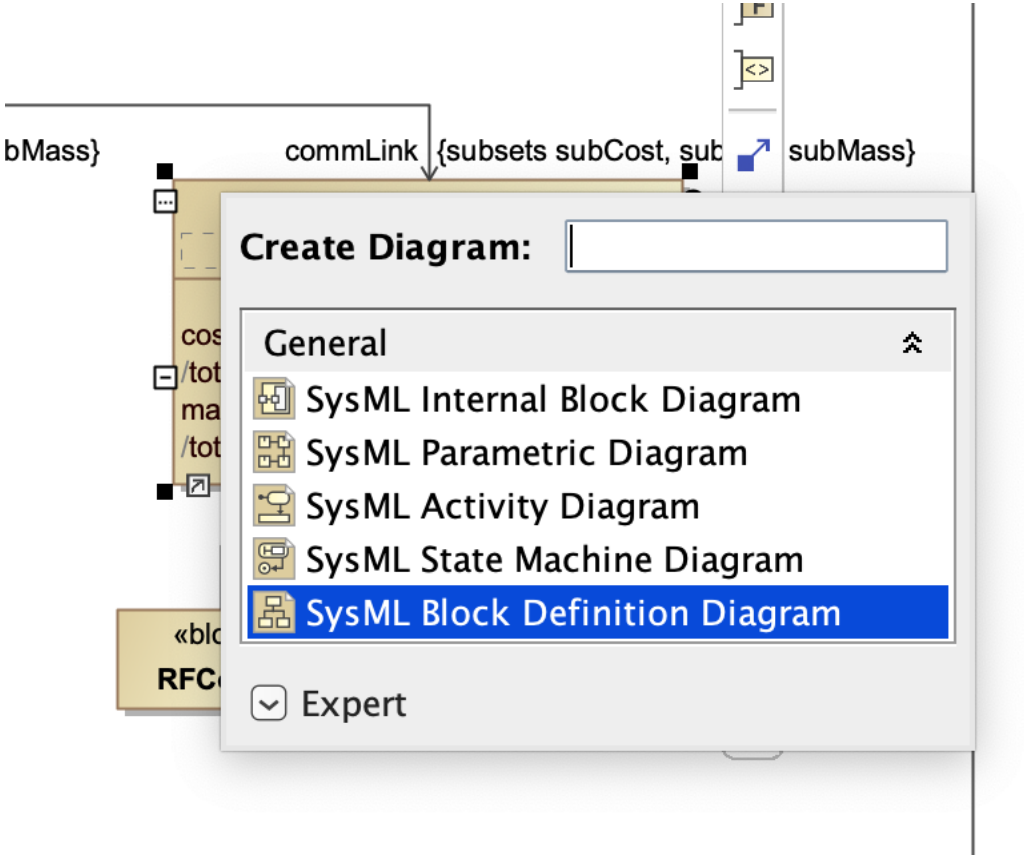
Result of the rollup pattern

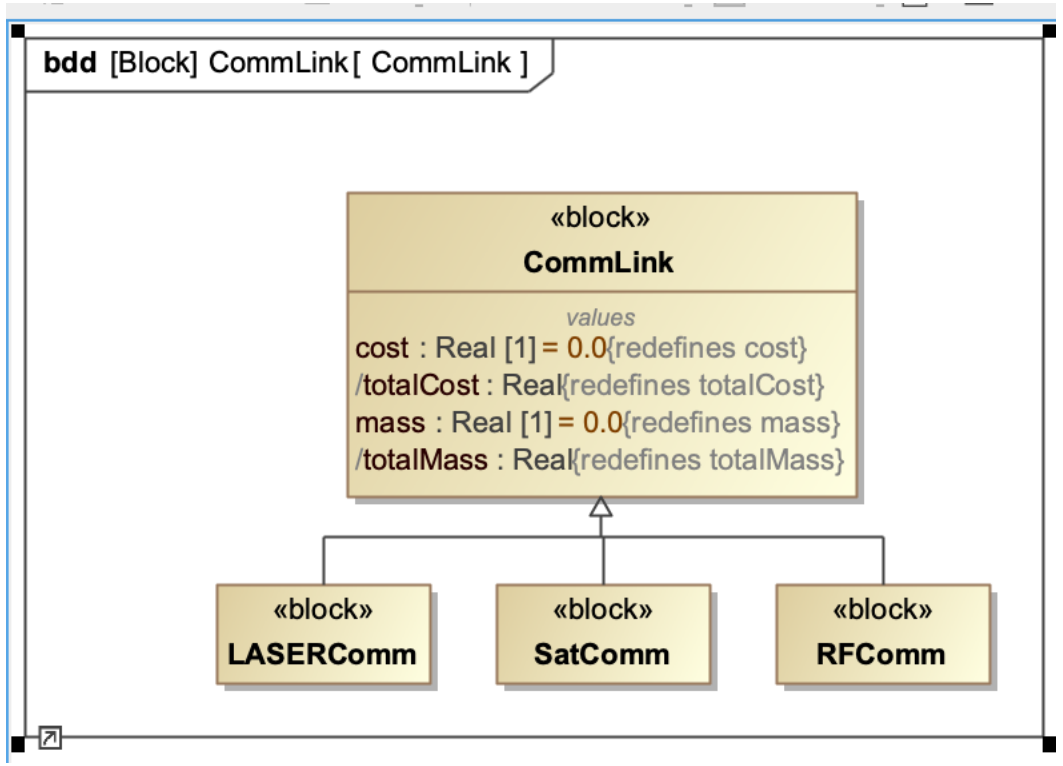


Return into the Systems BDD and add 3 CommLink alternatives

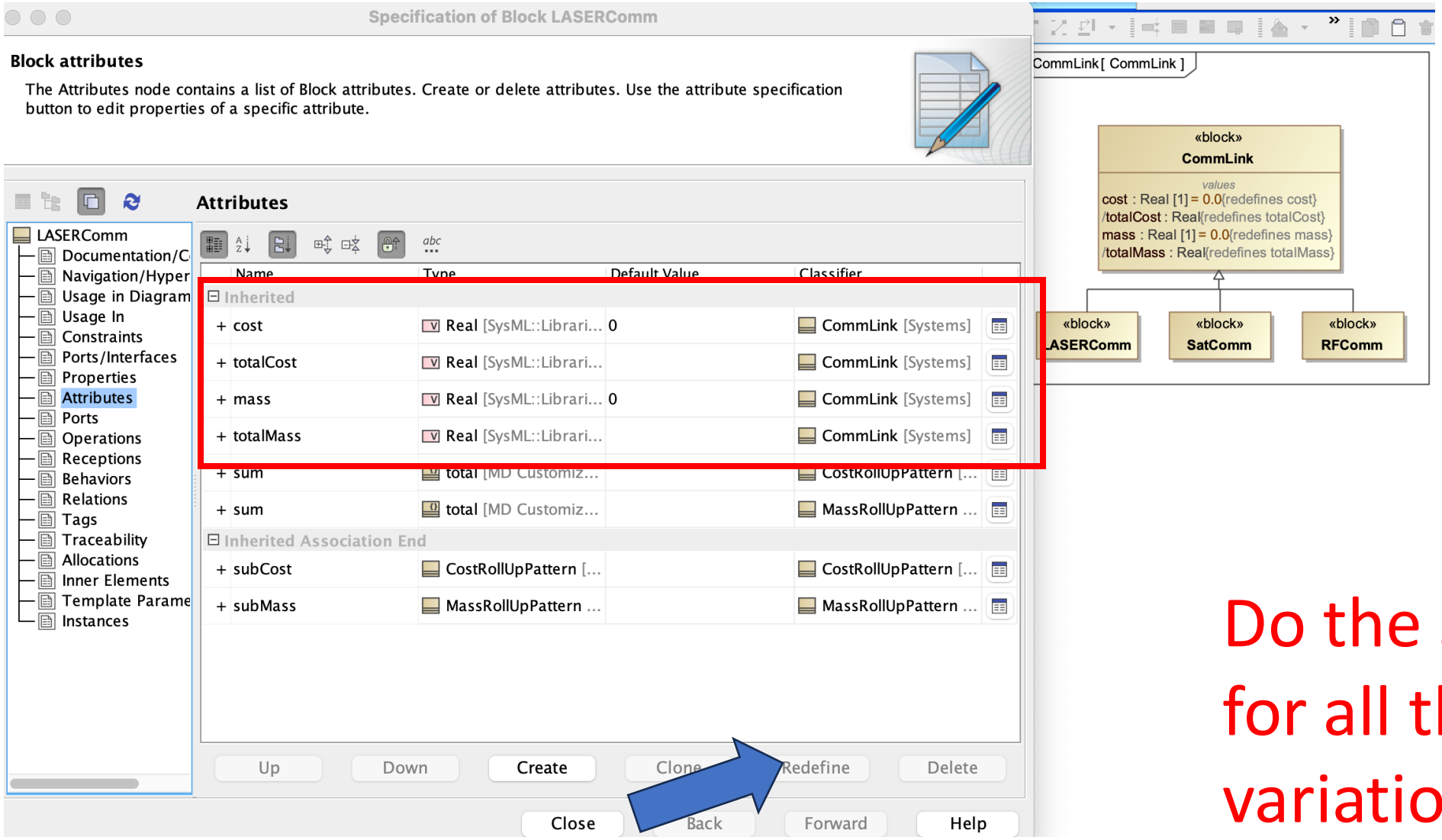


To easy things up create a bdd of the commlink



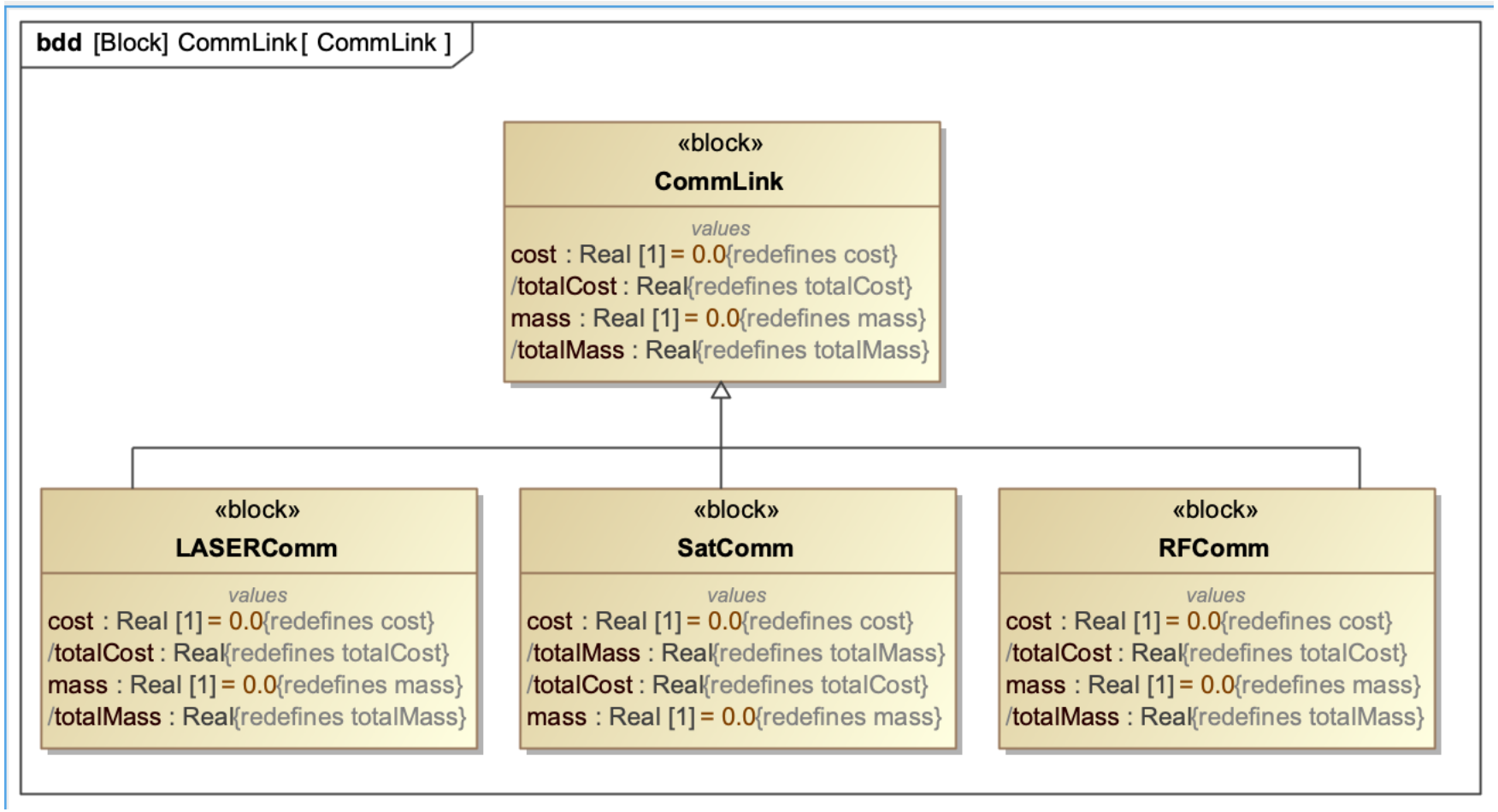


Redefine all the property values (ONLY THE PV)

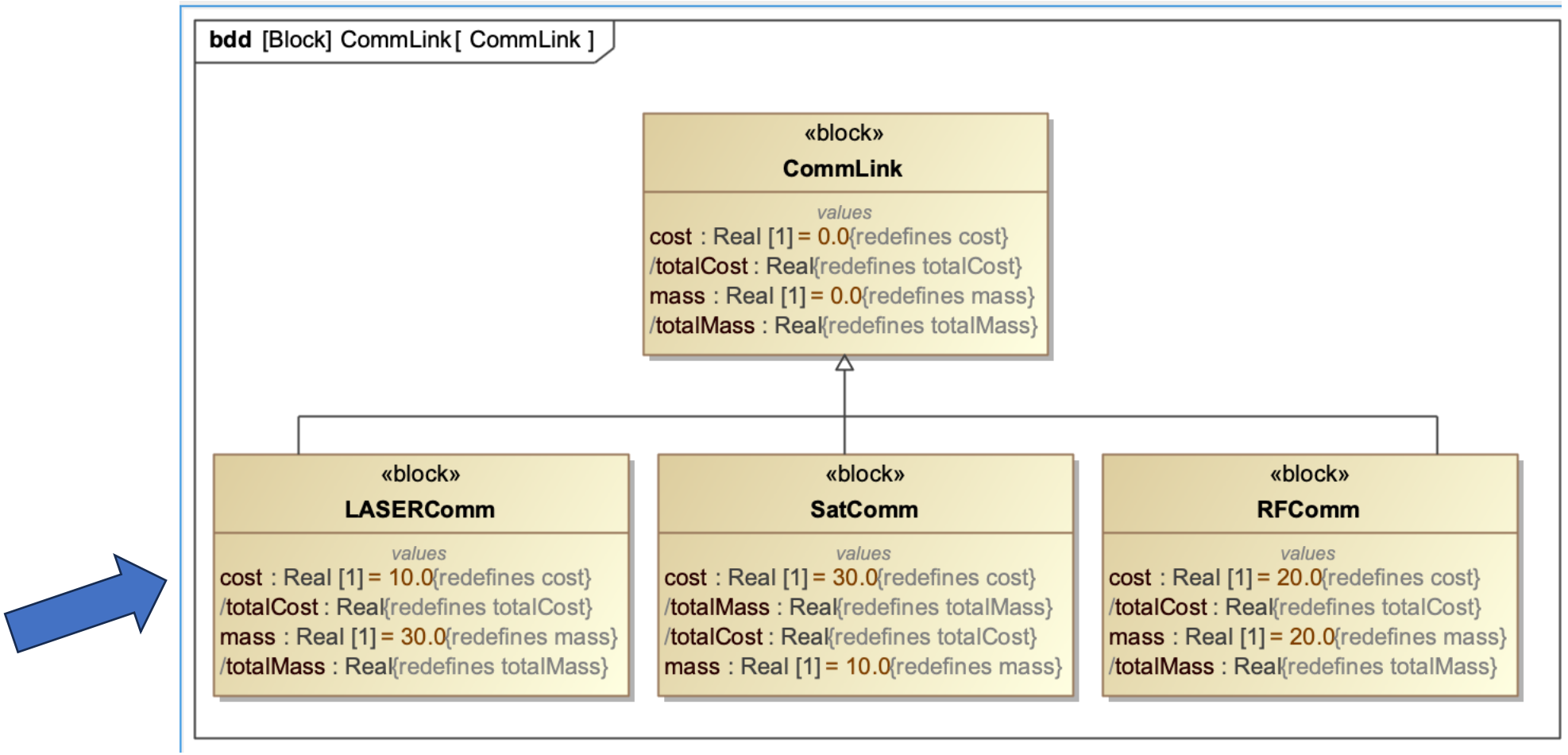


Do the same for all three variations

Will result in a thing like this

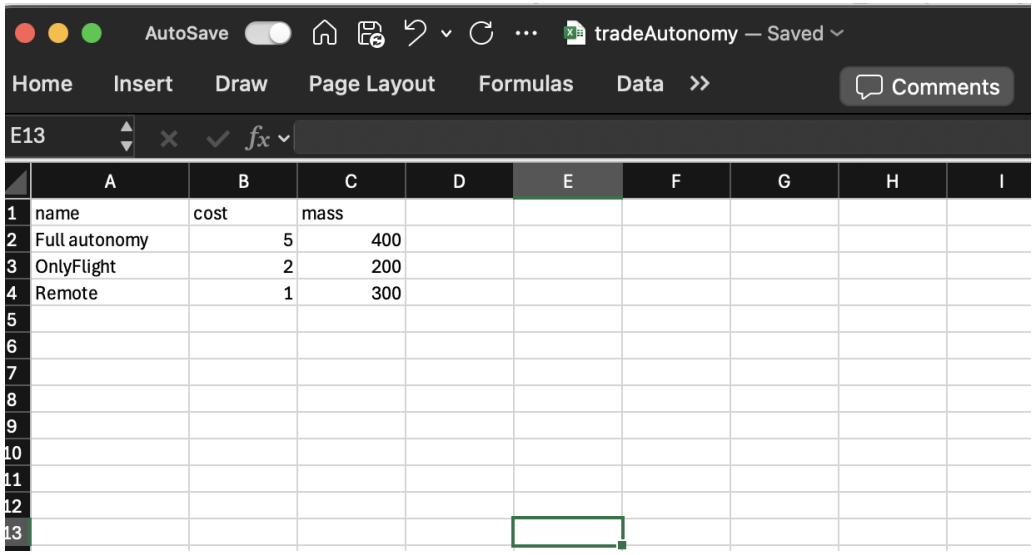


Place some numbers…

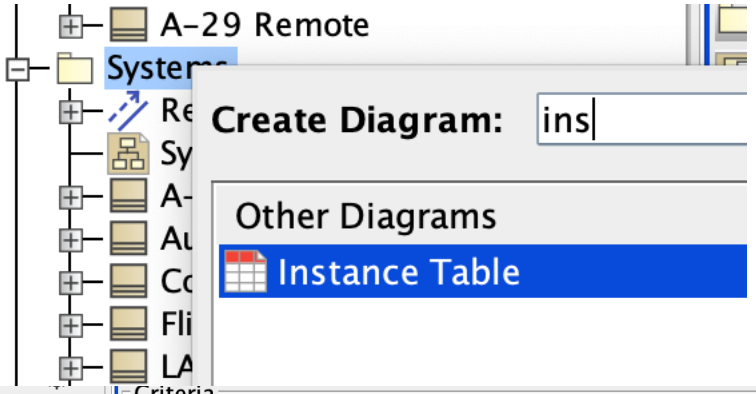


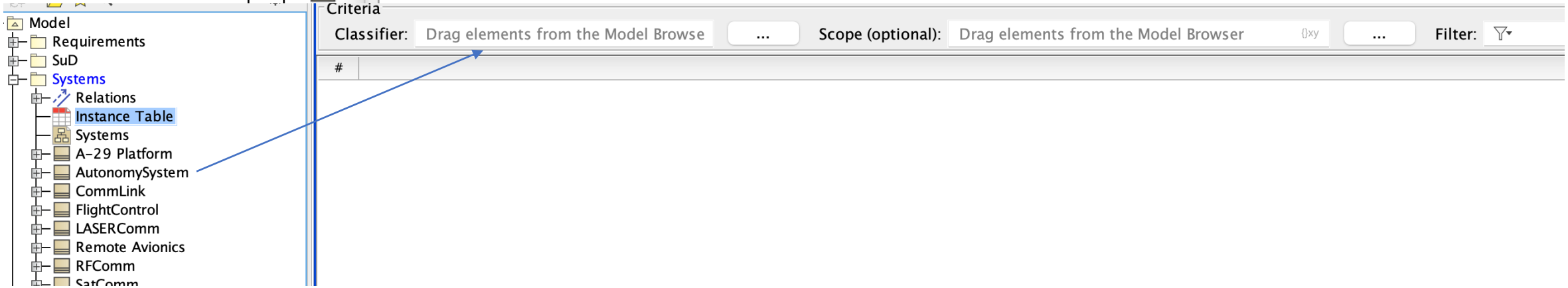
All this alternatives could be from a library of possible model configurations!!!

Create an Excel Alternative

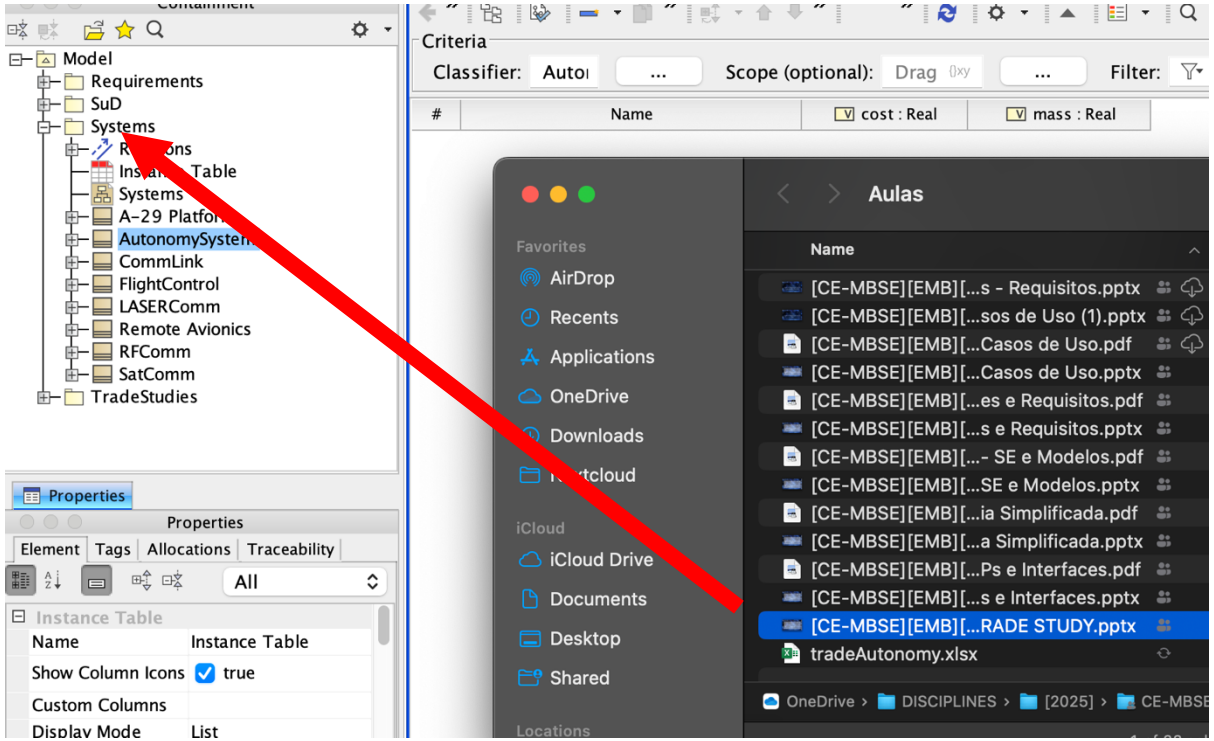


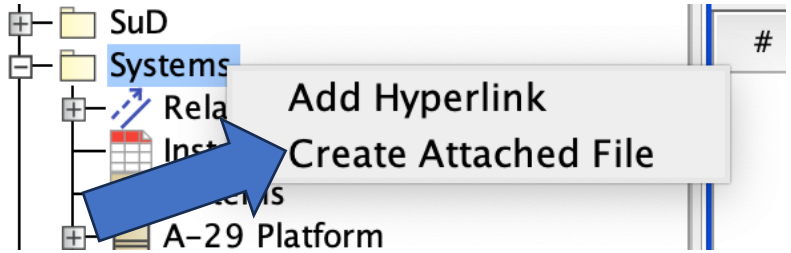
	A	B	C	D	E	F	G	H	I
1	name	cost	mass						
2	Full autonomy	5	400						
3	OnlyFlight	2	200						
4	Remote	1	300						
5									
6									
7									
8									
9									
10									
11									
12									
13									

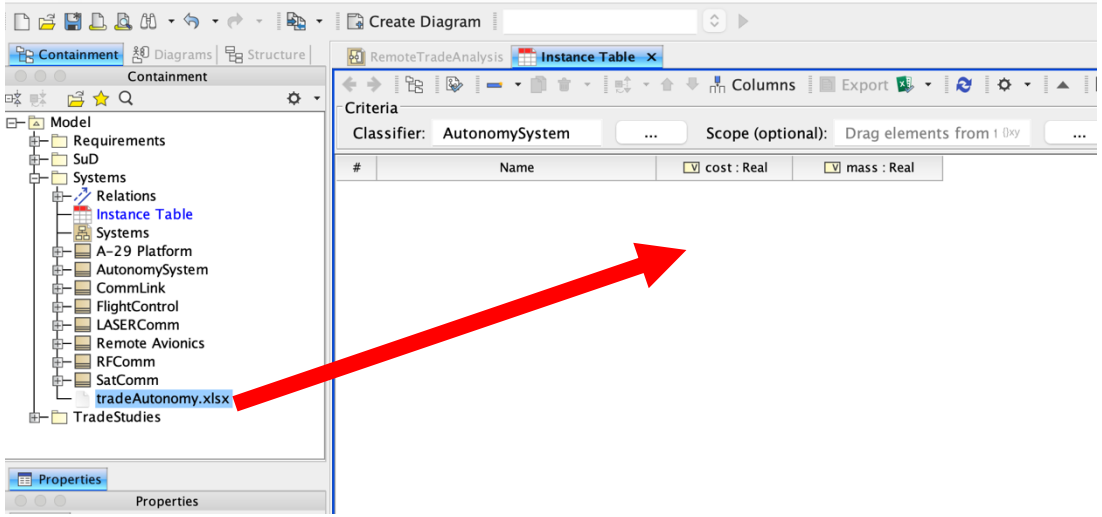




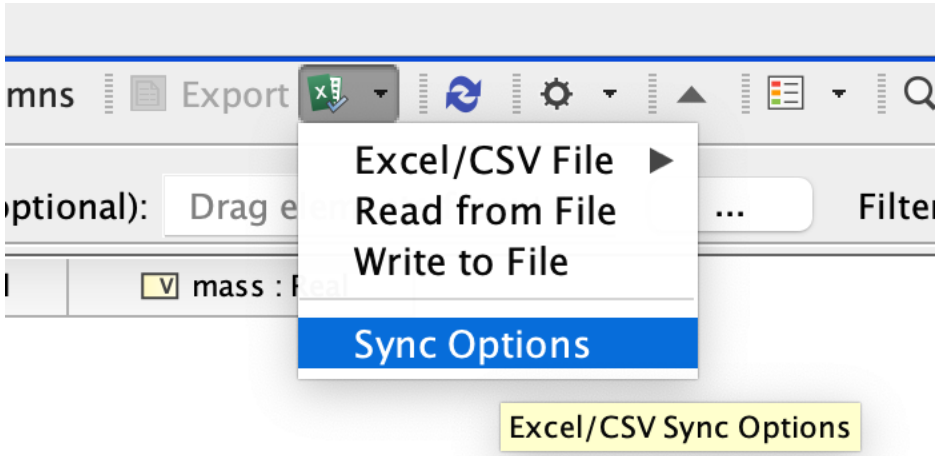
Connect the reference to the file

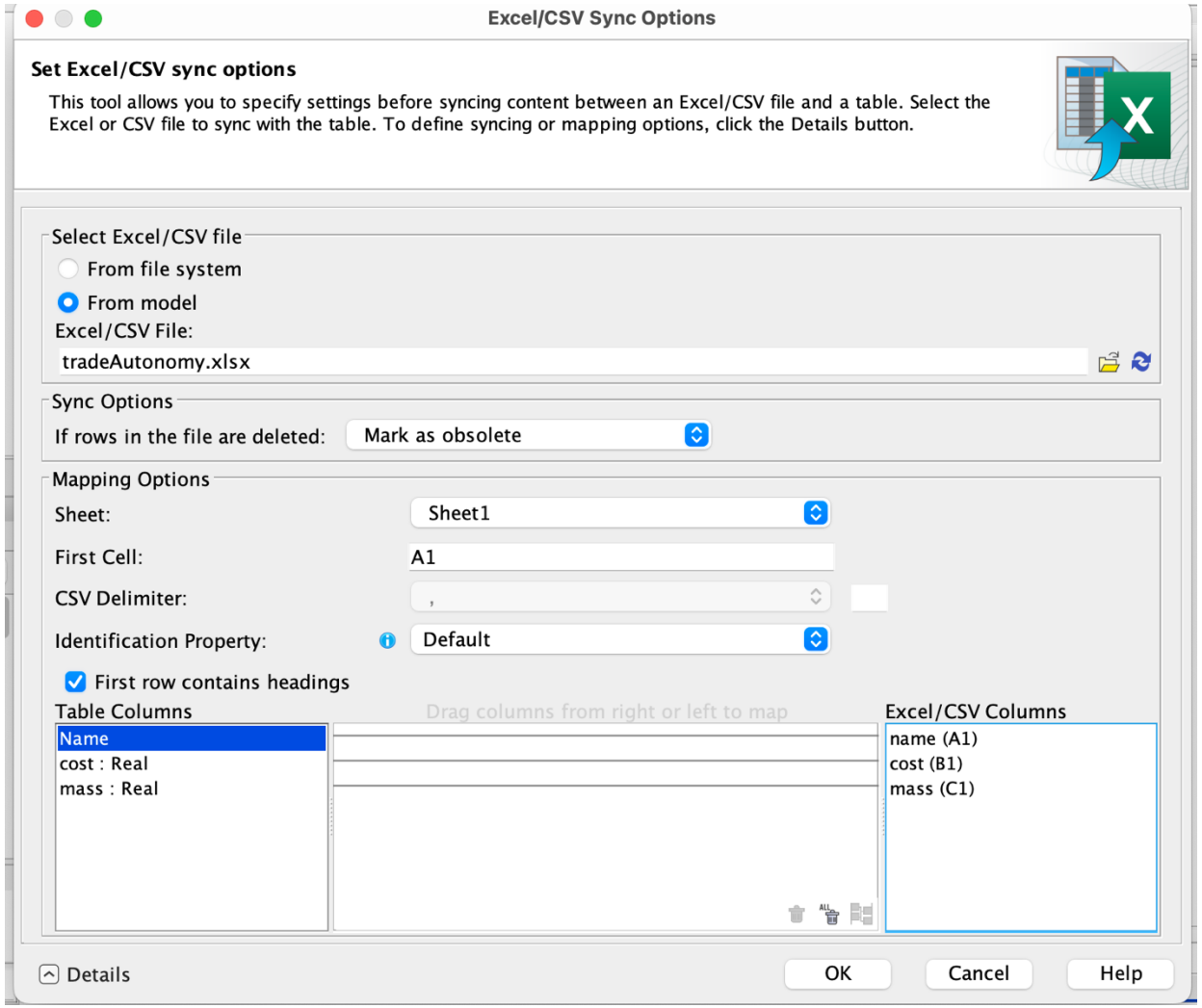






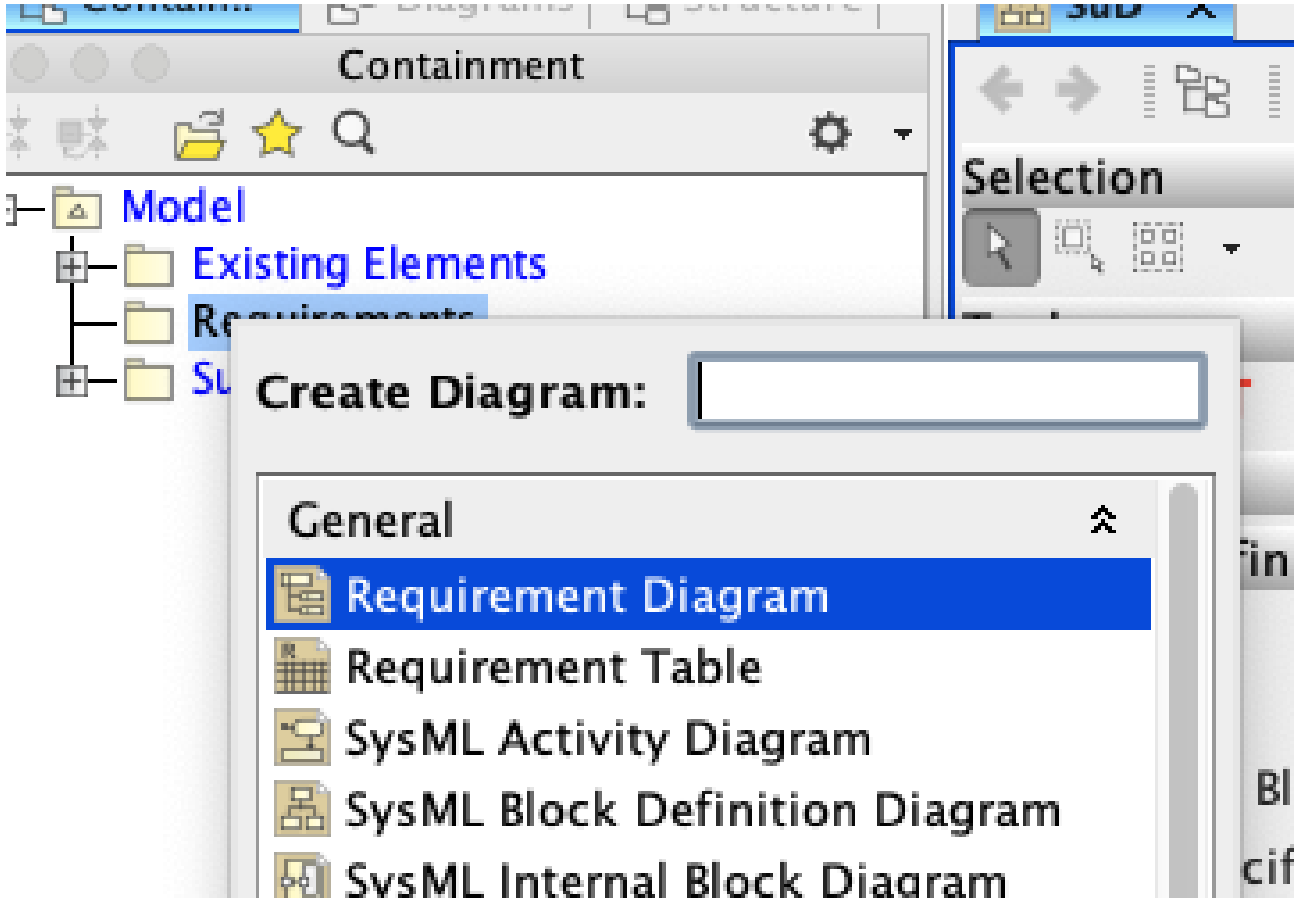
Check the sync





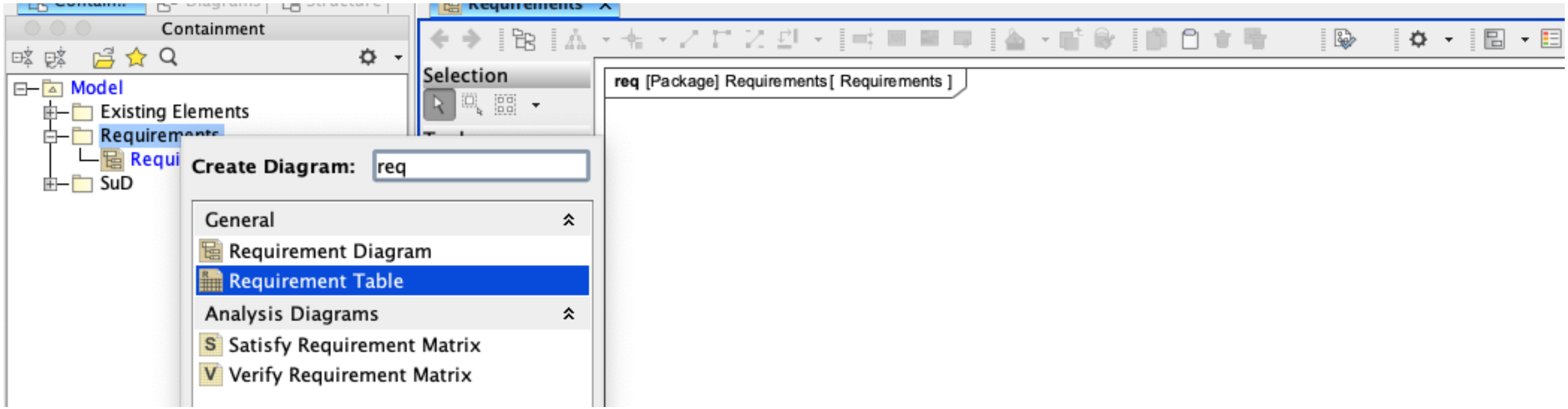
3) Requirements

- Create a requirement diagram



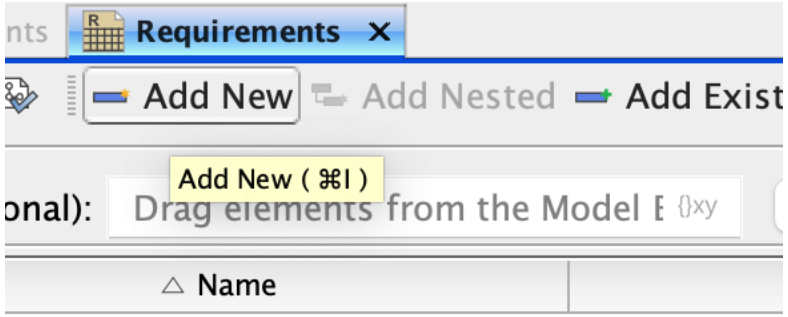
3) Requirements

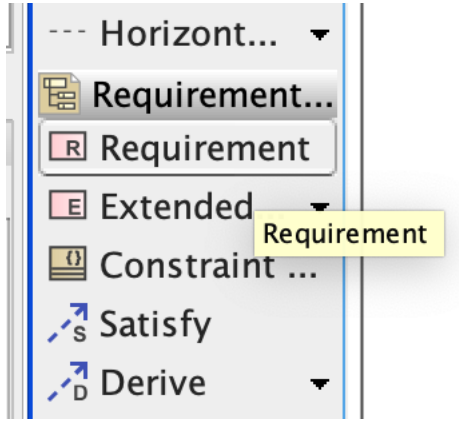
- Create a Requirement Table



Creating requirements

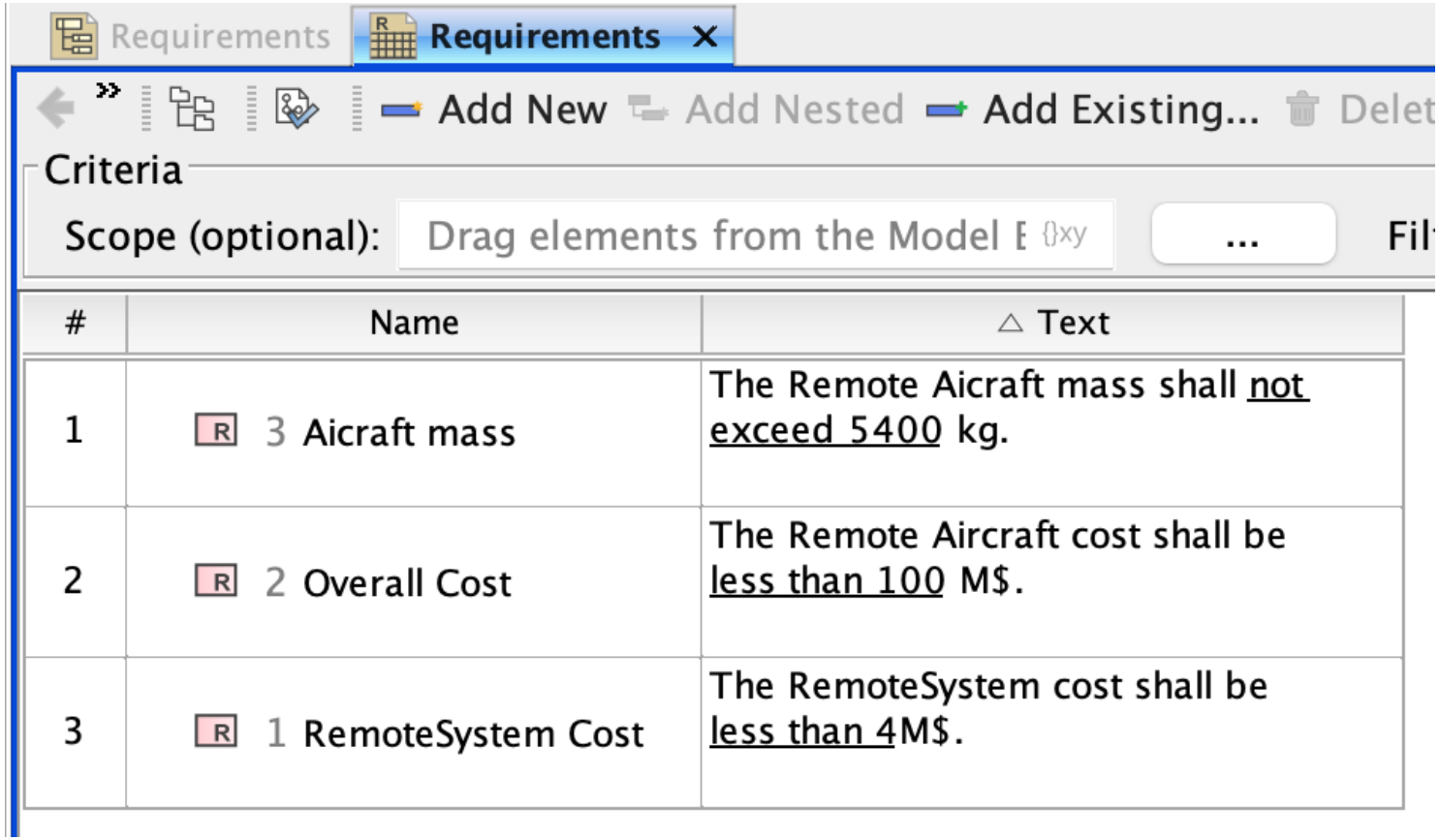
- **Create into table**
- Create into the diagram
- Copy and paste from excel.





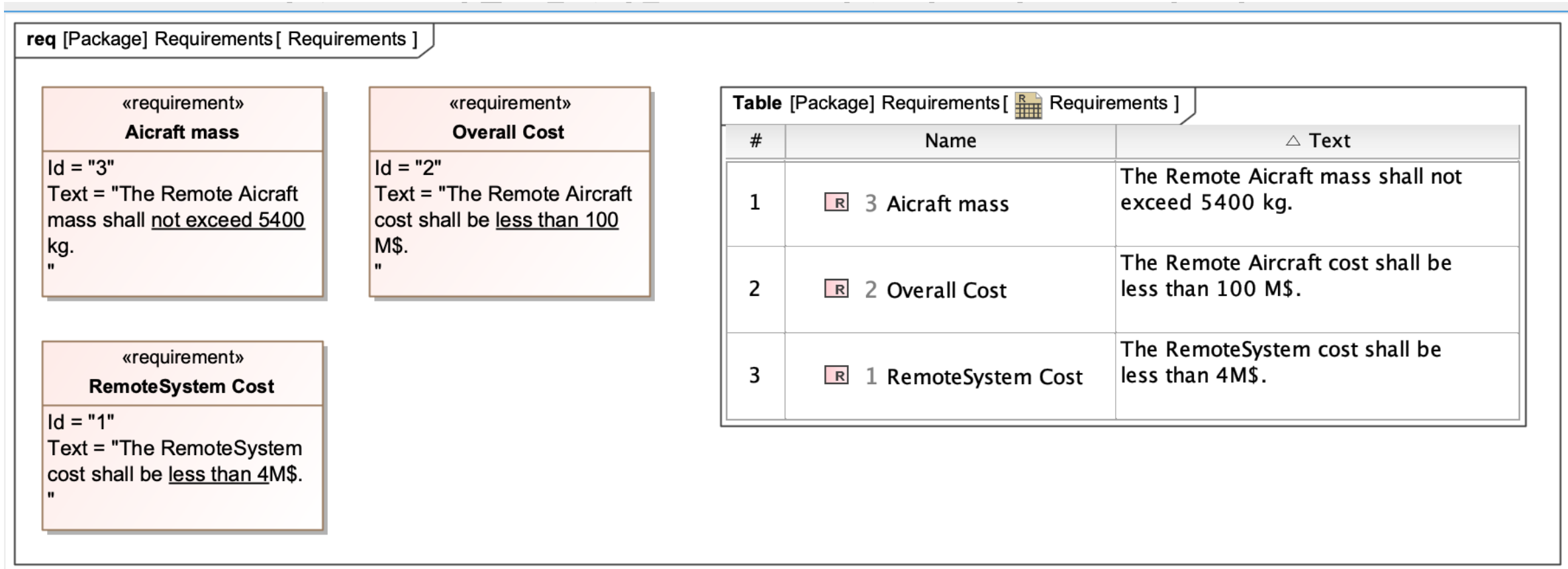
3) Requirements

- Create the requirements into the table view (could be on the requirement diagram as well) – addnew and fill it.

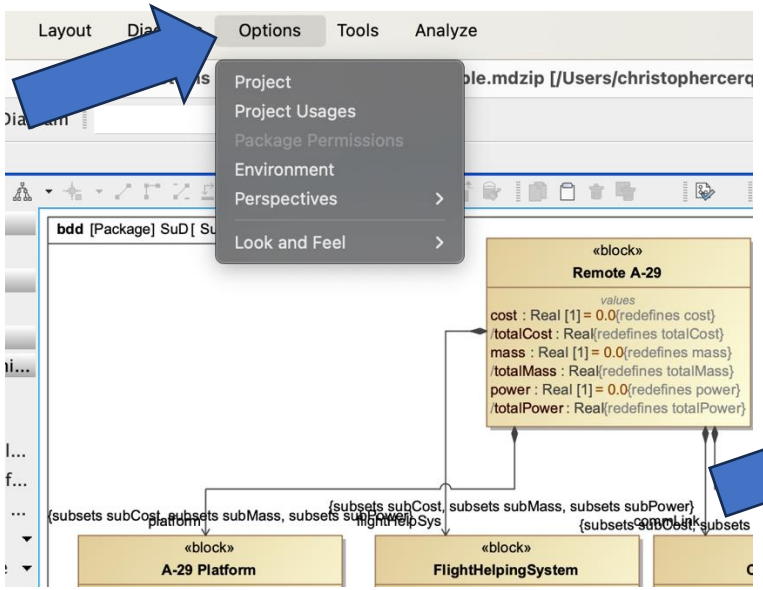


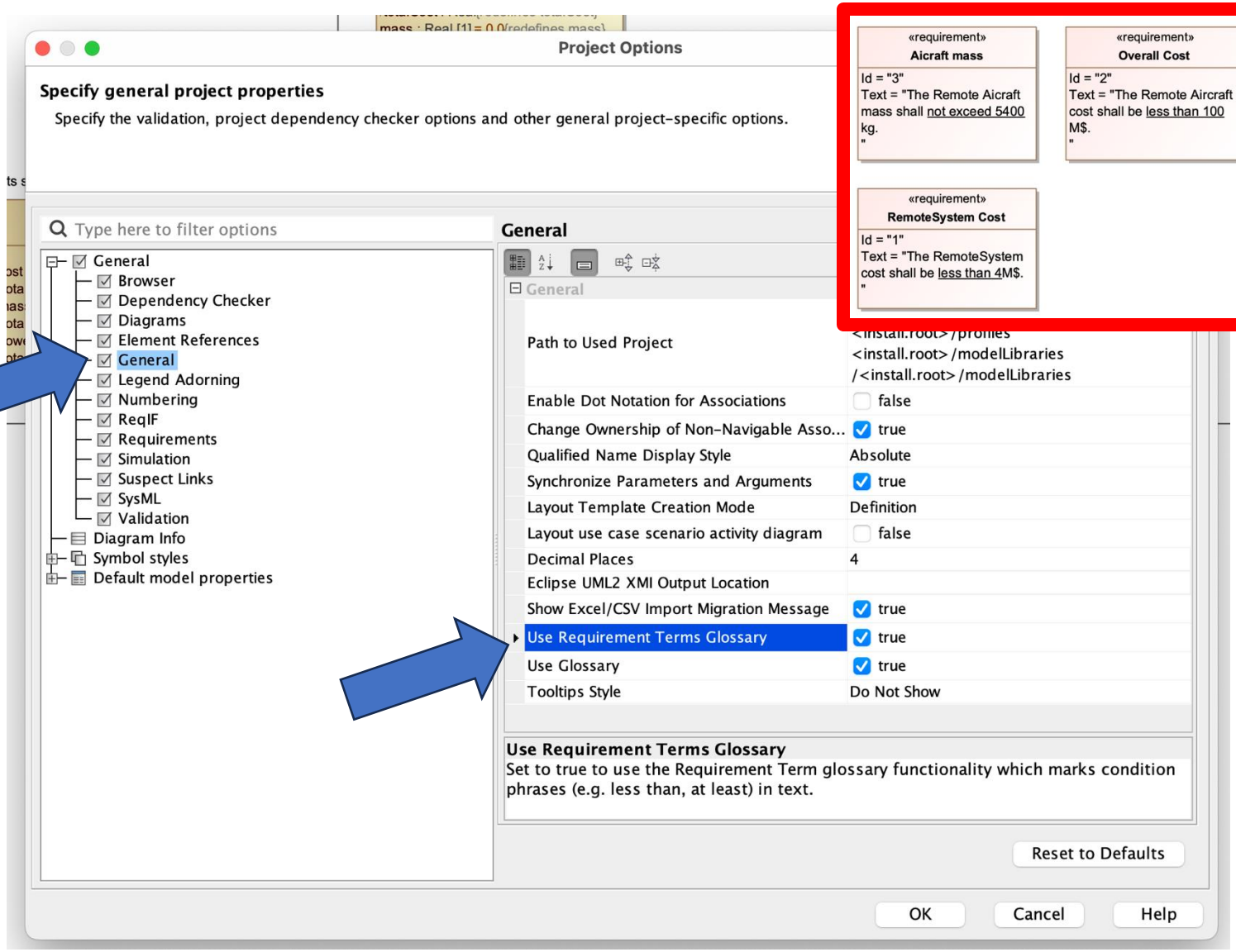
3) Requirements

- Check the requirements

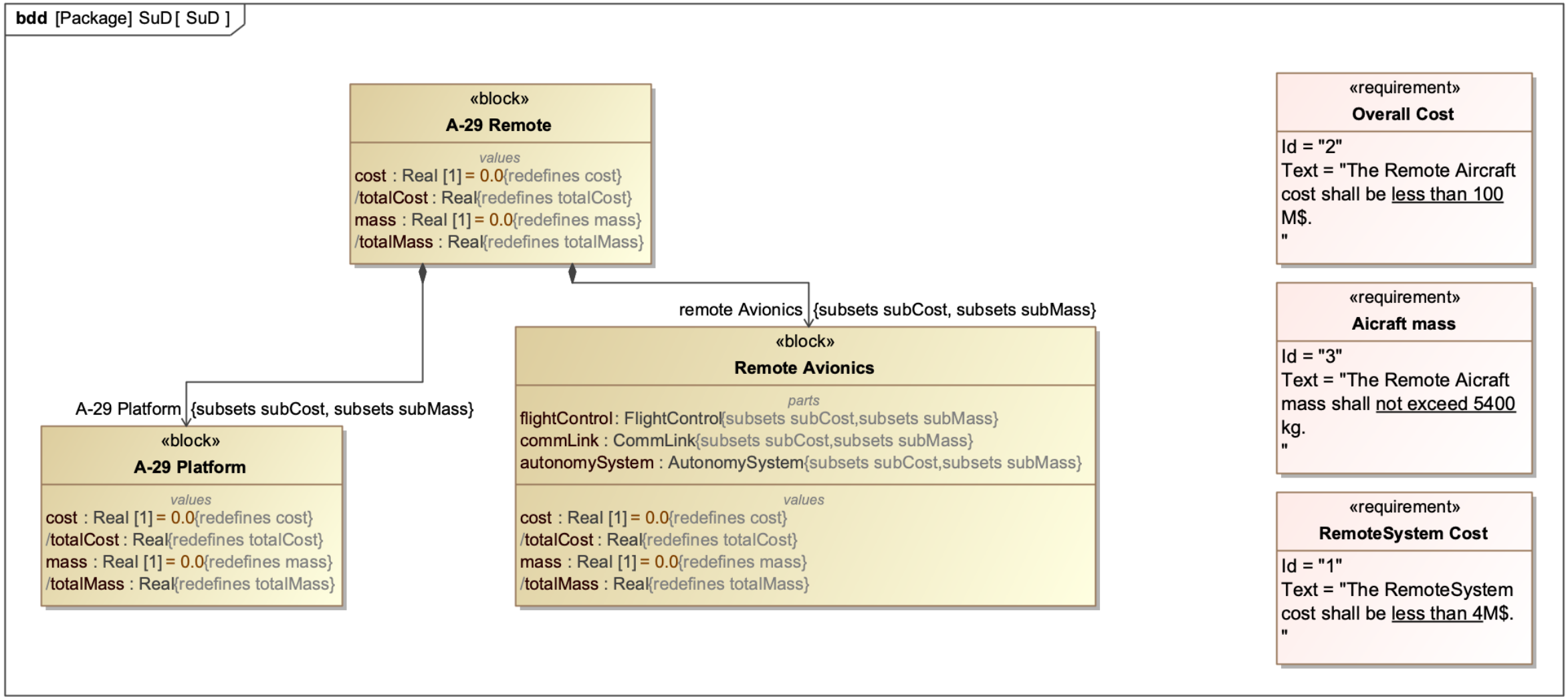


Enable requirement natural language processing



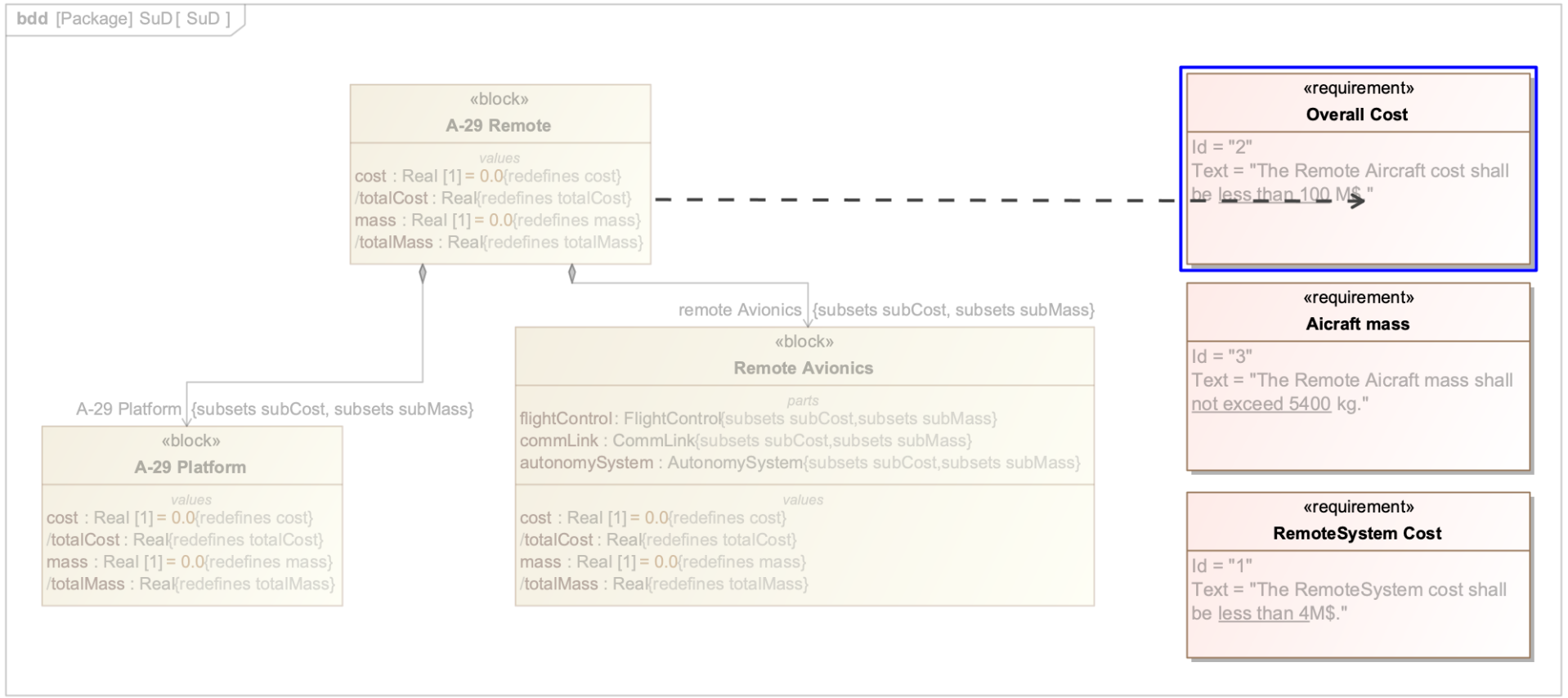


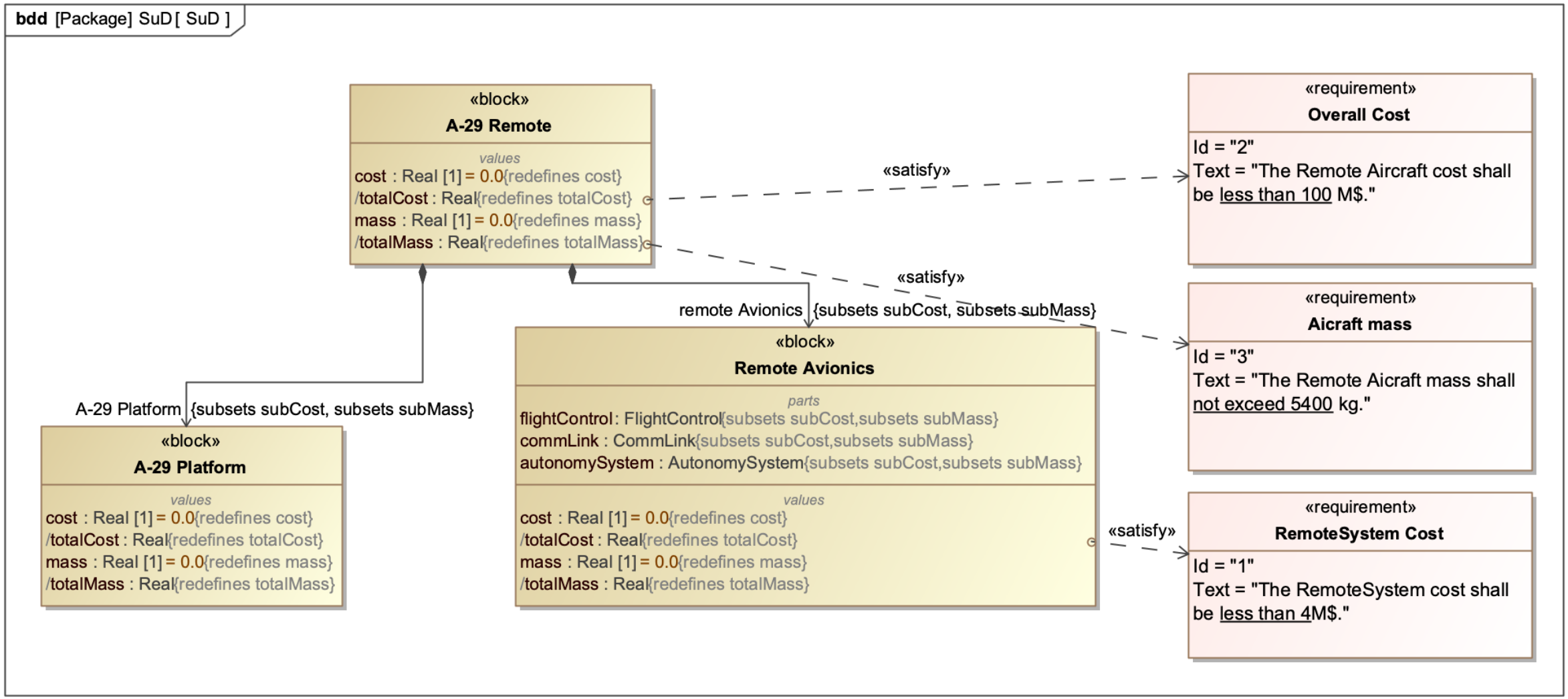
Connect the requirement w/ the architecture



Relate total<values> to the requirements

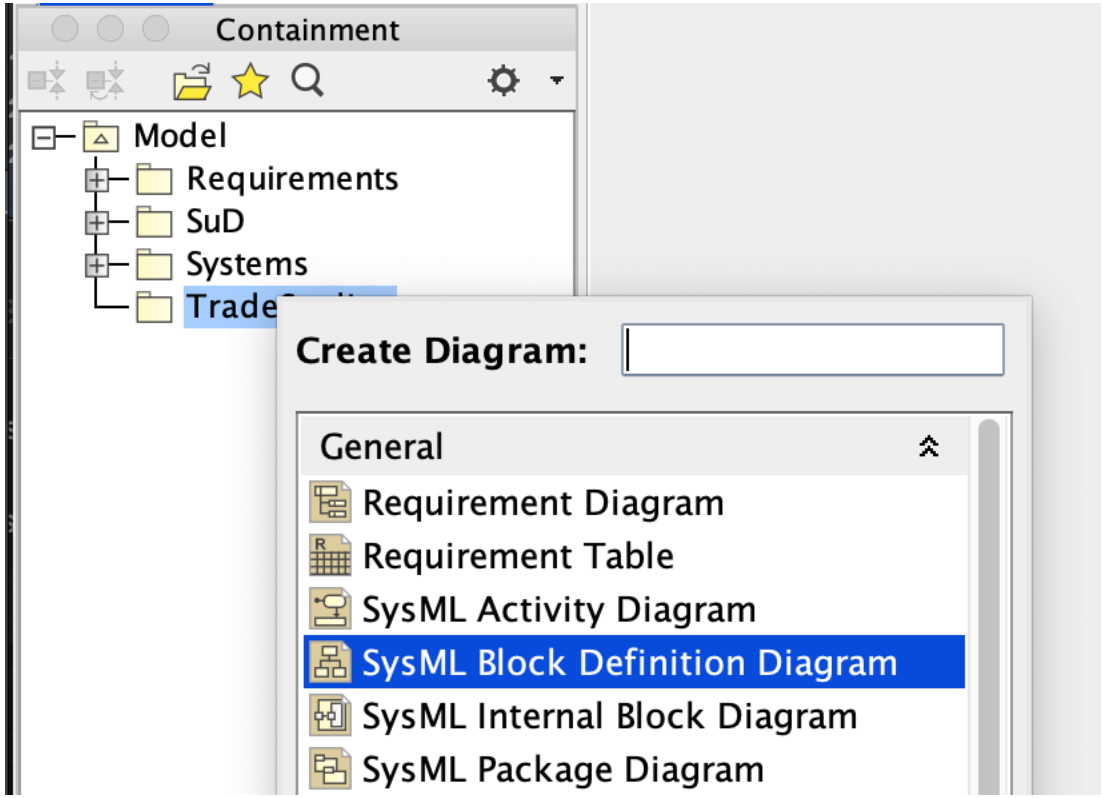
- Select a total<value> and link w/ the corresponding requirement: ex.: totalCost -> overall Cost



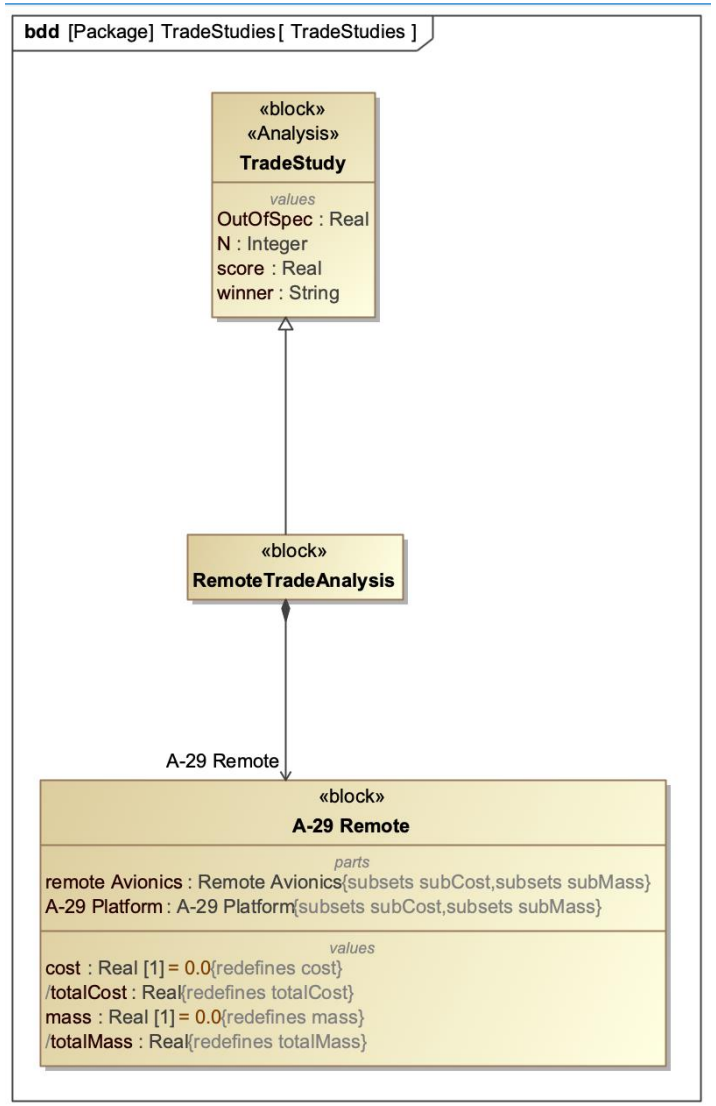

bdd [Package] SuD [SuD]
«block»
A-29 Remote
values
cost : Real [1] = 0.0{redefines cost}
/totalCost : Real{redefines totalCost}
mass : Real [1] = 0.0{redefines mass}
/totalMass : Real{redefines totalMass}
«satisfy»
«requirement»
Overall Cost
Id = "2"
Text = "The Remote Aircraft cost shall be less than 100 M$."
«satisfy»
remote Avionics {subsets subCost, subsets subMass}
«requirement»
Aicraft mass
Id = "3"
Text = "The Remote Aicraft mass shall not exceed 5400 kg."
A-29 Platform {subsets subCost, subsets subMass}
«block»
A-29 Platform
values
cost : Real [1] = 0.0{redefines cost}
/totalCost : Real{redefines totalCost}
mass : Real [1] = 0.0{redefines mass}
/totalMass : Real{redefines totalMass}
«block»
Remote Avionics
parts
flightControl : FlightControl{subsets subCost,subsets subMass}
commLink : CommLink{subsets subCost,subsets subMass}
autonomySystem : AutonomySystem{subsets subCost,subsets subMass}
values
cost : Real [1] = 0.0{redefines cost}
/totalCost : Real{redefines totalCost}
mass : Real [1] = 0.0{redefines mass}
/totalMass : Real{redefines totalMass}
«satisfy»
«requirement»
RemoteSystem Cost
Id = "1"
Text = "The RemoteSystem cost shall be less than 4M$."

4) Start the trade study analysis

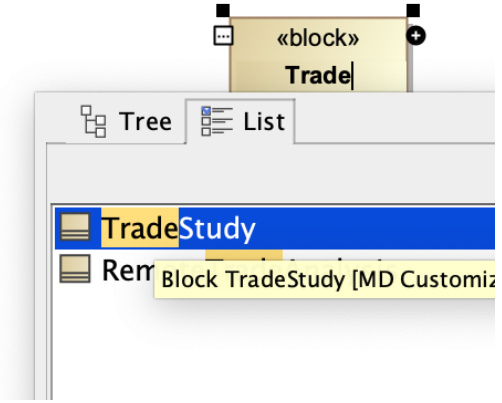
- Create the trade package
- Create the bdd from the package



Create the trade study



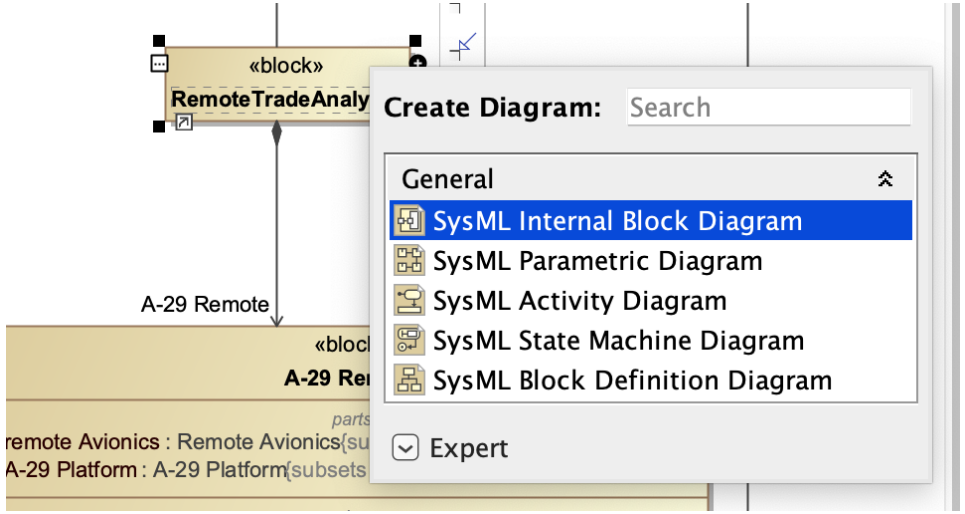
3) Create a block and name it TradeStudy. It will bring the stereotype.

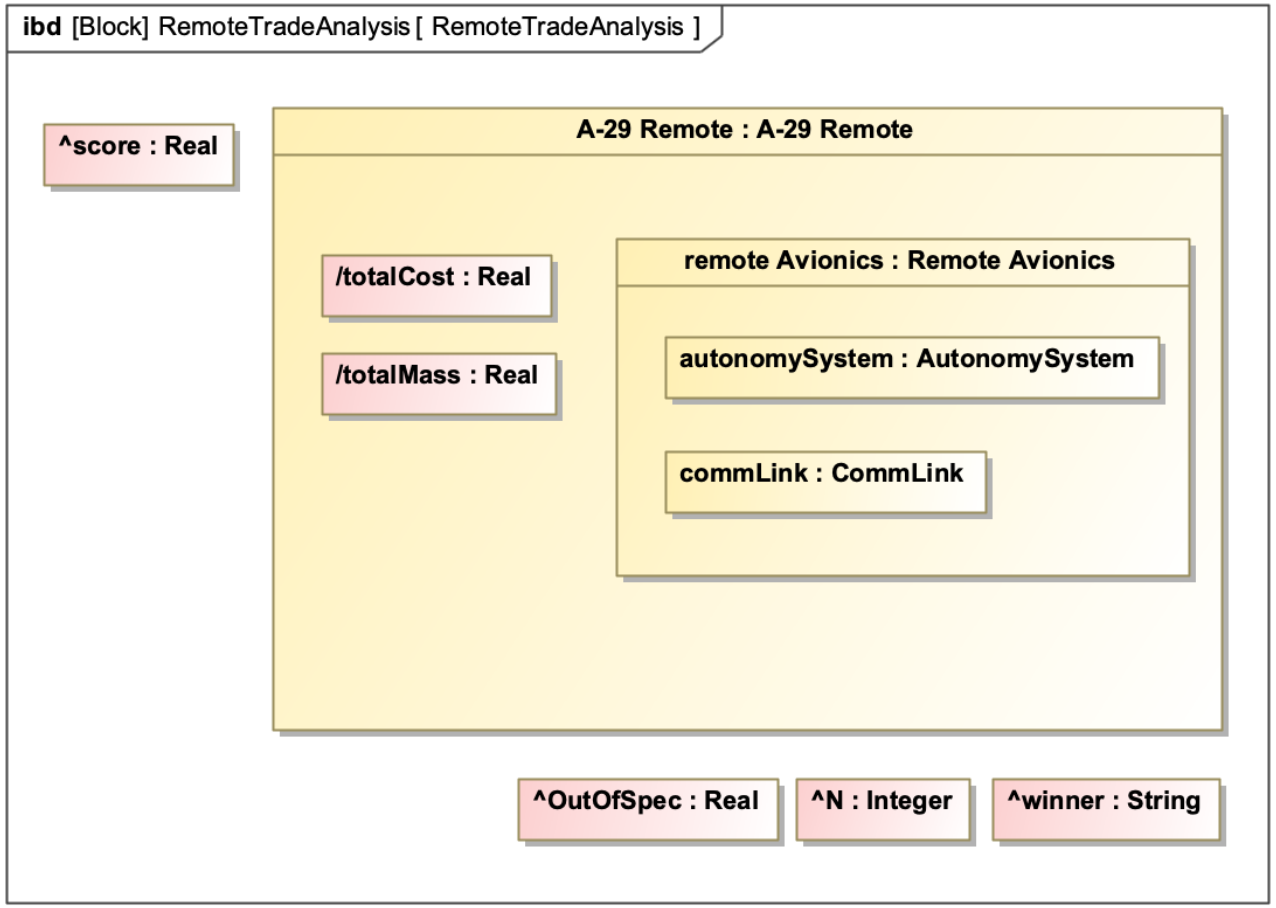


2) Create the analysis context

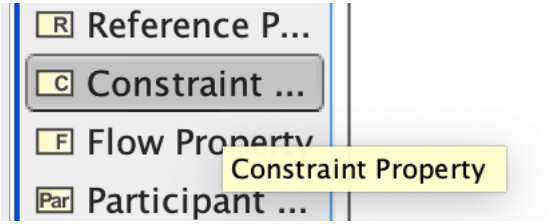
1) Drag the the system that will be trade

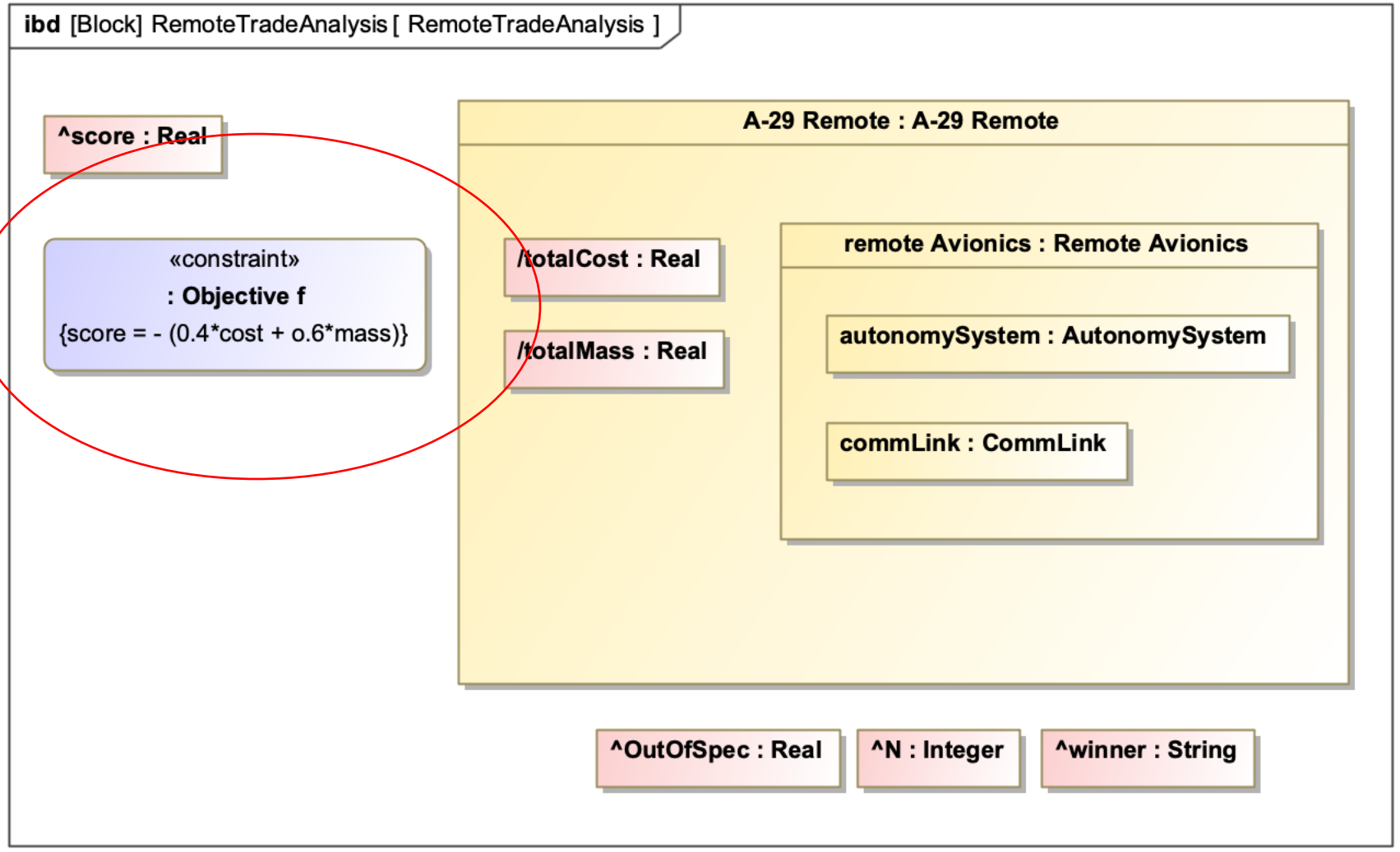
Results of creating the IBD of the trade analysis



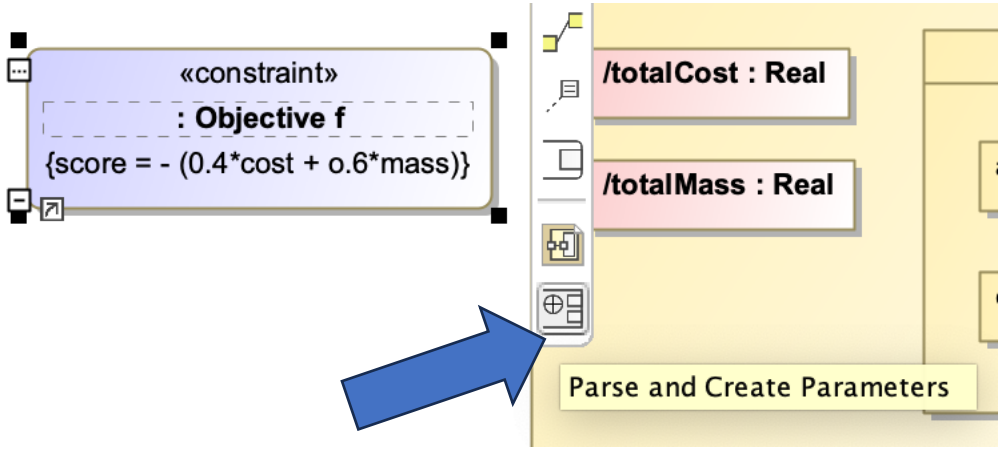


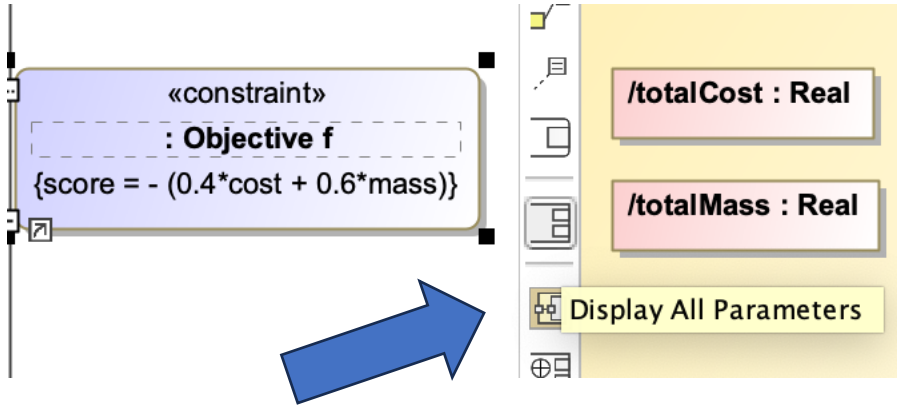
Create a constraint of the object function

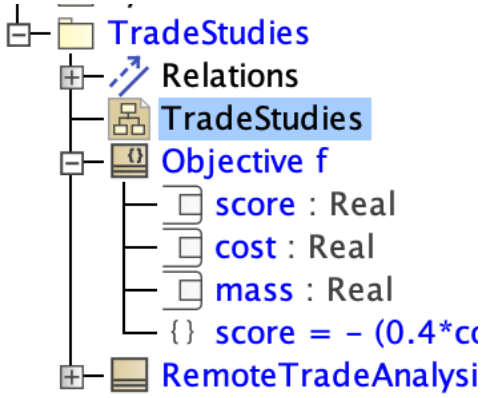


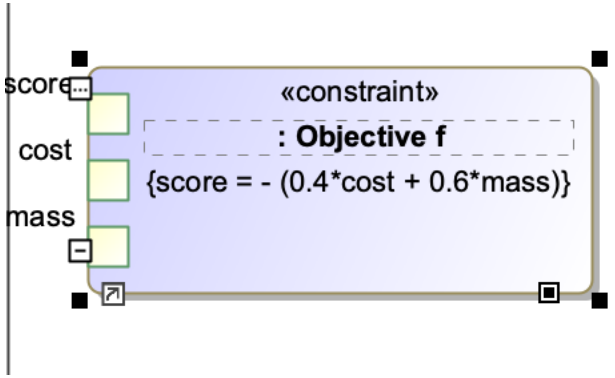


Show the parameter ports

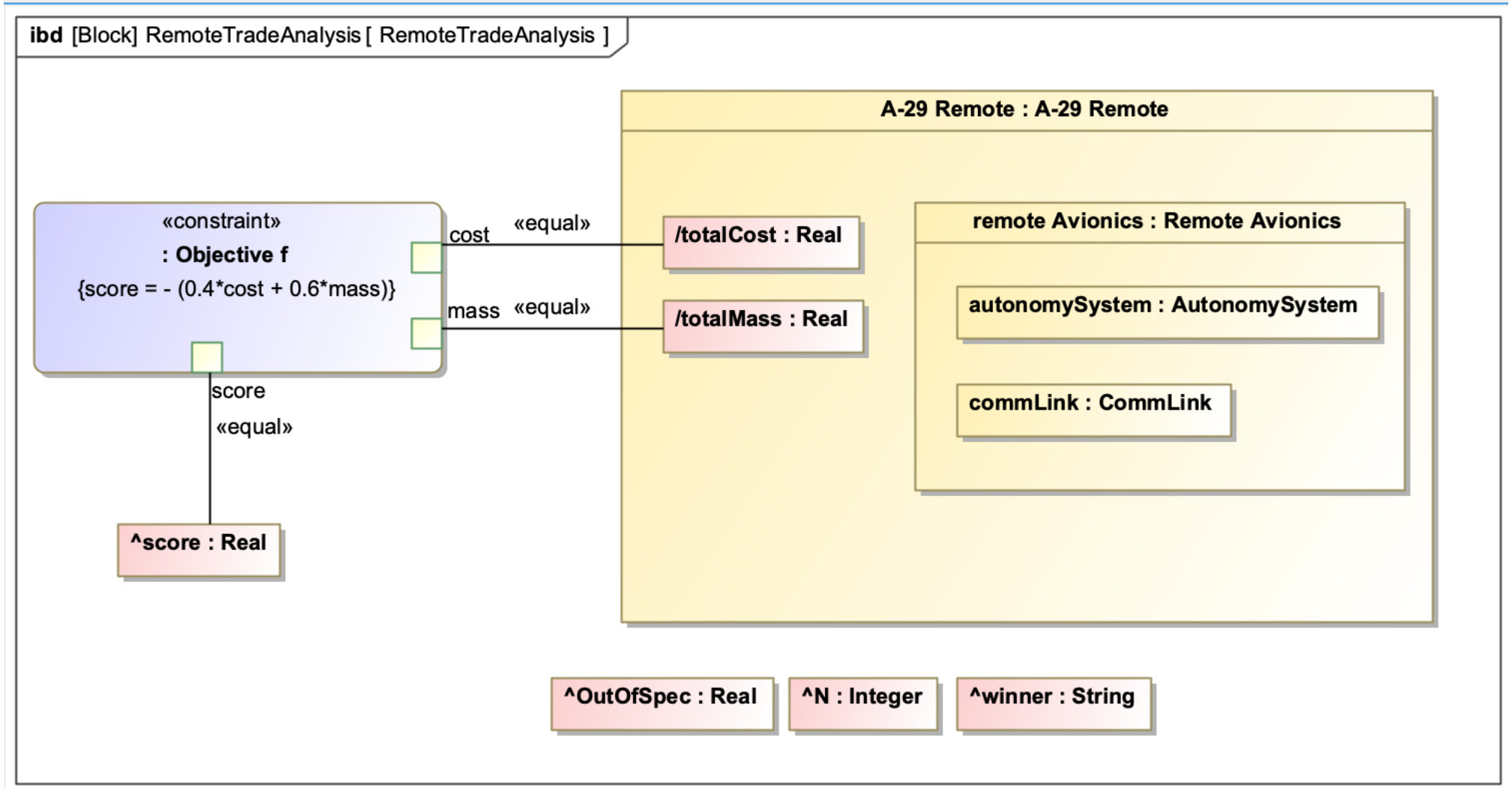








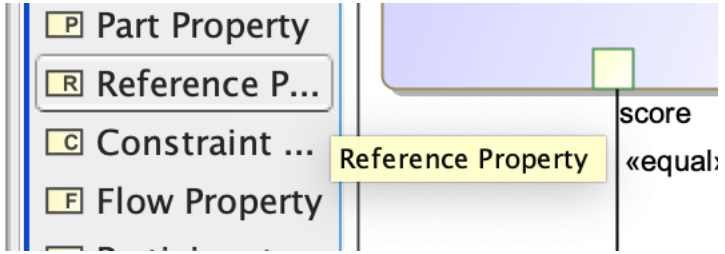
Interconnect w/ the architecture elements

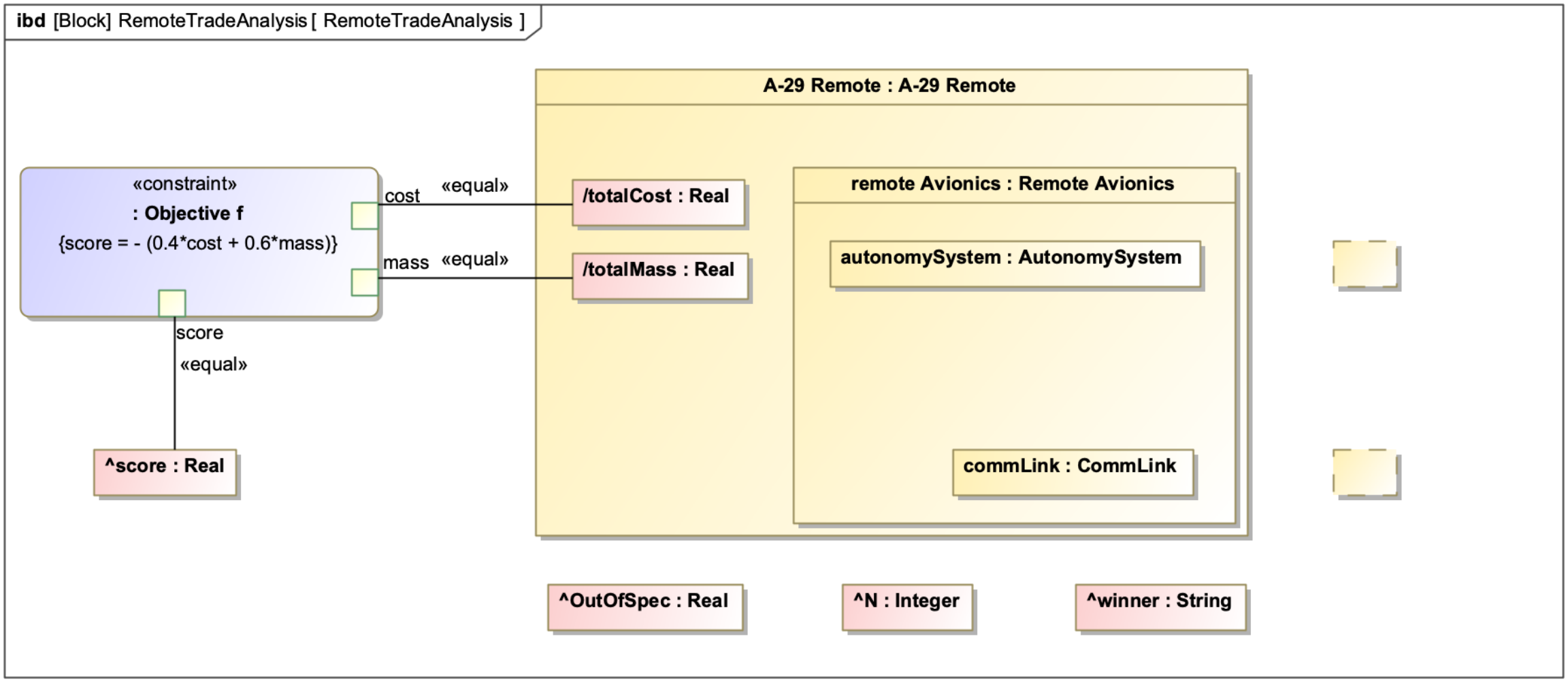


5) Setting up the alternatives

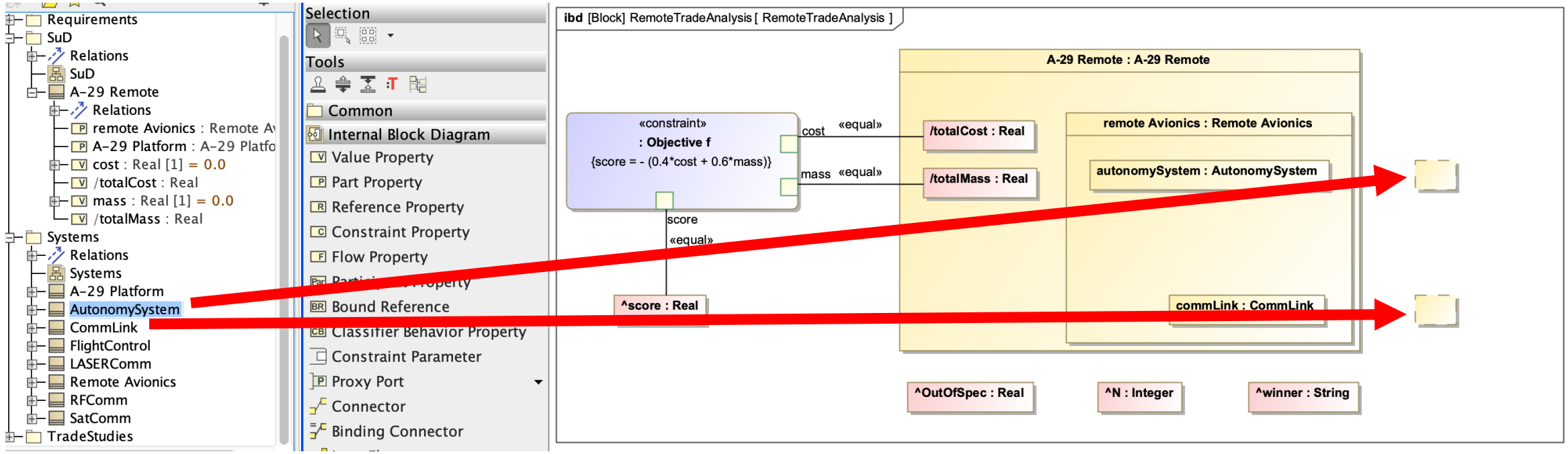
- Via the possible specializations
- Via a excel file

Create source references..

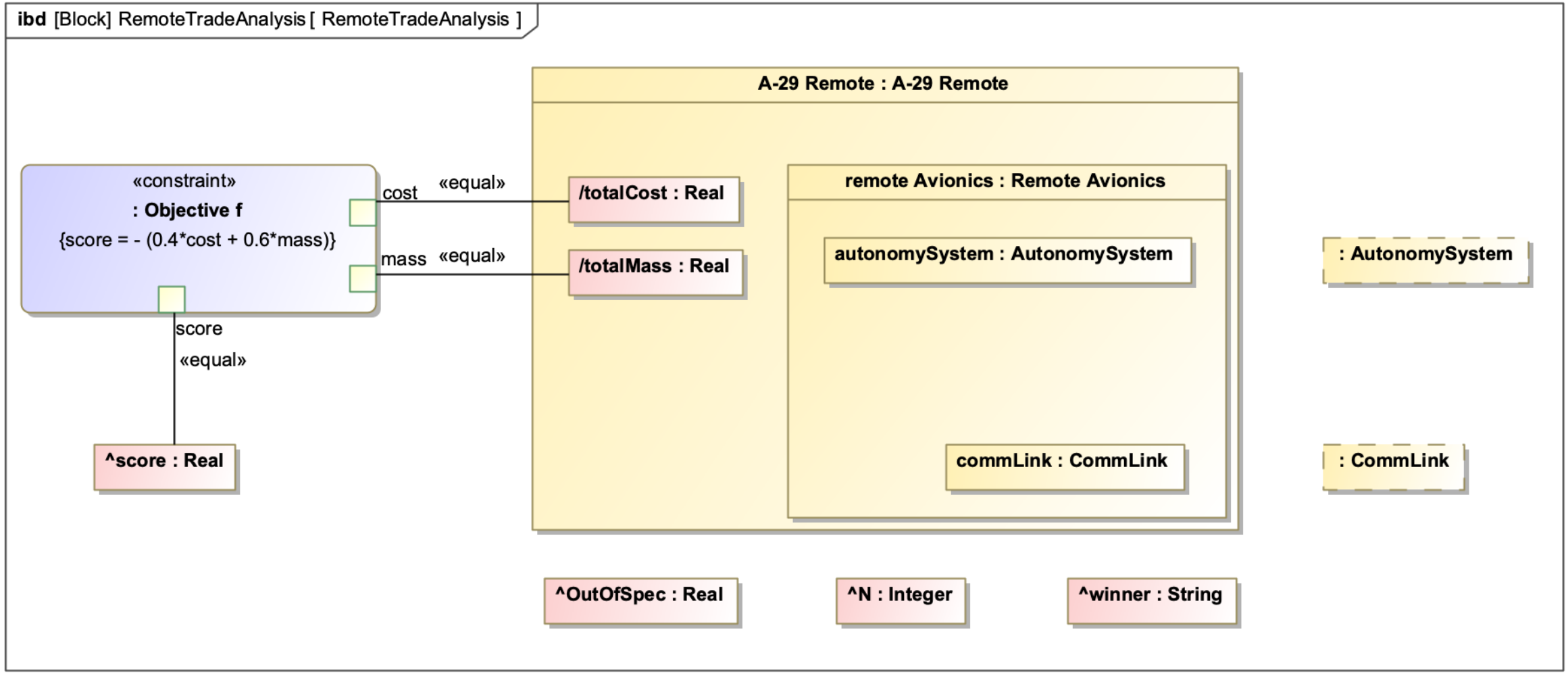




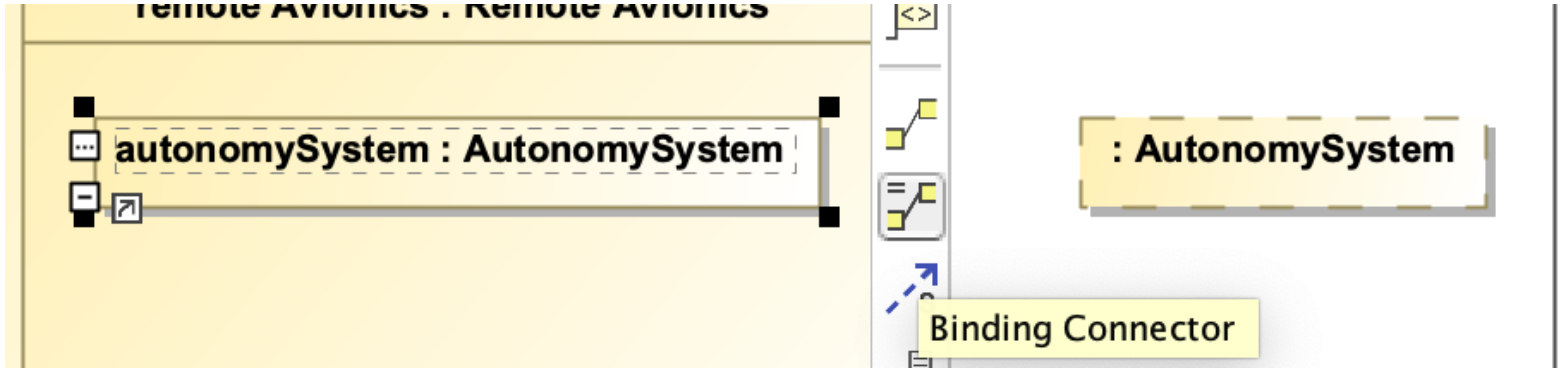
Interconnect the block w/ the reference

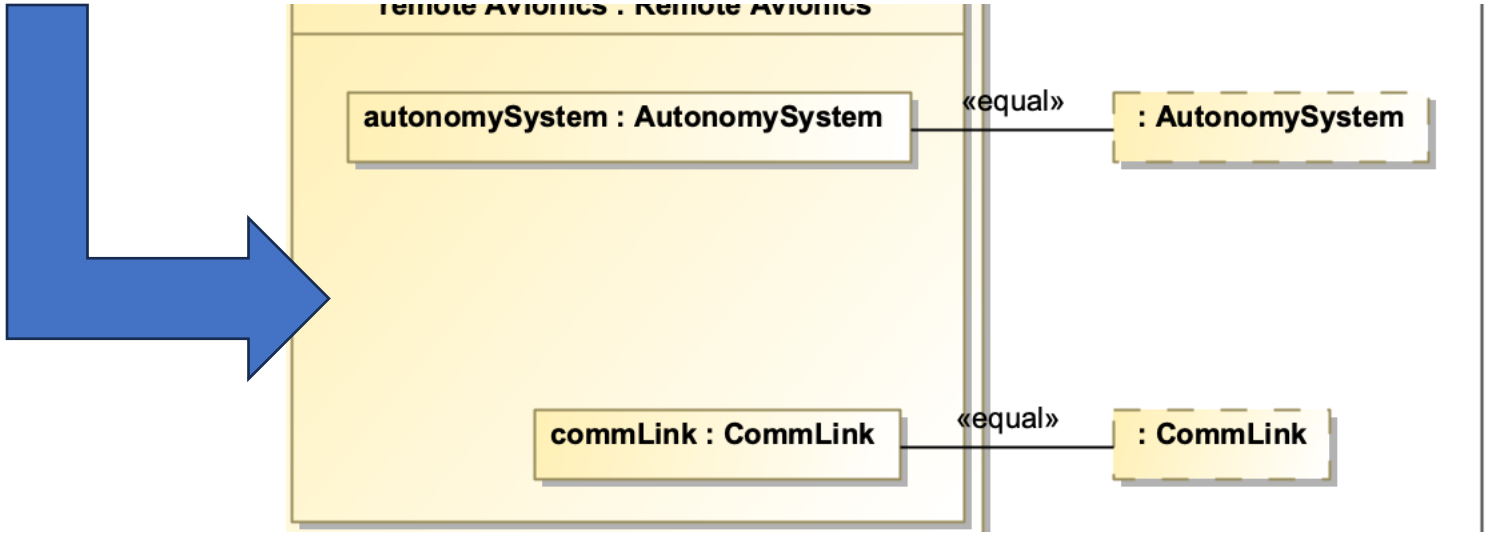


How it shows

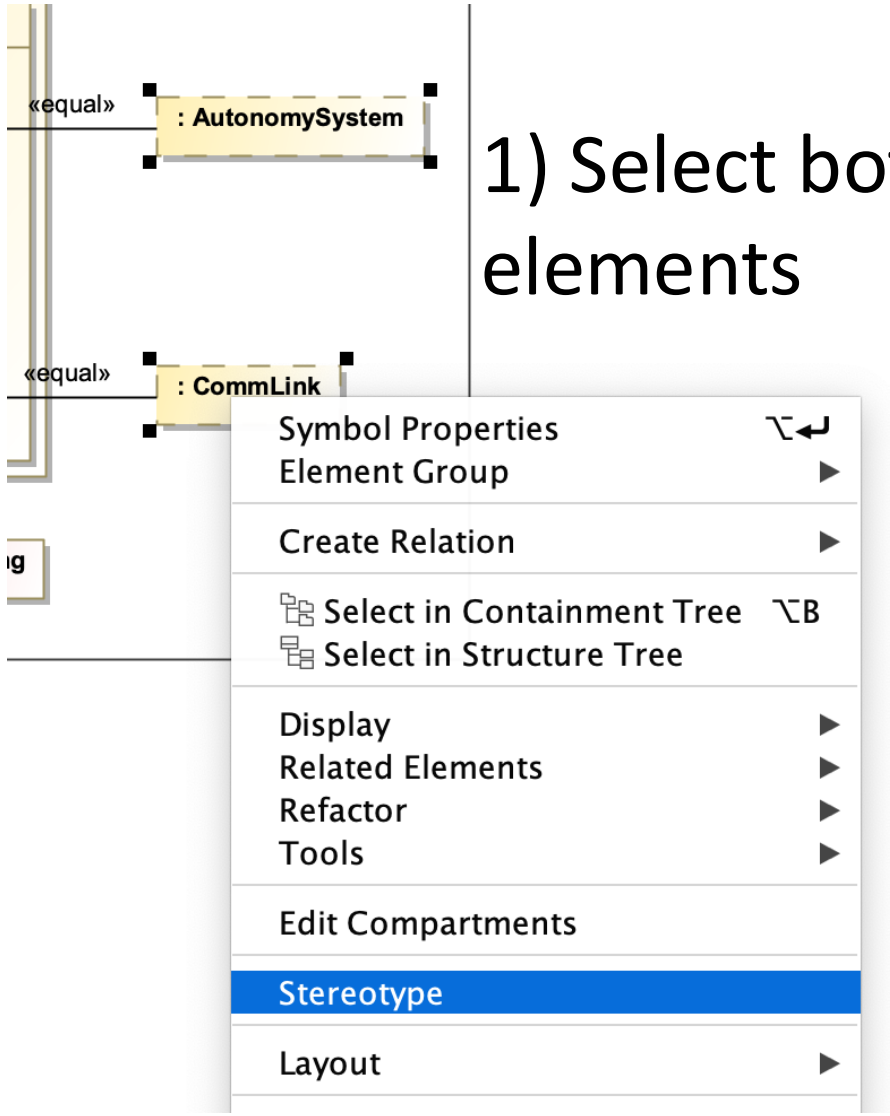

ibd [Block] RemoteTradeAnalysis [RemoteTradeAnalysis]
A-29 Remote : A-29 Remote
«constraint»
: Objective f
{score = - (0.4*cost + 0.6*mass)}
cost
«equal»
mass
«equal»
score
«equal»
/totalCost : Real
/totalMass : Real
remote Avionics : Remote Avionics
autonomySystem : AutonomySystem
commLink : CommLink
^score : Real
: AutonomySystem
: CommLink
^OutOfSpec : Real
^N : Integer
^winner : String

Do a binding connection

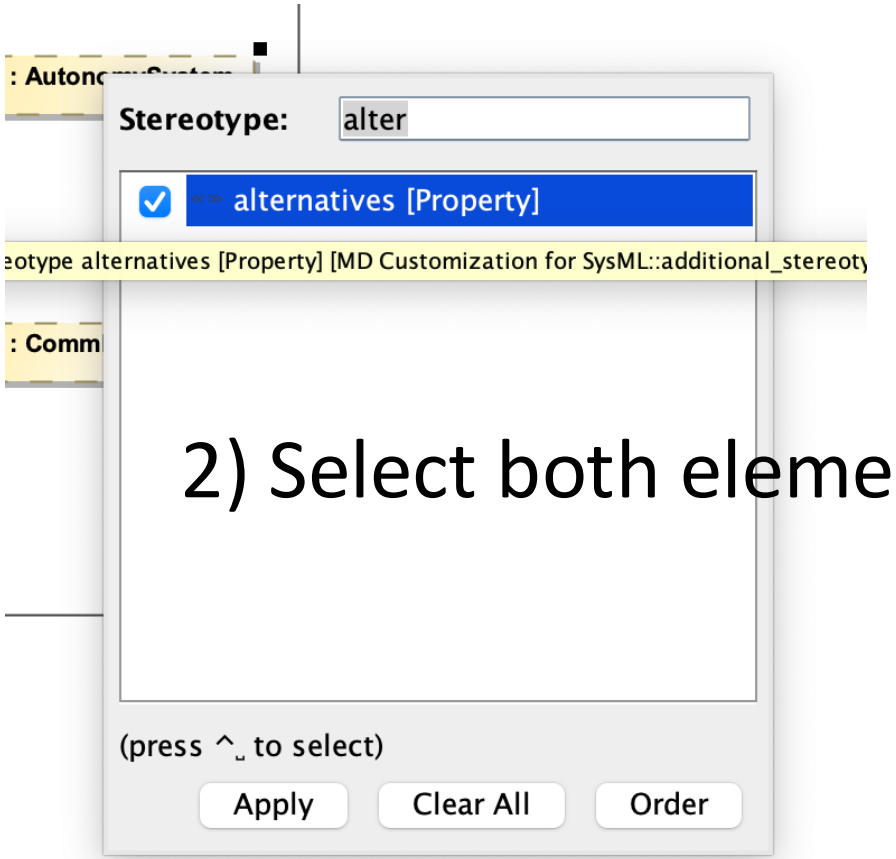




Change the stereotypes to alternatives

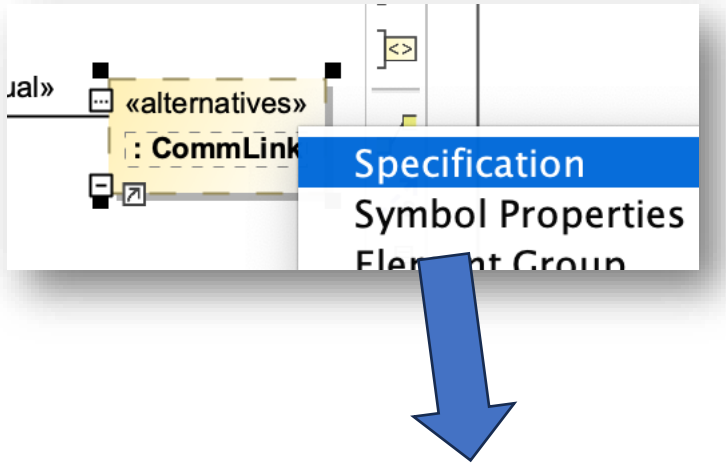


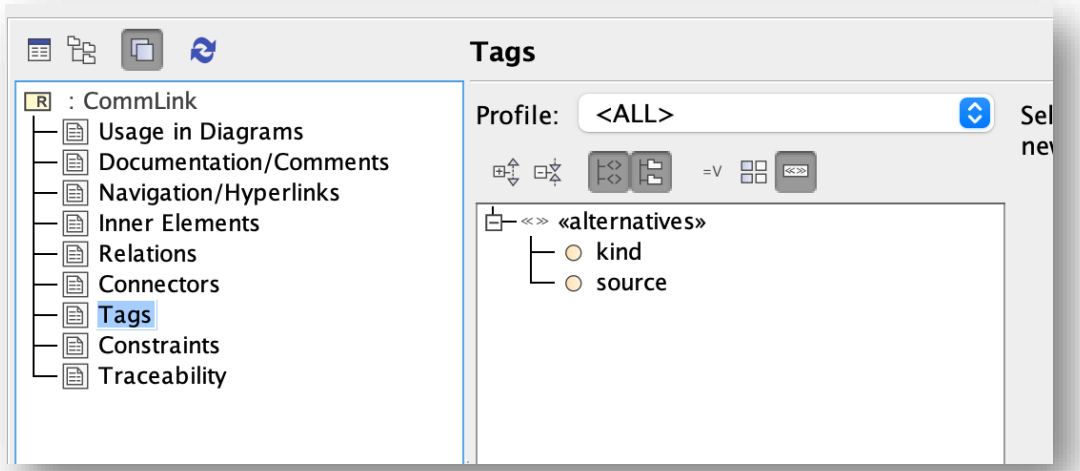
1) Select both elements

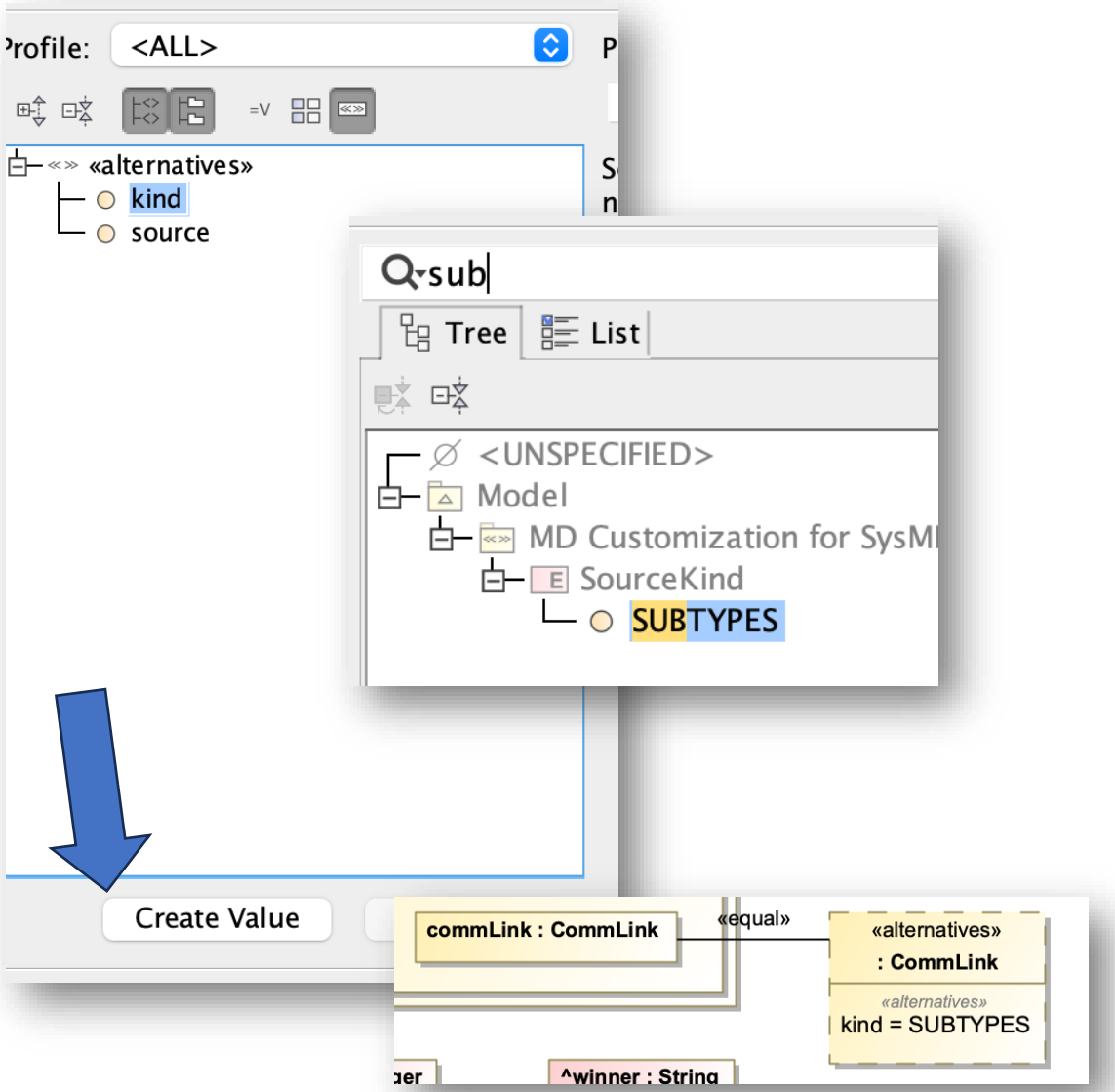


2) Select both elements

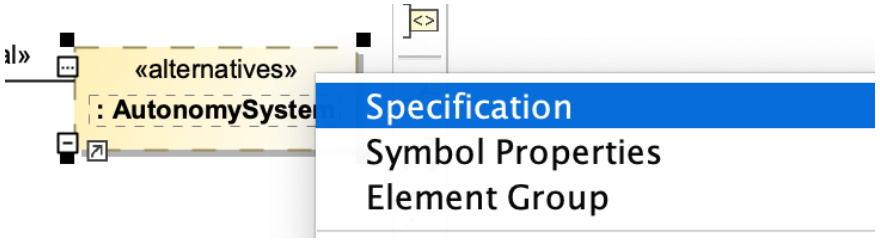
Link the specialization w/ the references

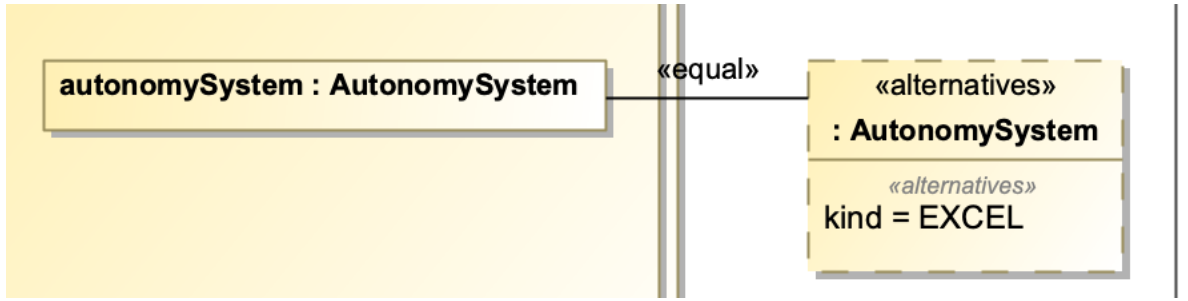


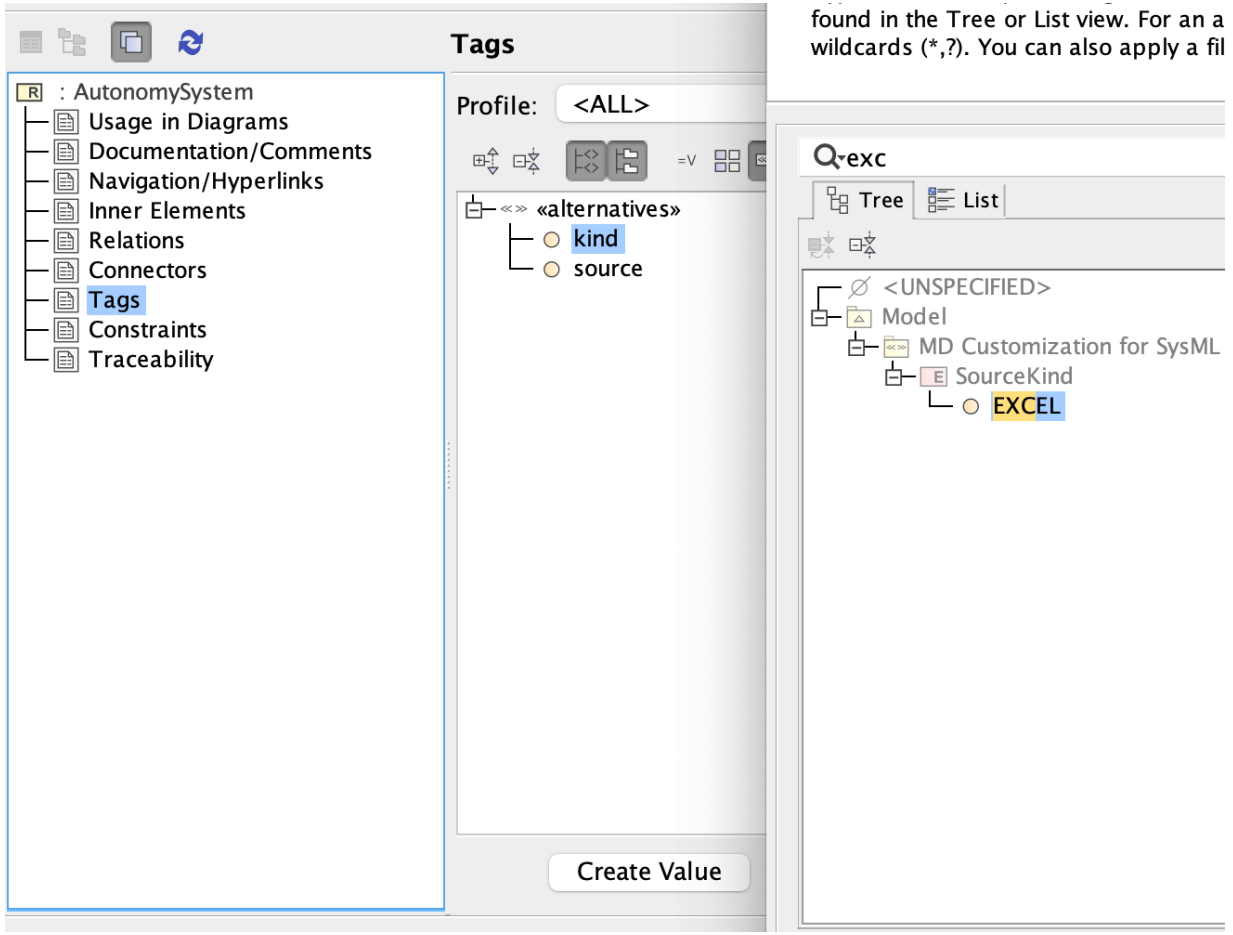




Now link to the excel.

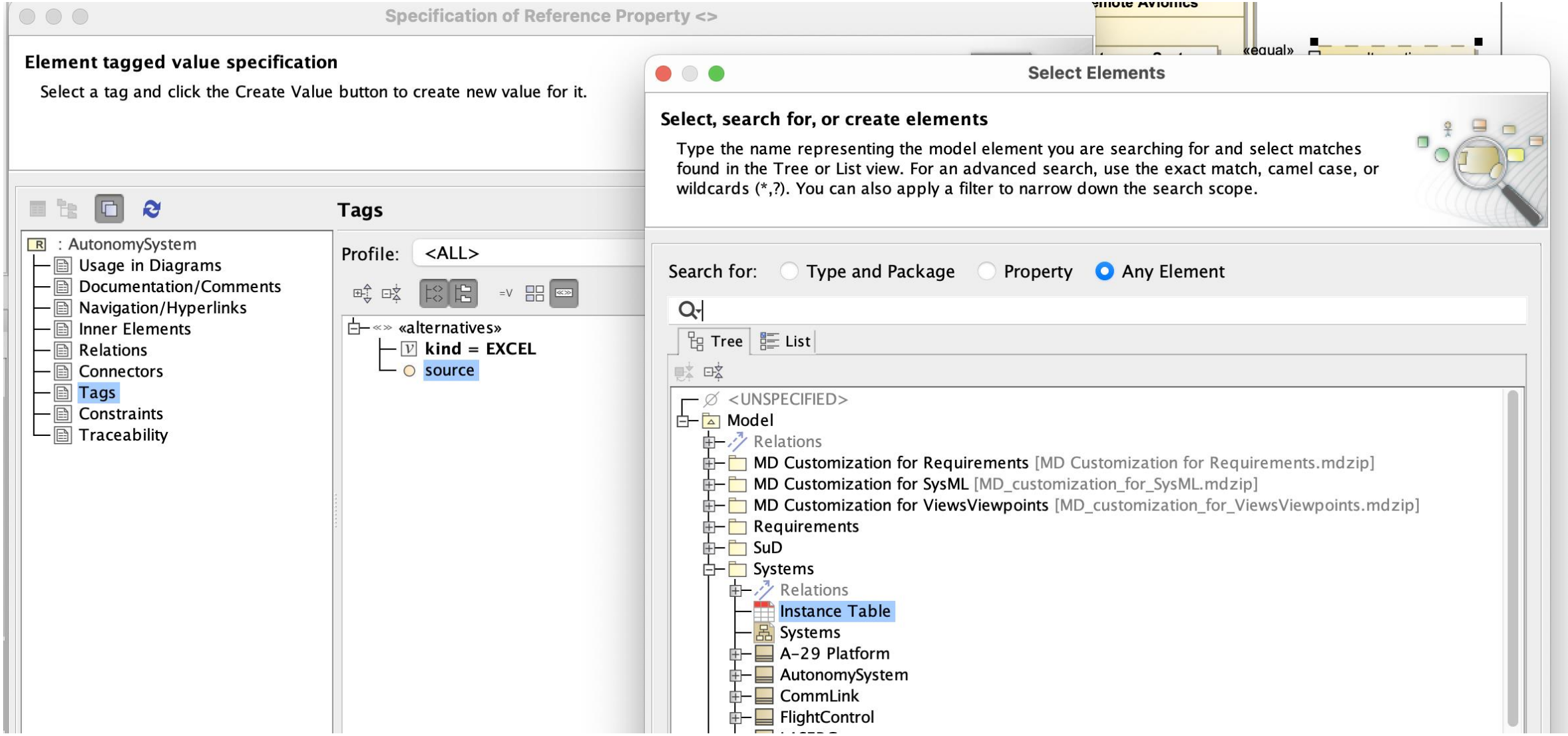




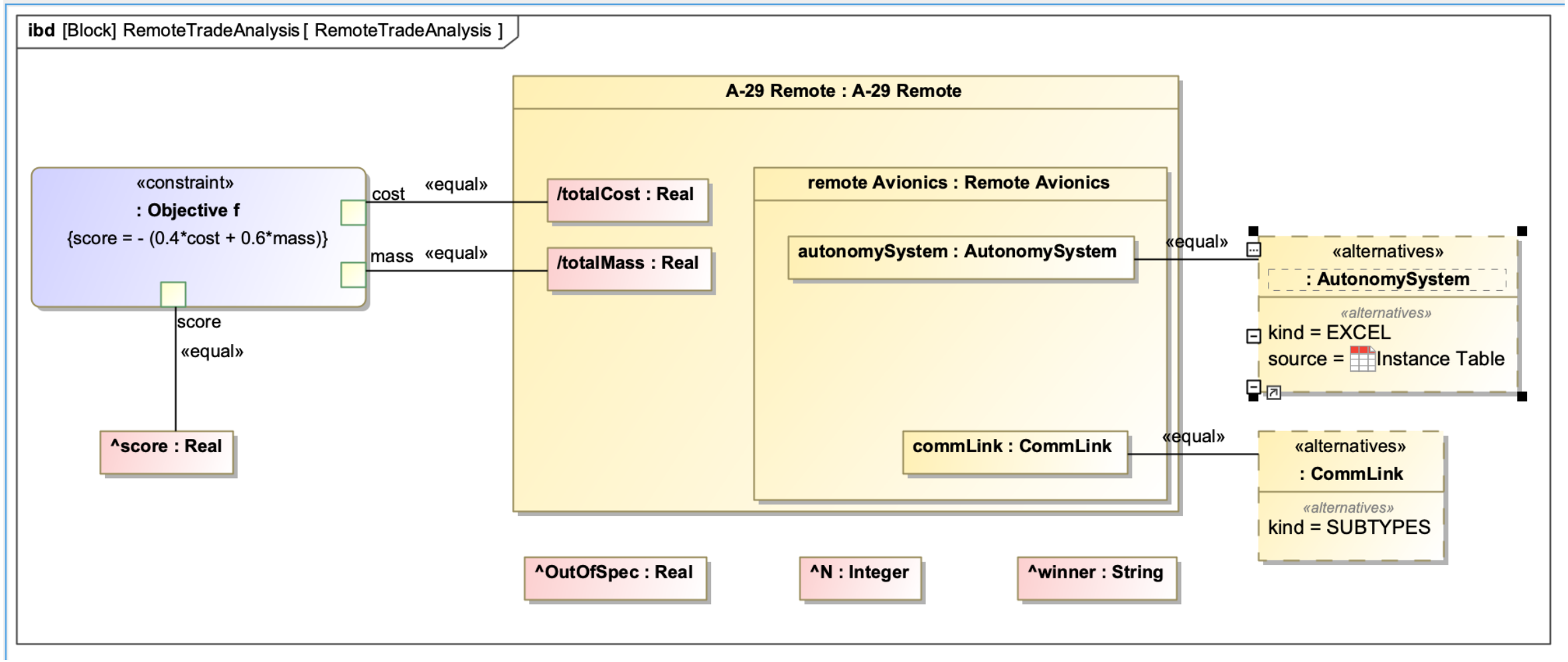


found in the Tree or List view. For an a
wildcards (*,?). You can also apply a fil

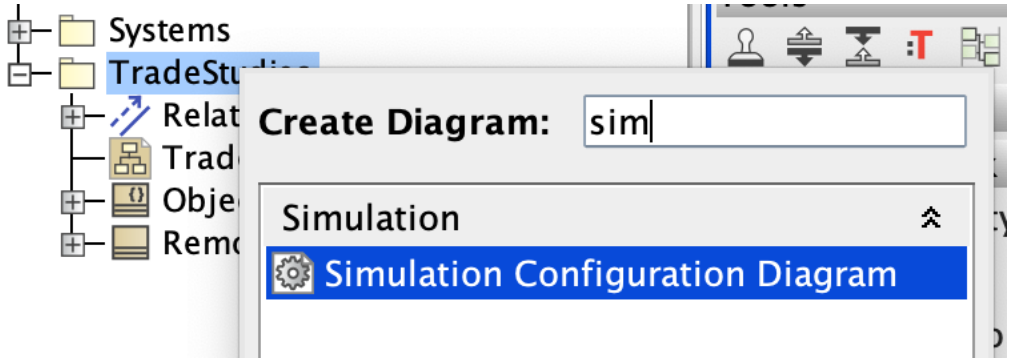
Adjust the source link w/ the excel - instance table of the autonomy

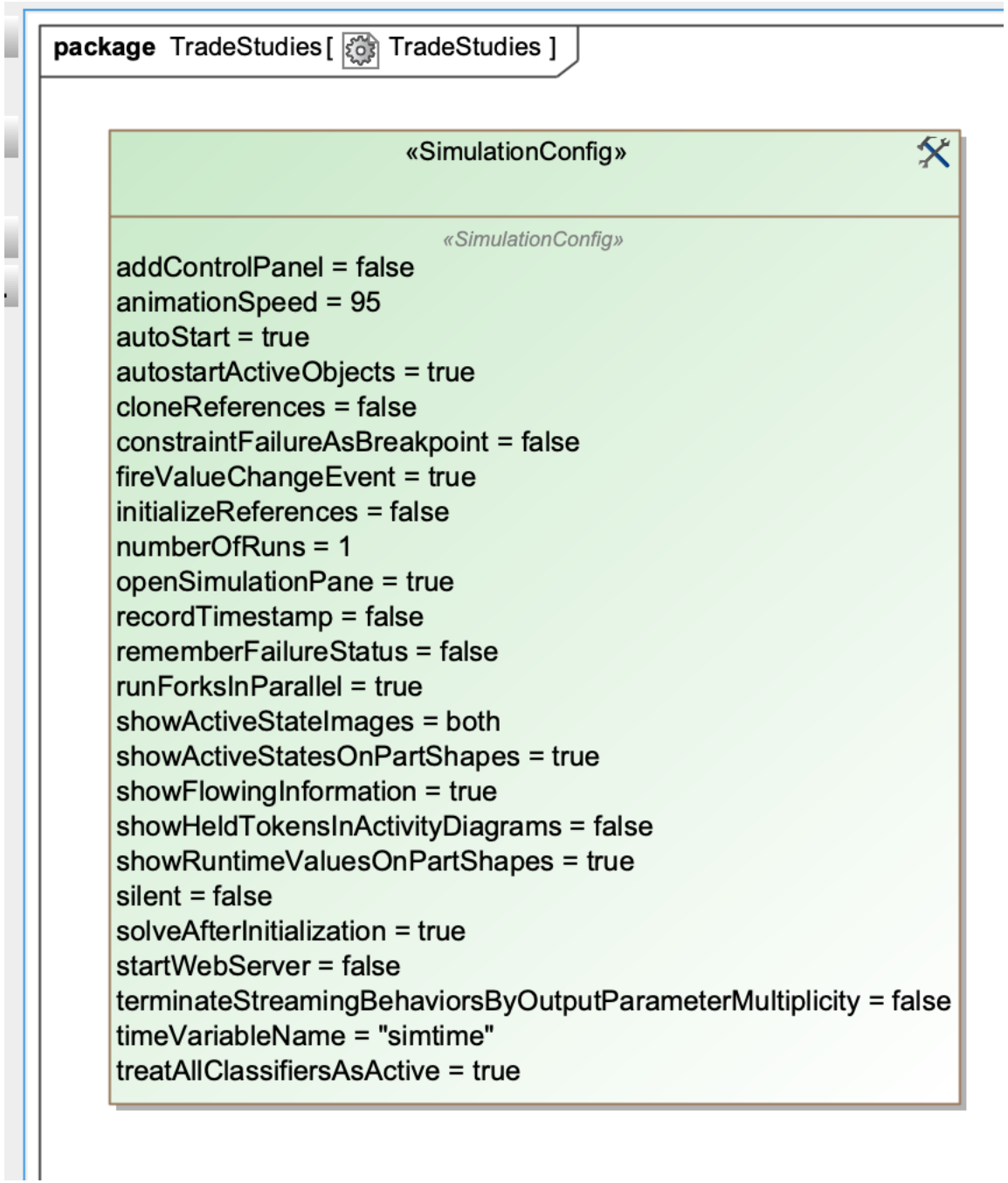


All settle

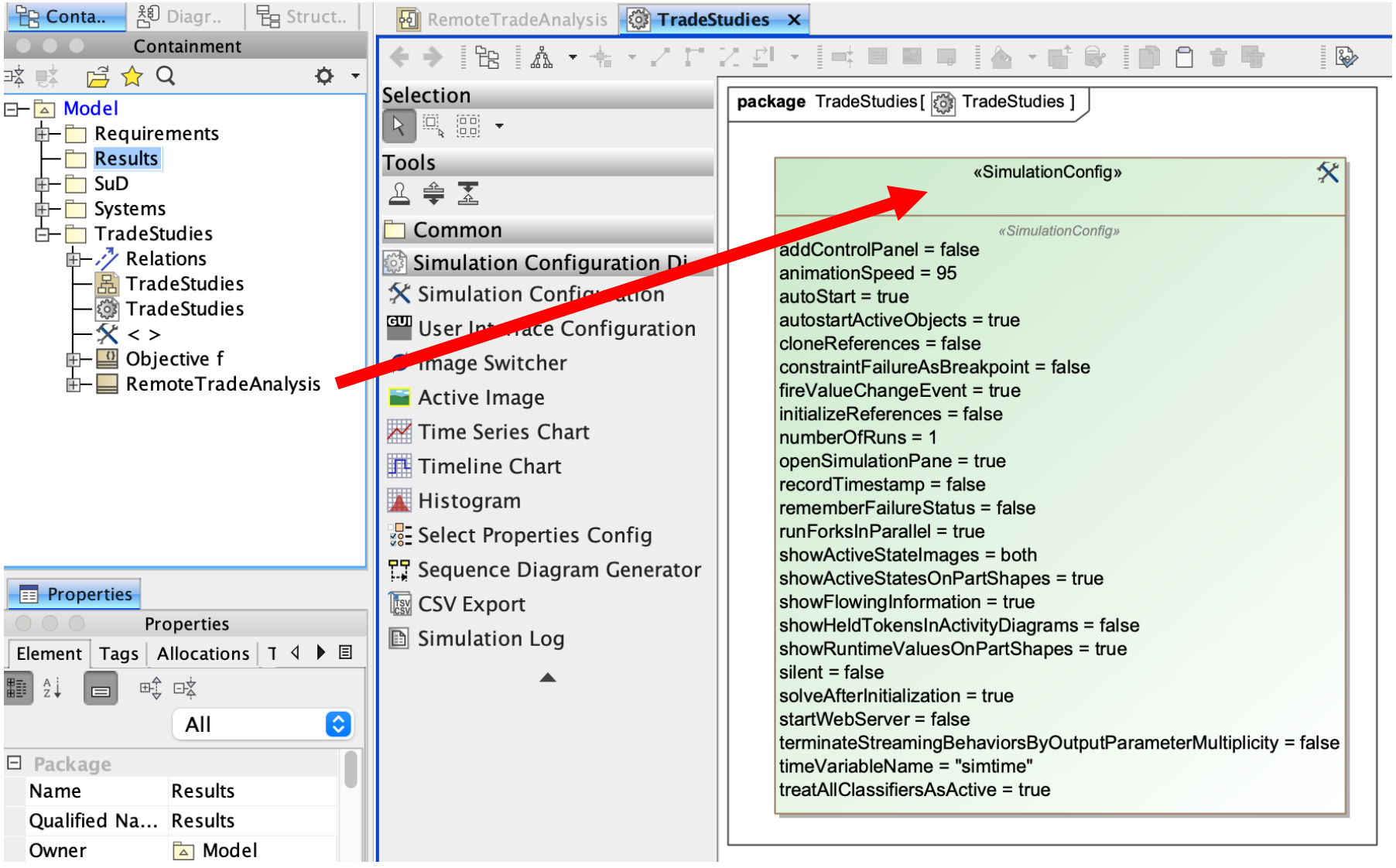

ibd [Block] RemoteTradeAnalysis [RemoteTradeAnalysis]
A-29 Remote : A-29 Remote
«constraint»
: Objective f
{score = - (0.4*cost + 0.6*mass)}
cost
«equal»
/totalCost : Real
mass
«equal»
/totalMass : Real
score
«equal»
^score : Real
remote Avionics : Remote Avionics
autonomySystem : AutonomySystem
«equal»
«alternatives»
: AutonomySystem
«alternatives»
kind = EXCEL
source = Instance Table
commLink : CommLink
«equal»
«alternatives»
: CommLink
«alternatives»
kind = SUBTYPES
^OutOfSpec : Real
^N : Integer
^winner : String

Create the Execution

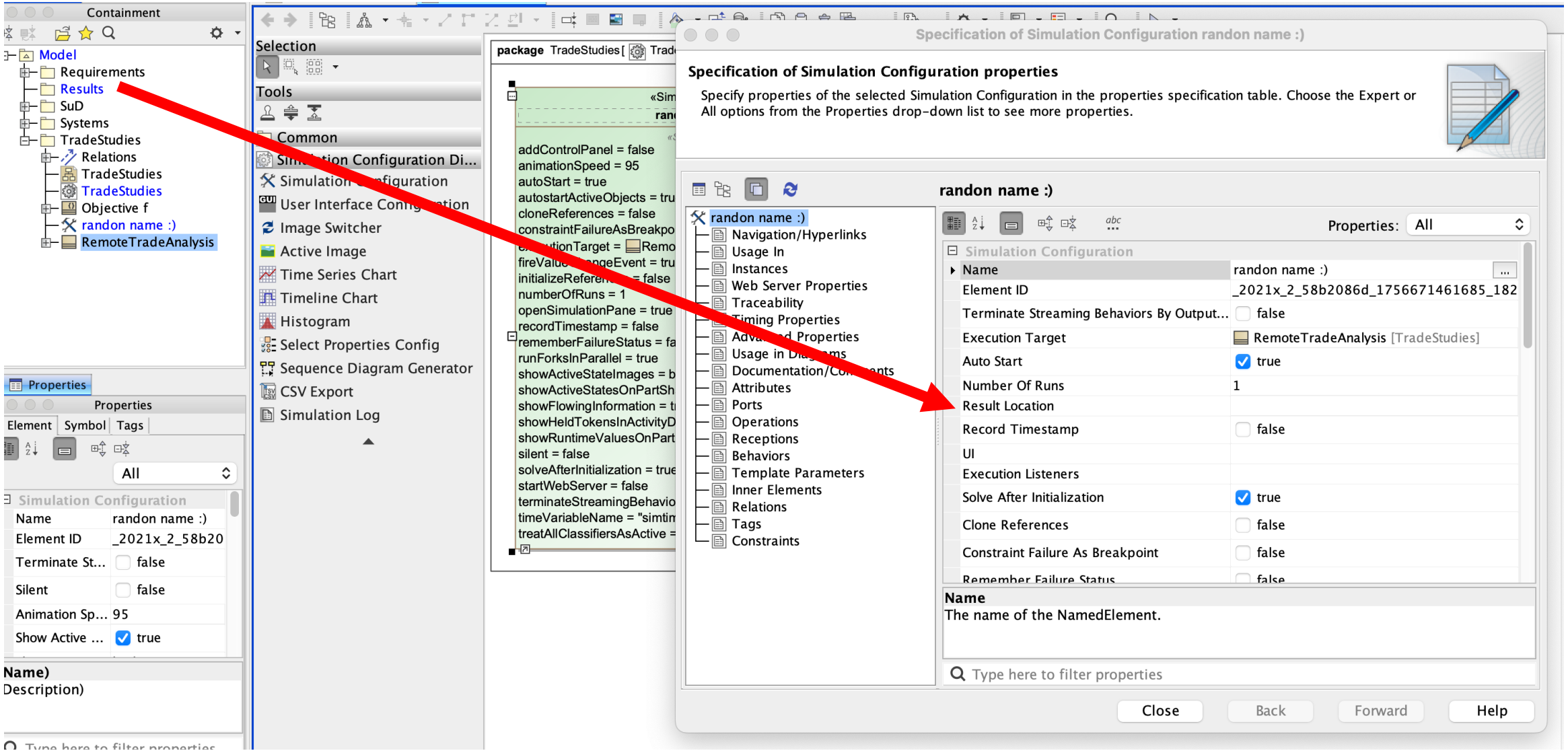




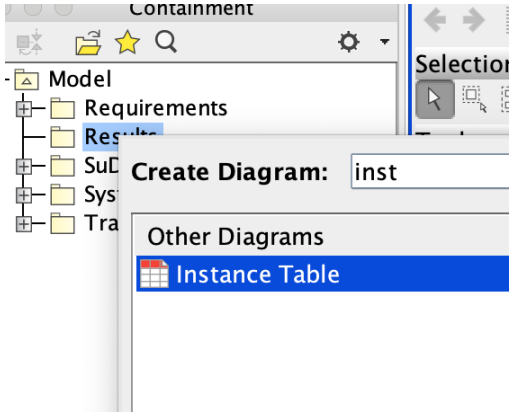
Link the trade studies



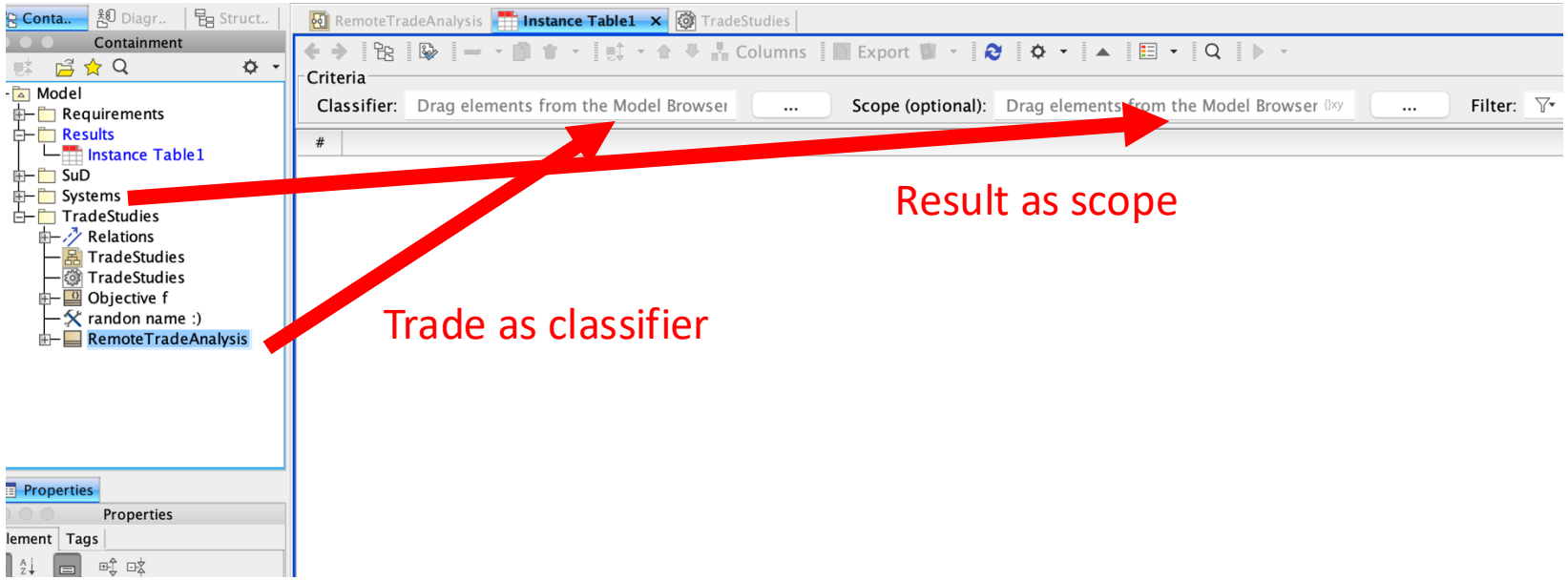
Setup result location



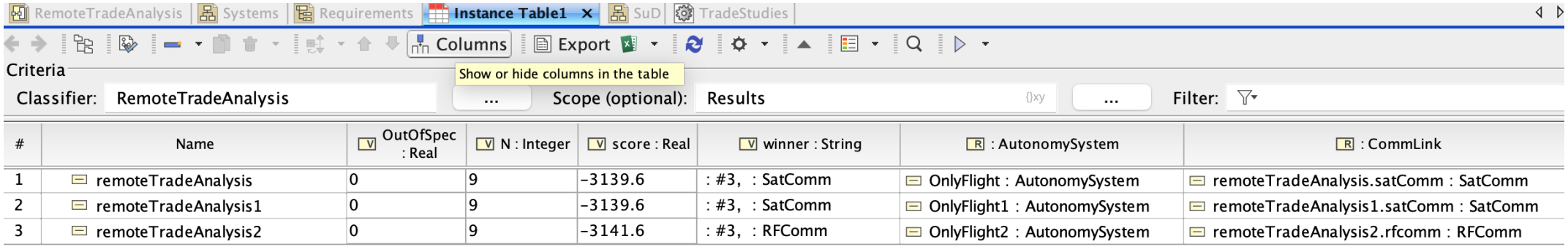
Create an instance table to save the results



define the columns to add the selected system alternatives



- Now:
  - Play w/ the requirements and values to see the magic happening
  - If zero answers.. Remove all requirements and start to play.



#	Name	OutOfSpec : Real	N : Integer	score : Real	winner : String	: AutonomySystem	: CommLink
1	remoteTradeAnalysis	0	9	-3139.6	: #3, : SatComm	OnlyFlight : AutonomySystem	remoteTradeAnalysis.satComm : SatComm
2	remoteTradeAnalysis1	0	9	-3139.6	: #3, : SatComm	OnlyFlight1 : AutonomySystem	remoteTradeAnalysis1.satComm : SatComm
3	remoteTradeAnalysis2	0	9	-3141.6	: #3, : RFComm	OnlyFlight2 : AutonomySystem	remoteTradeAnalysis2.rfcomm : RFComm

Monte Carlo

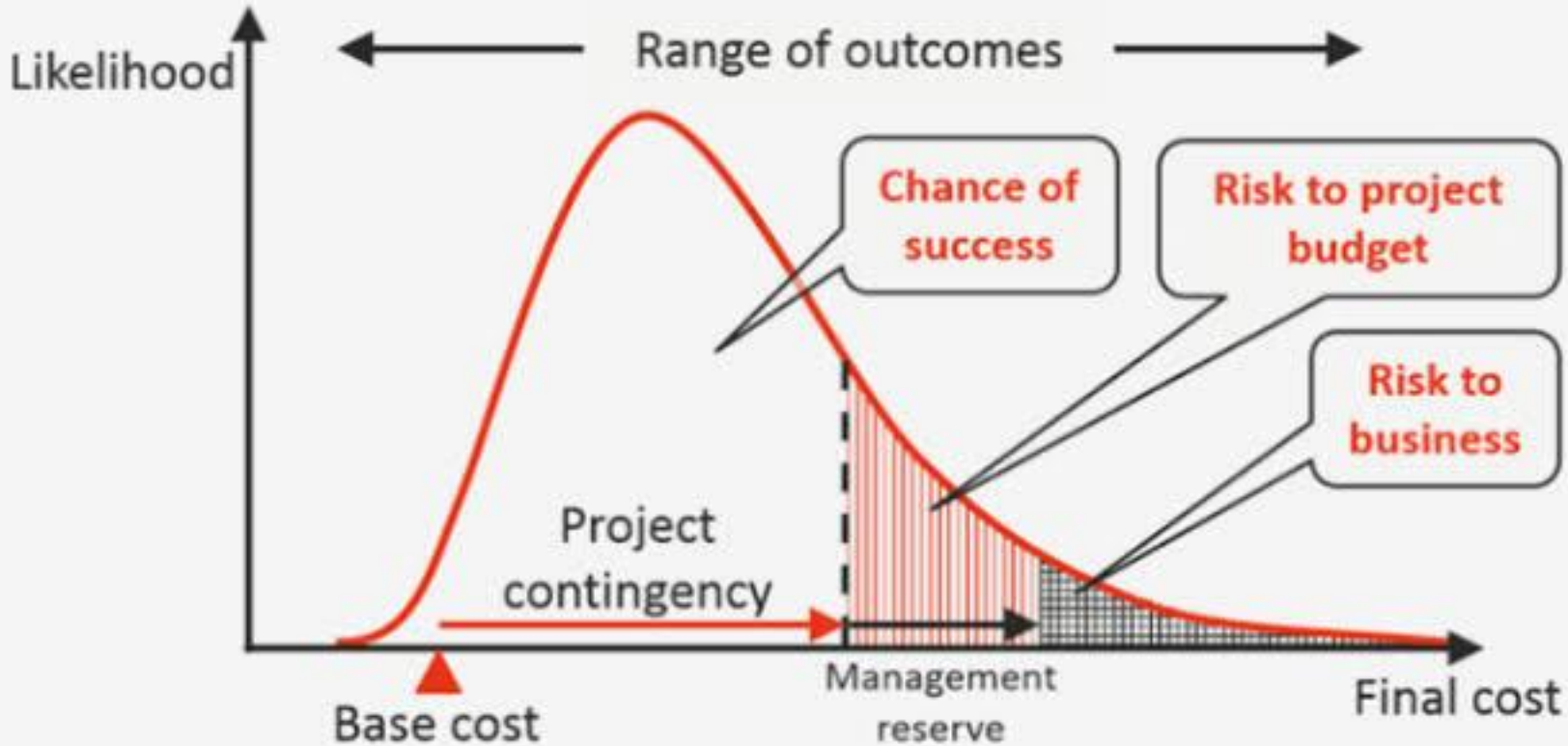
What is a Monte Carlo Analysis?

- Definition:
  - A **Monte Carlo Analysis** is a method that uses **random sampling and probability** to explore the behavior of a system.
  - Instead of calculating a single deterministic answer, we simulate the system **thousands of times** with different possible input values.

How it works

- Define uncertain variables (e.g., component reliability, cost, fuel burn).
- Assign probability distributions (normal, uniform, triangular, etc.).
- Run many simulations (e.g., 10,000 runs).
- Collect the results to see the range of possible outcomes.

- It shows not just “what could happen” but **how likely** different outcomes are.


Likelihood
Range of outcomes
Chance of success
Risk to project budget
Risk to business
Project contingency
Base cost
Management reserve
Final cost

Why is it Important in Systems Engineering?

- Systems Engineering deals with uncertainty
  - System performance **depends on many interacting variables**.
  - Requirements often include tolerances, probabilities, and margins.

- Monte Carlo helps to:
  - **Quantify risk** and uncertainty in performance, cost, and schedule.
  - **Identify probability of requirement satisfaction** (e.g., 95% chance aircraft range ≥ 3000 km).
  - **Support trade studies and design decisions** with evidence.
  - Communicate **confidence levels** to stakeholders (not just “it works” vs “it fails”).

Example

- Instead of saying: “The aircraft has a range of 3000 km.”
- We say: “There is a 90% probability that the aircraft range will exceed 3000 km, given uncertainties in fuel efficiency and payload weight.”

Tutorial step by step

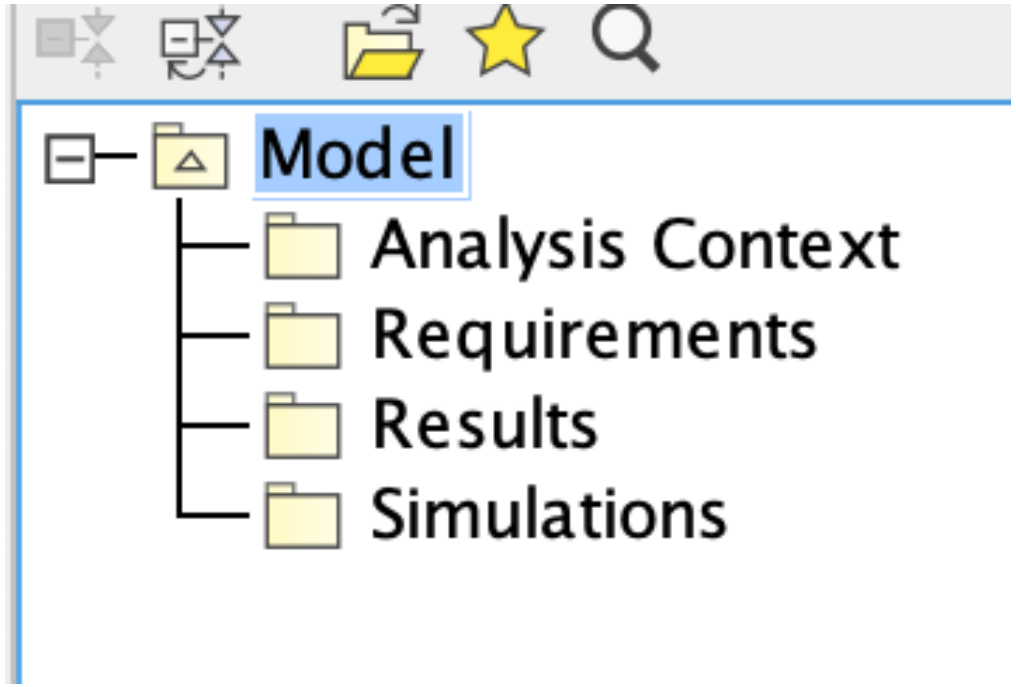
So ok…

- Different of the Trade Study, where you select the alternatives and there is no value iteration.
- The Monte Carlo will change the values of some parameters into a distribution (usually normal).

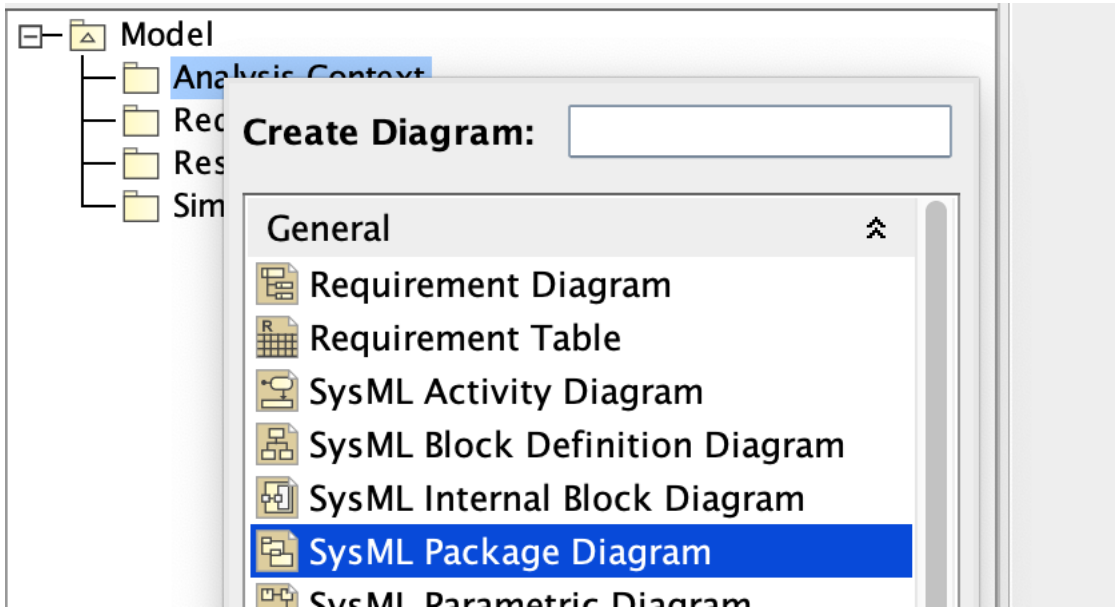
Steps:

1. Create the study scope (pkg/par)
2. Add the normal variables
3. Create the Simulation
    1. Setup the Simulation to show histograms and export the data to csv
4. Run

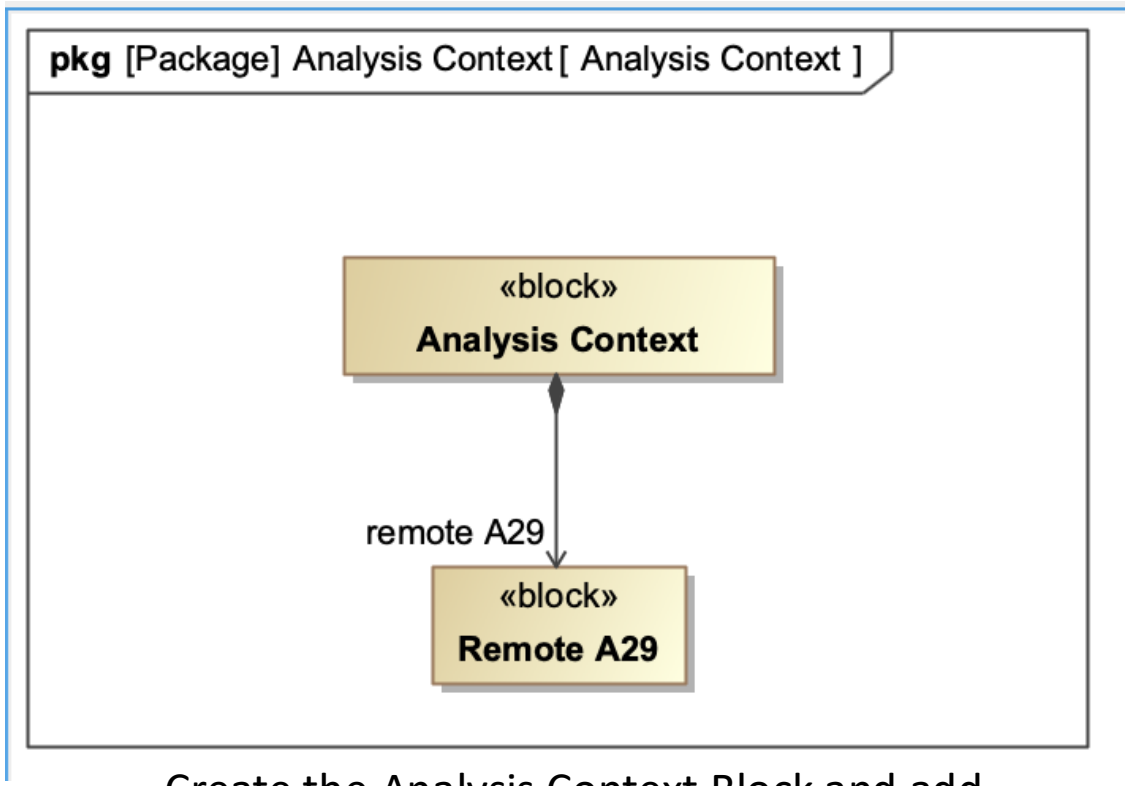
1 – Start creating the Package Structure



1 – Create a single element that we will apply Monte Carlo

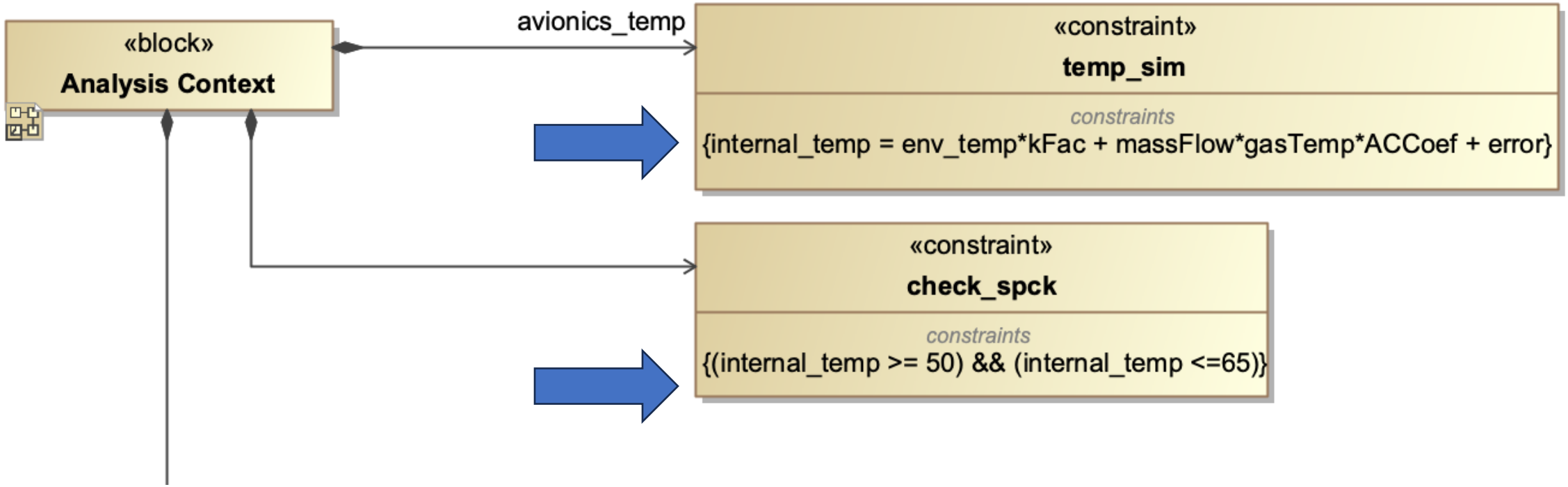


Create the pkg of the analysis context (to change a little)



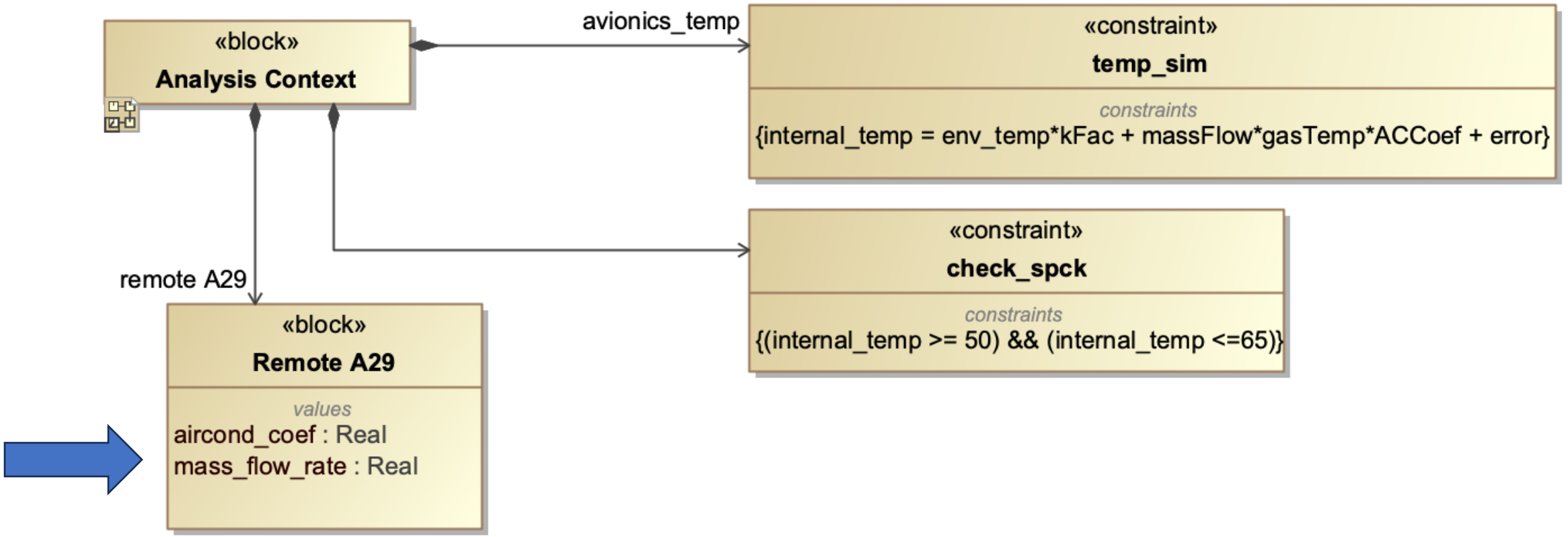
Create the Analysis Context Block and add the Remote A29

1. Define two constraints to create a simple “math” (remember the parametric diagram!?)

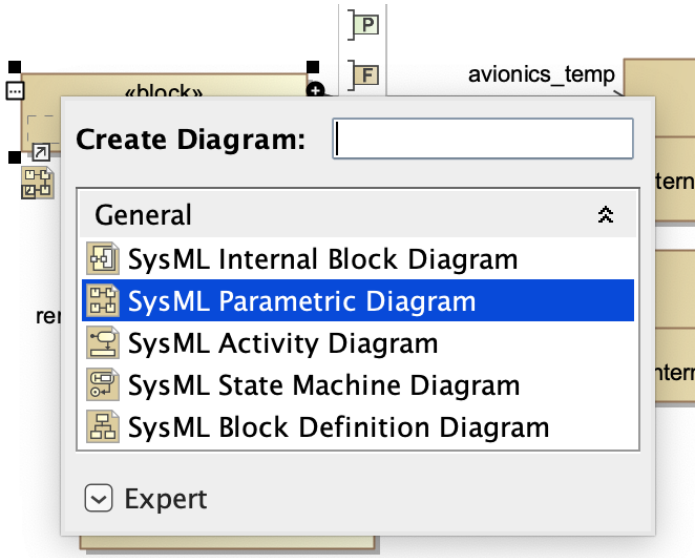


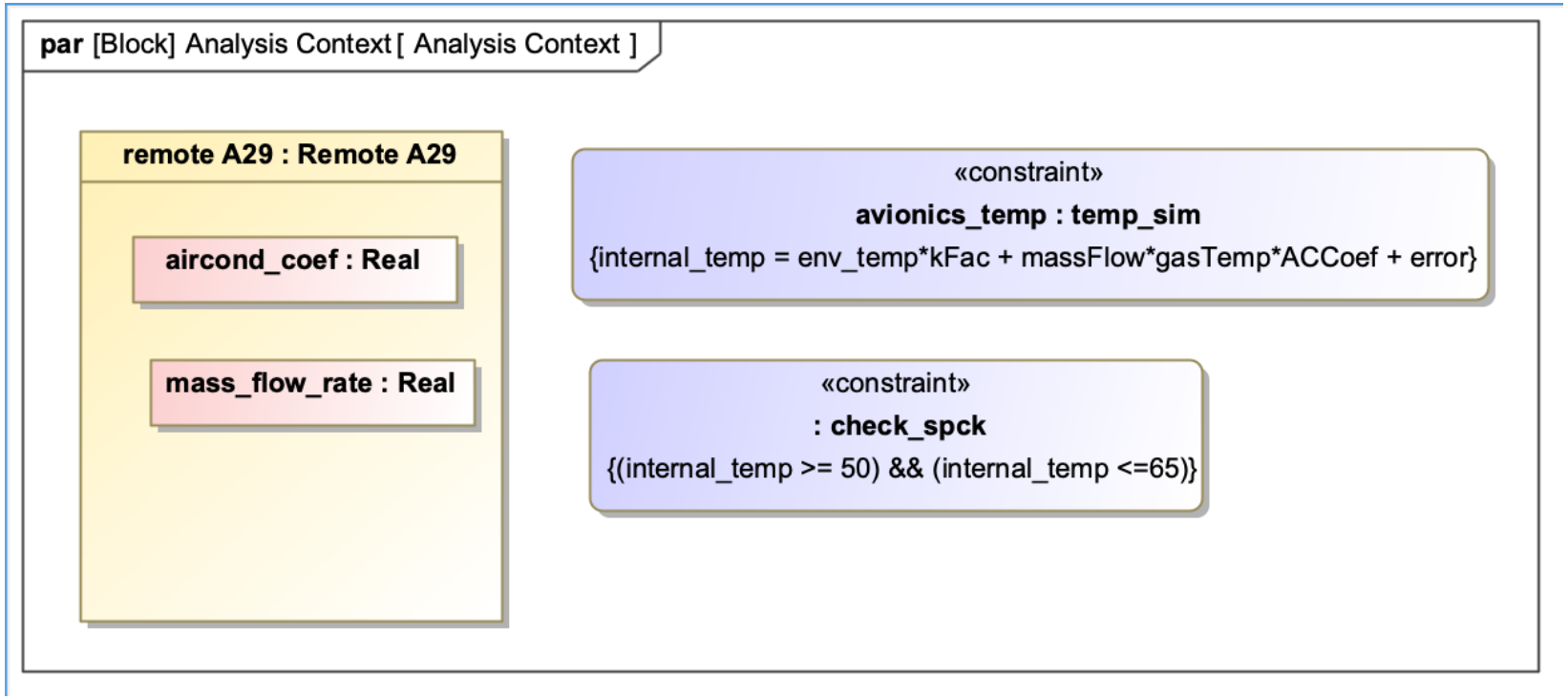
**random equation

1. Create values to the SuD

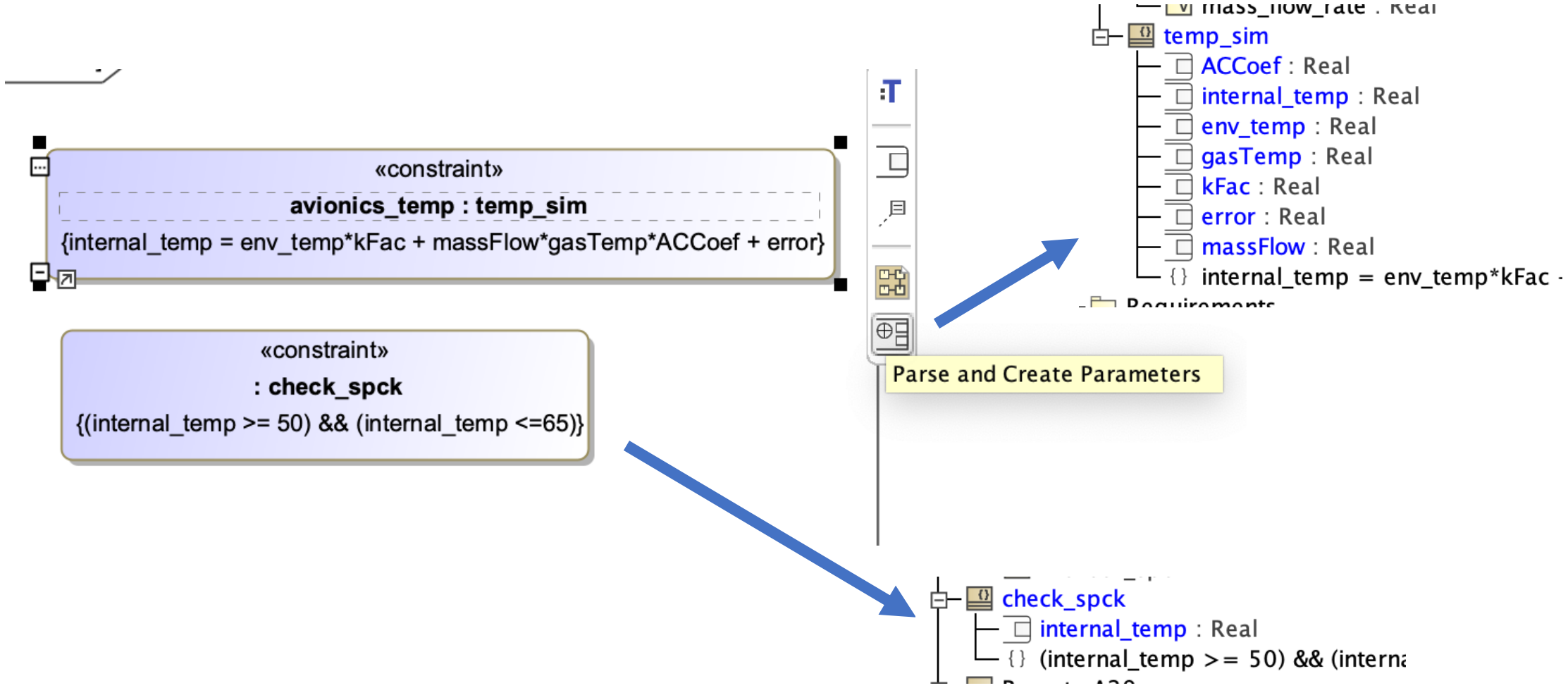


1. Create the Parametric Diagram of the Analysis Context

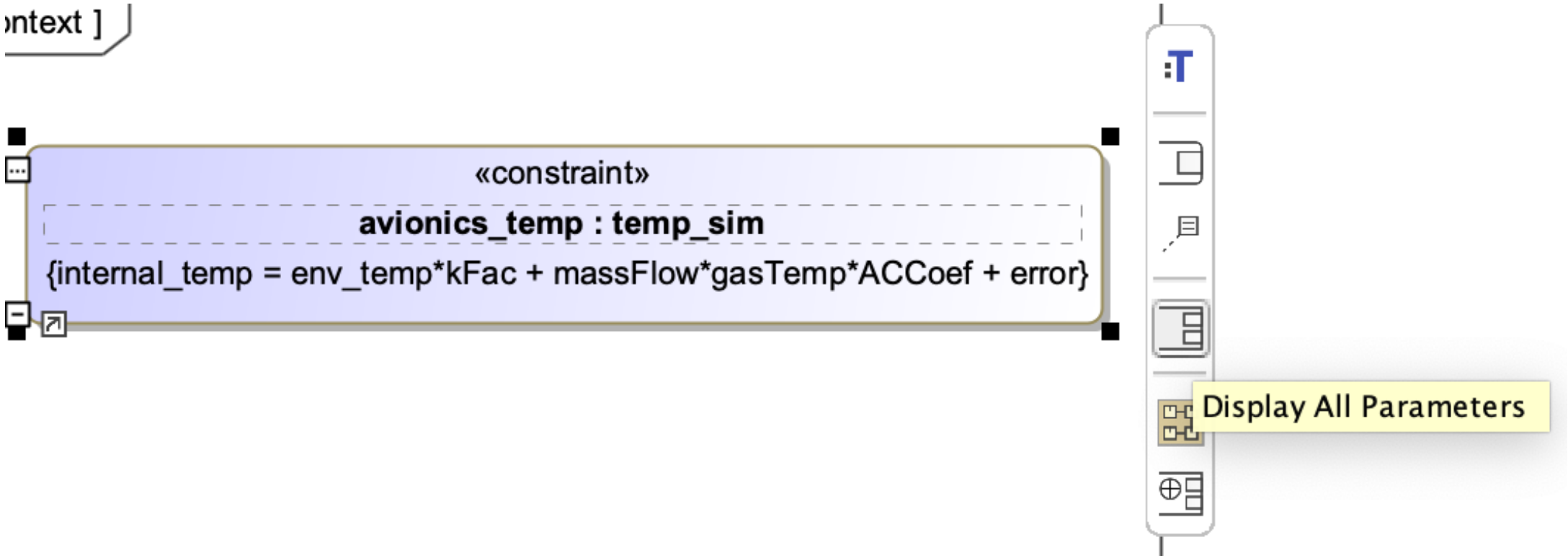


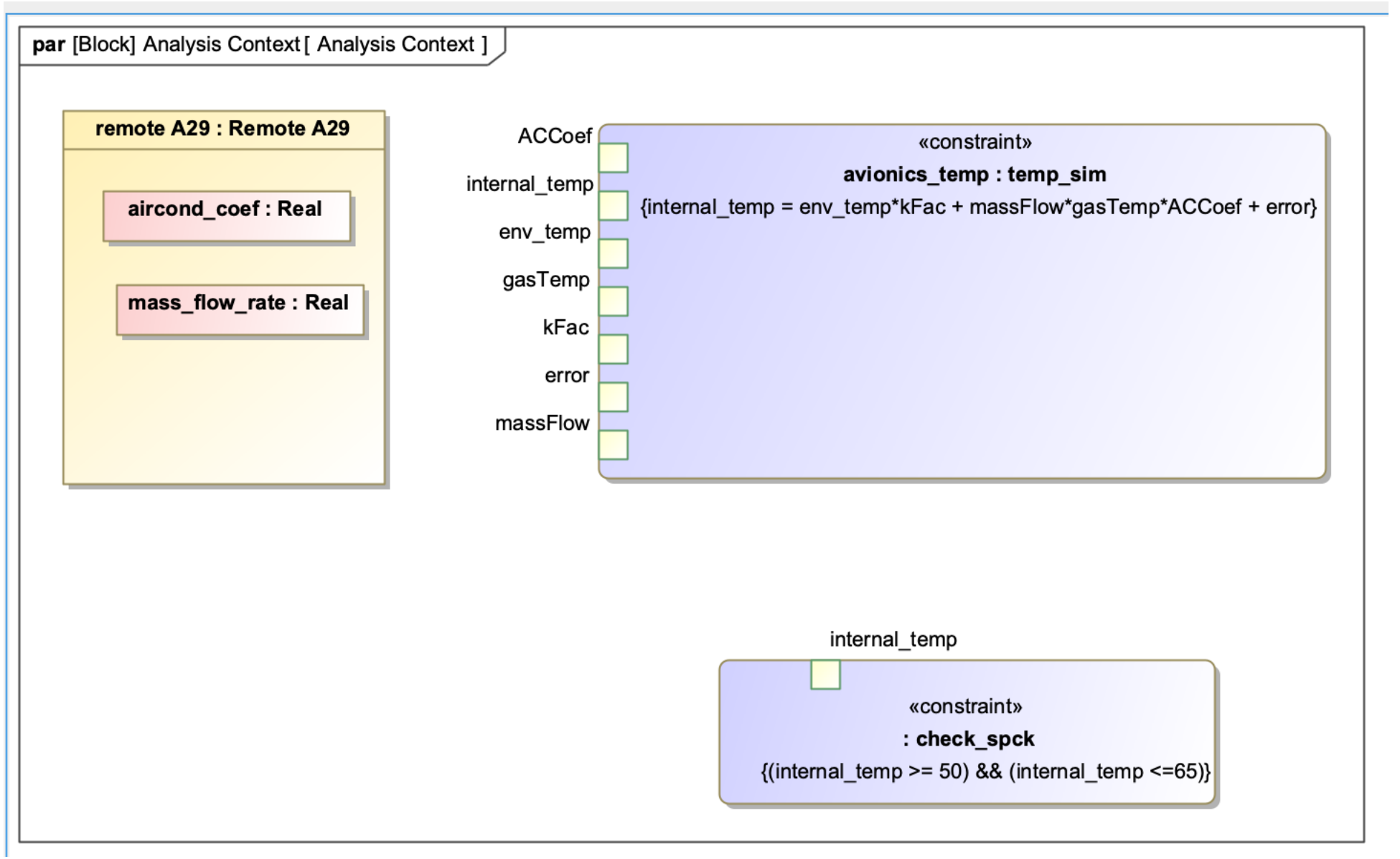


1. Parse the constraints

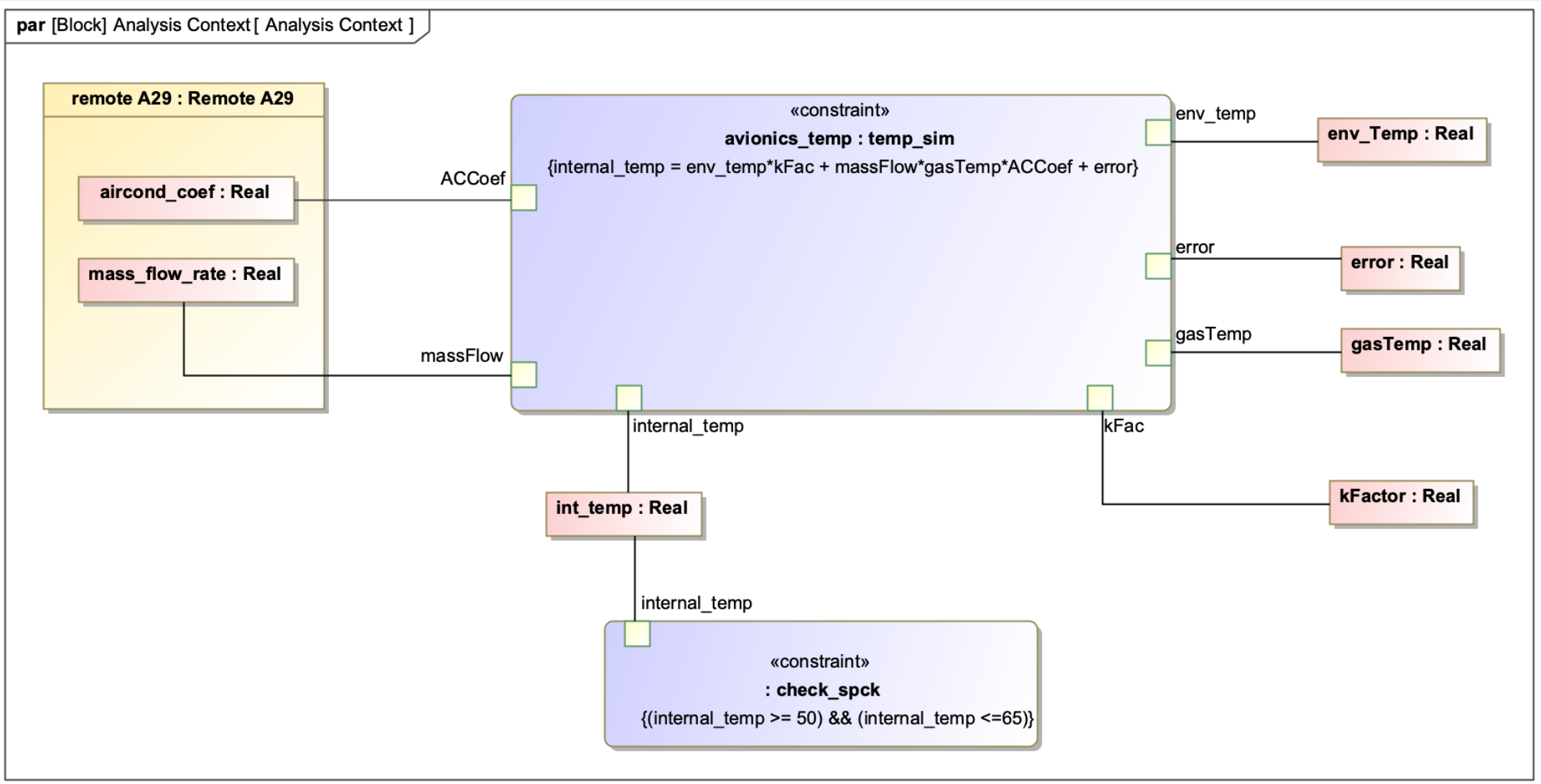


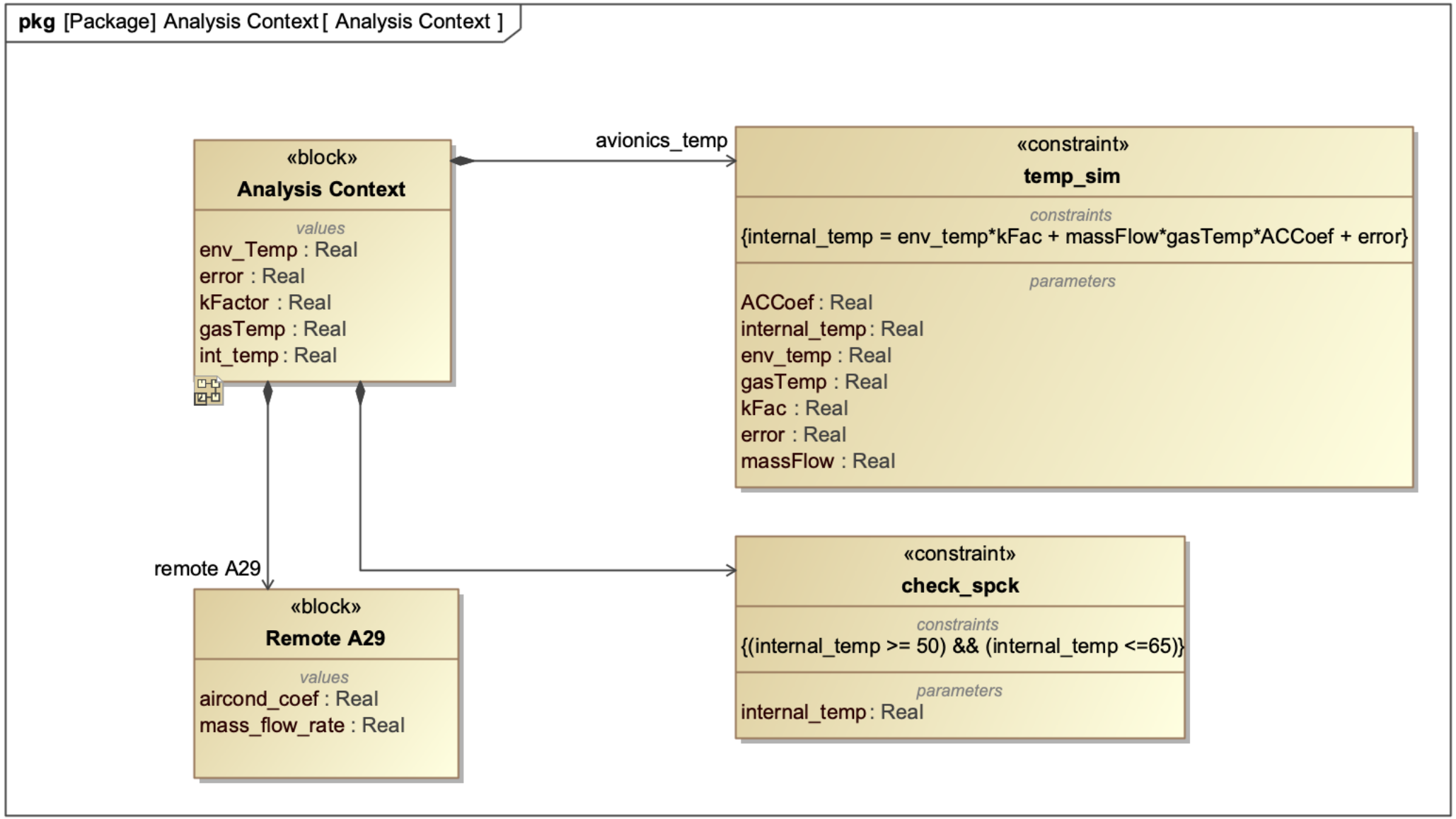
1. Expose the ports



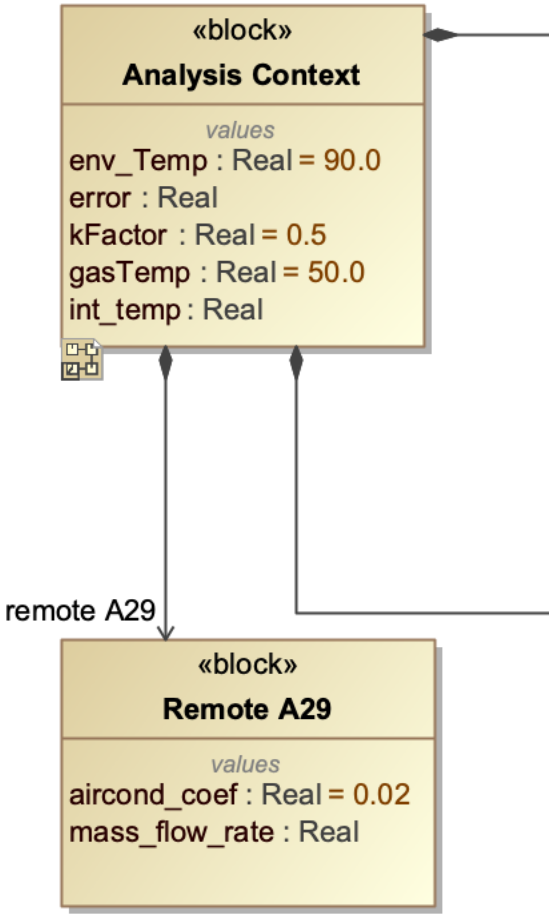

par [Block] Analysis Context [Analysis Context]
remote A29 : Remote A29
aircond_coef : Real
mass_flow_rate : Real
ACCoef
internal_temp
env_temp
gasTemp
kFac
error
massFlow
«constraint»
avionics_temp : temp_sim
{internal_temp = env_temp*kFac + massFlow*gasTemp*ACCoef + error}
internal_temp
«constraint»
: check_spck
{(internal_temp >= 50) && (internal_temp <=65)}

Create analysis variables and link to the aircraft:



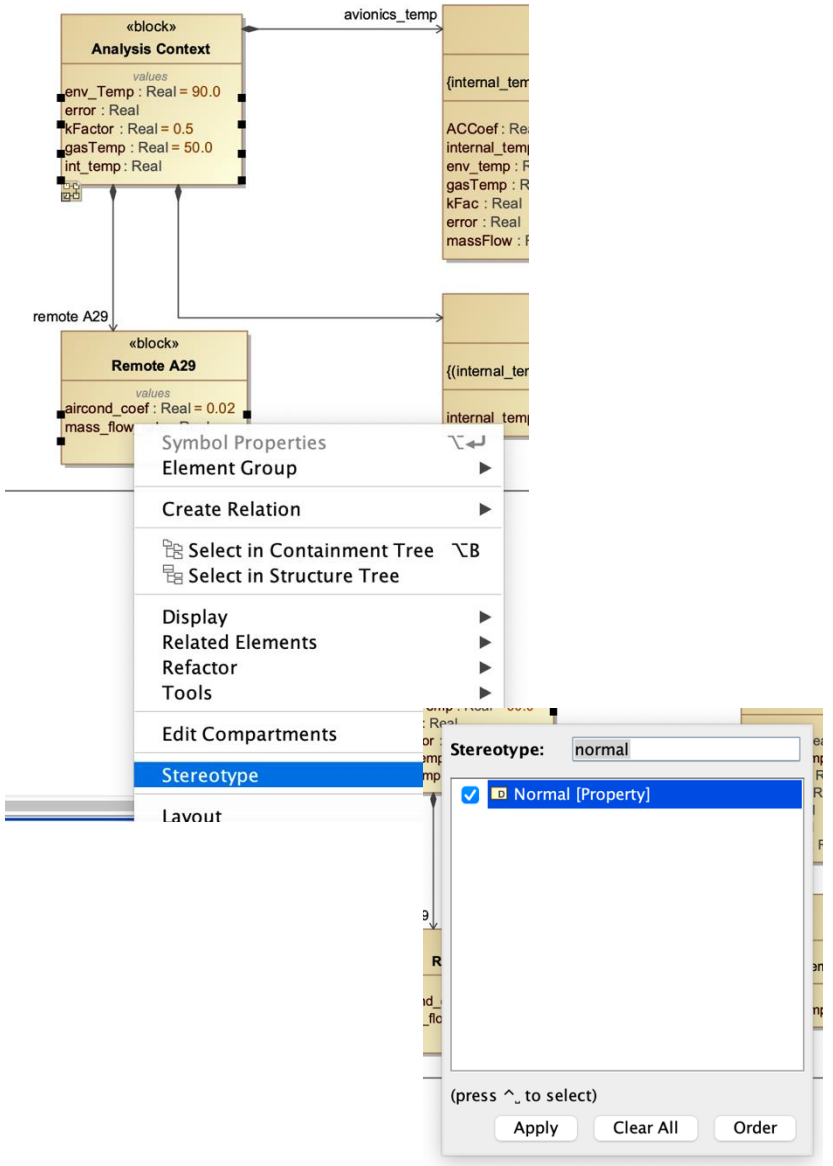

pkg [Package] Analysis Context [Analysis Context]
«block»
Analysis Context
values
env_Temp : Real
error : Real
kFactor : Real
gasTemp : Real
int_temp : Real
avionics_temp
«constraint»
temp_sim
constraints
{internal_temp = env_temp*kFac + massFlow*gasTemp*ACCoef + error}
parameters
ACCoef : Real
internal_temp : Real
env_temp : Real
gasTemp : Real
kFac : Real
error : Real
massFlow : Real
remote A29
«block»
Remote A29
values
aircond_coef : Real
mass_flow_rate : Real
«constraint»
check_spck
constraints
{(internal_temp >= 50) && (internal_temp <=65)}
parameters
internal_temp : Real

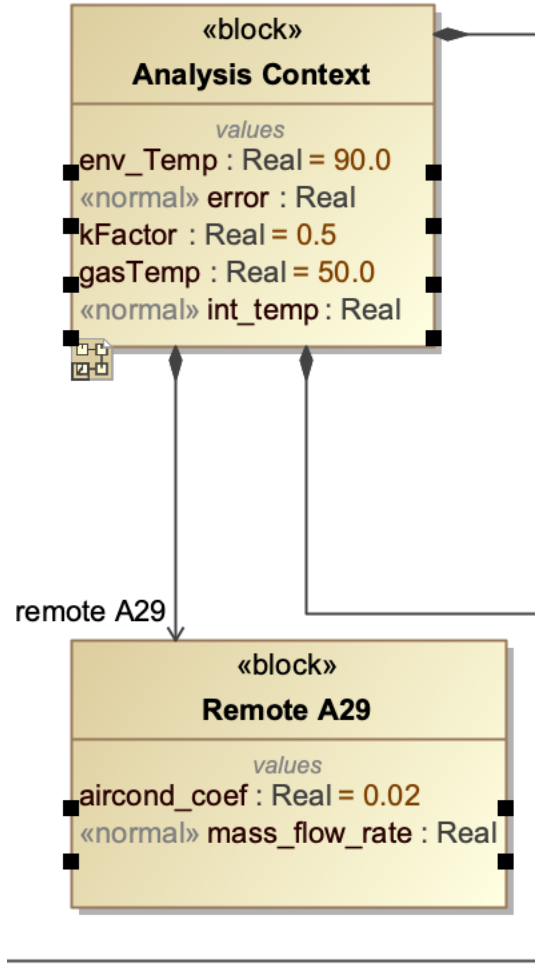
1. Initialize some values



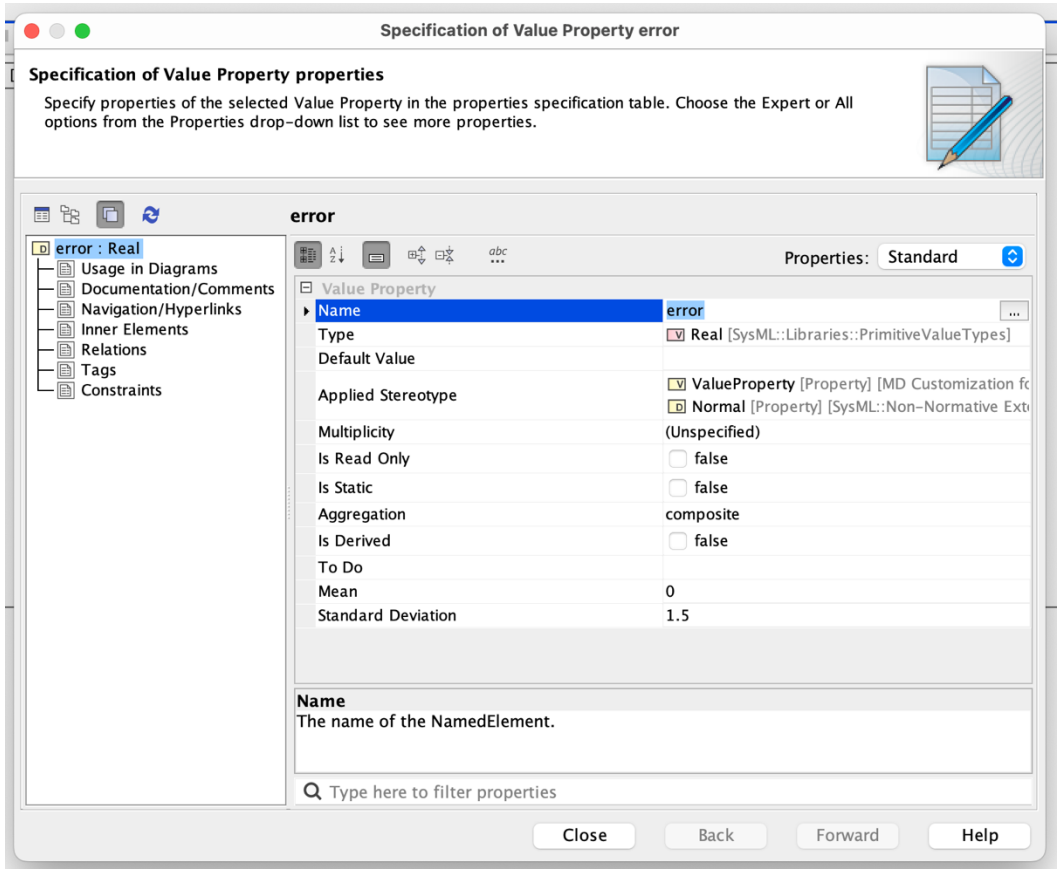
2. Add the normal distribution types

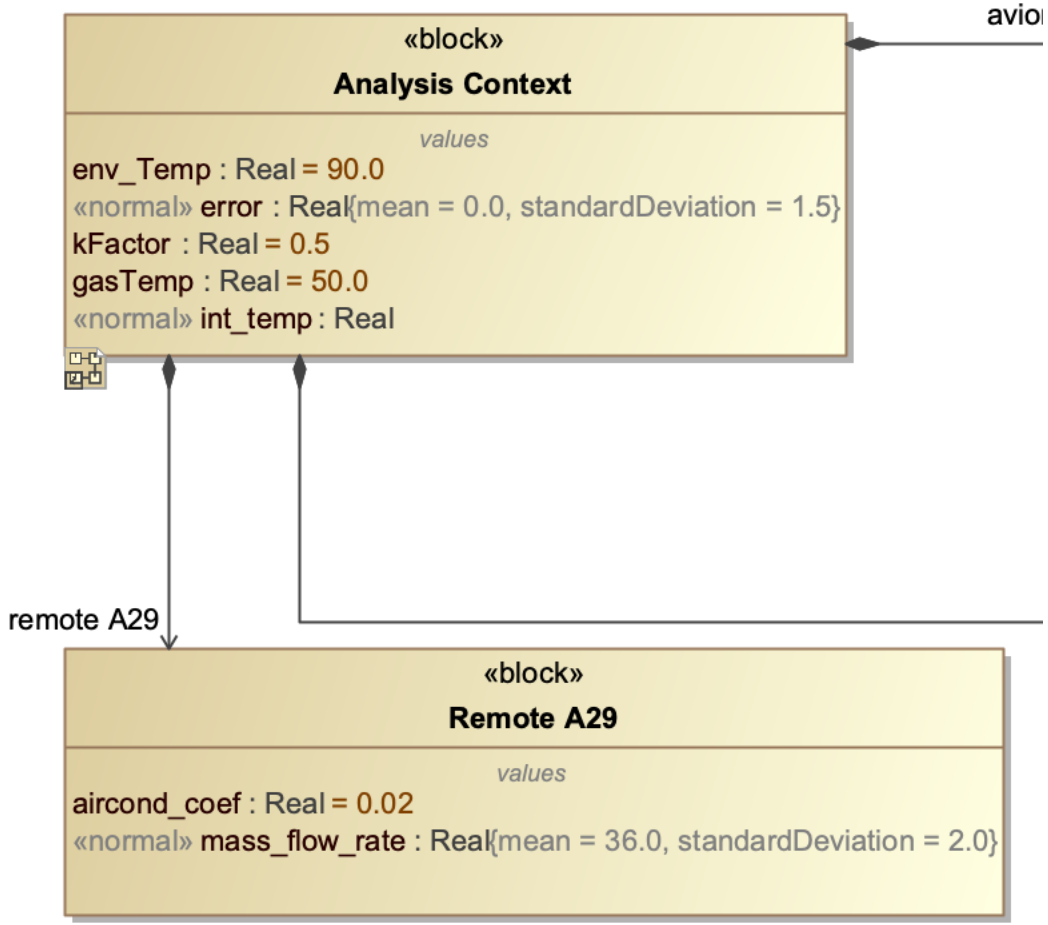
- Select the values to vary:
  - Error
  - int_temp
  - mass_flow
- Add Normal stereotype



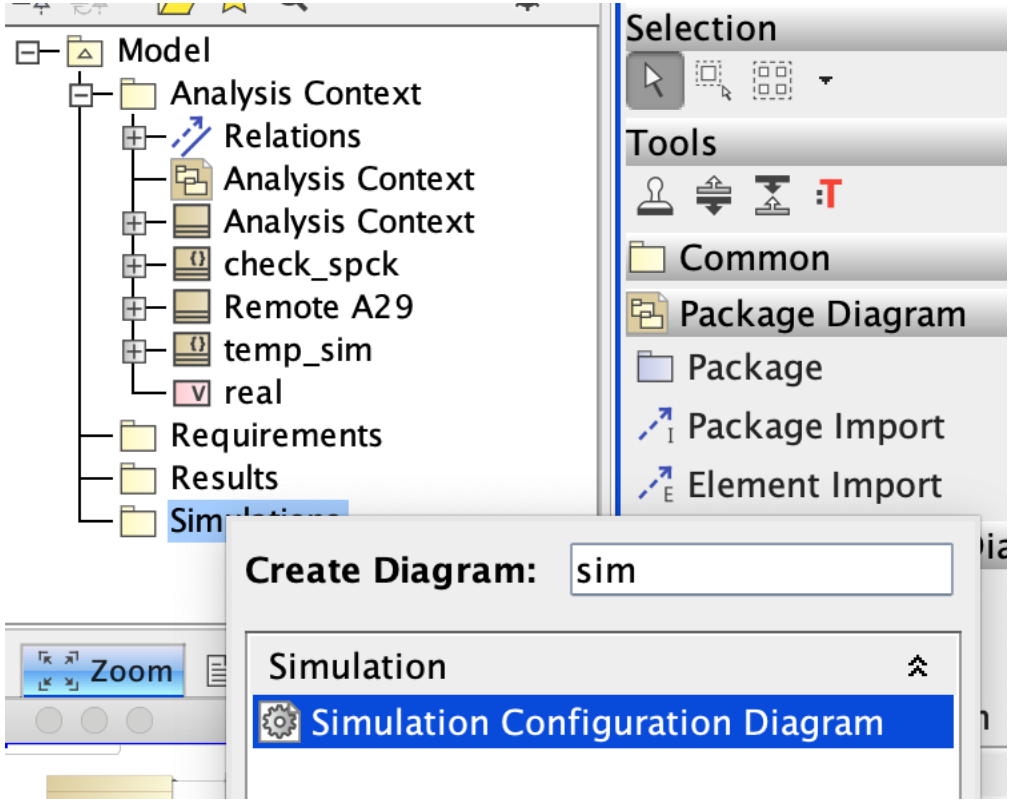


2. Fill mean and std deviation in error / mass_flow





3. Create Simulation



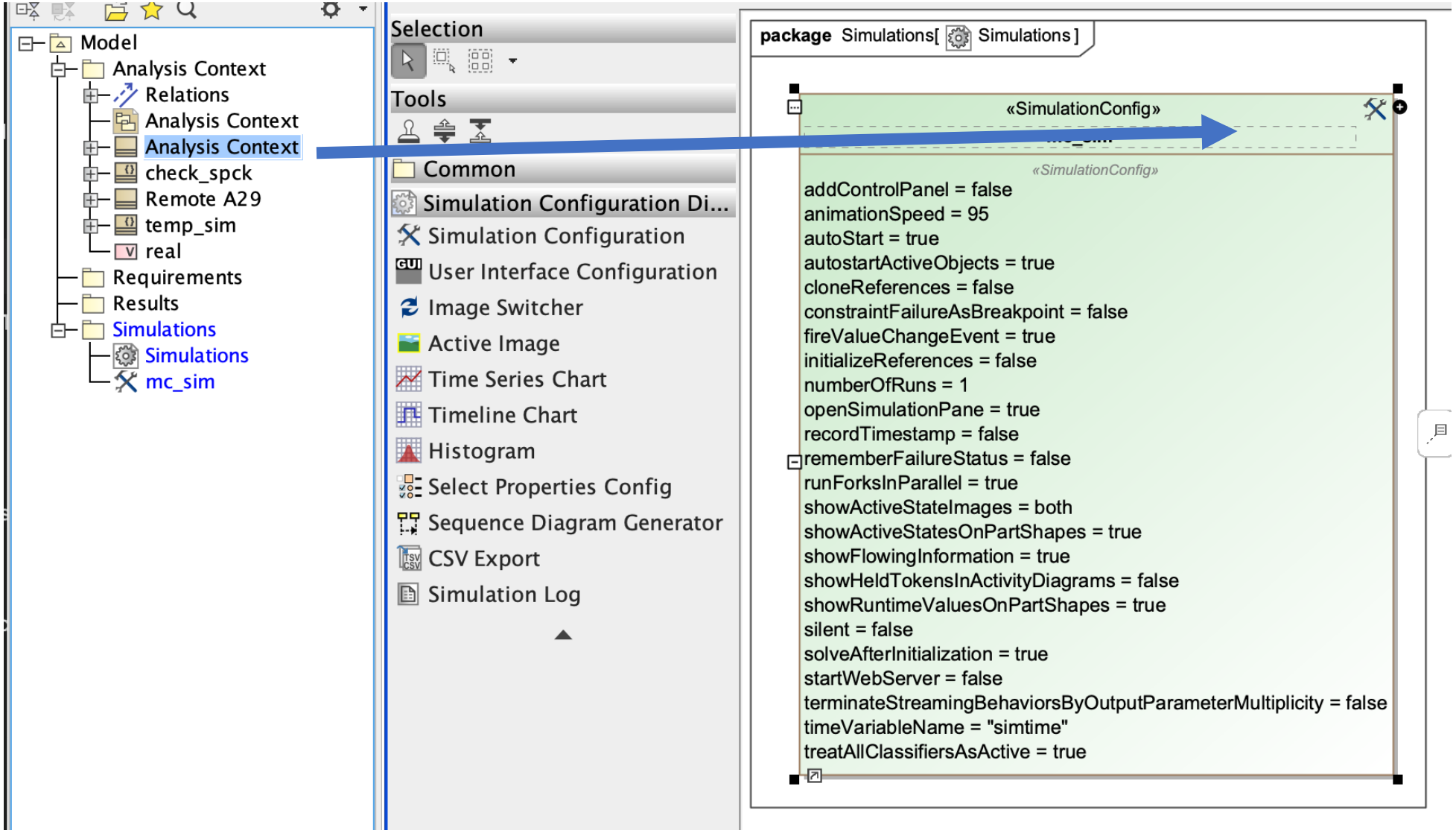
package Simulations[Simulations]

«SimulationConfig»

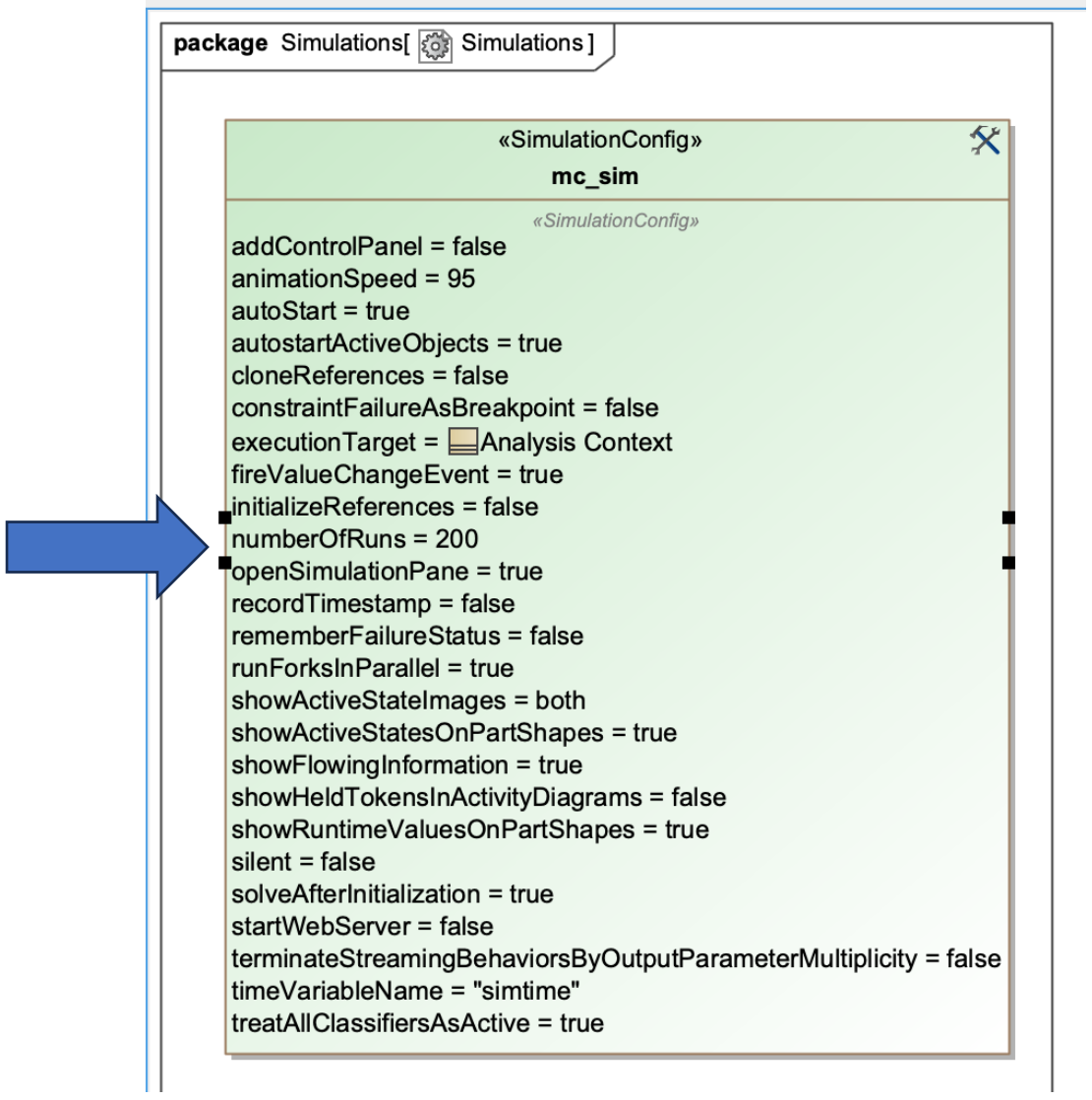
«SimulationConfig»

addControlPanel = false
animationSpeed = 95
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1
openSimulationPane = true
recordTimestamp = false
rememberFailureStatus = false
runForksInParallel = true
showActiveStateImages = both
showActiveStatesOnPartShapes = true
showFlowingInformation = true
showHeldTokensInActivityDiagrams = false
showRuntimeValuesOnPartShapes = true
silent = false
solveAfterInitialization = true
startWebServer = false
terminateStreamingBehaviorsByOutputParameterMultiplicity = false
timeVariableName = "simtime"
treatAllClassifiersAsActive = true

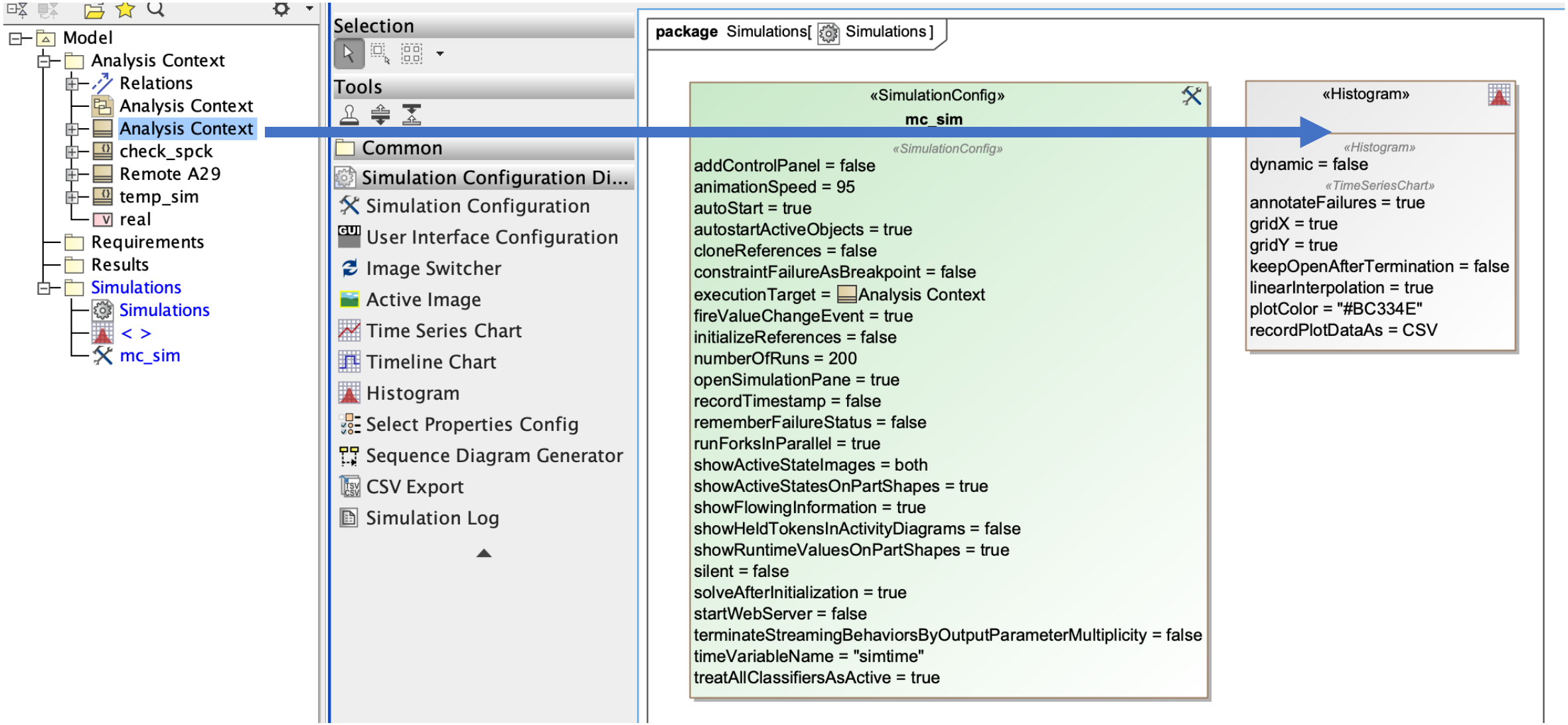
3. Setup Simulation – bring the execution target



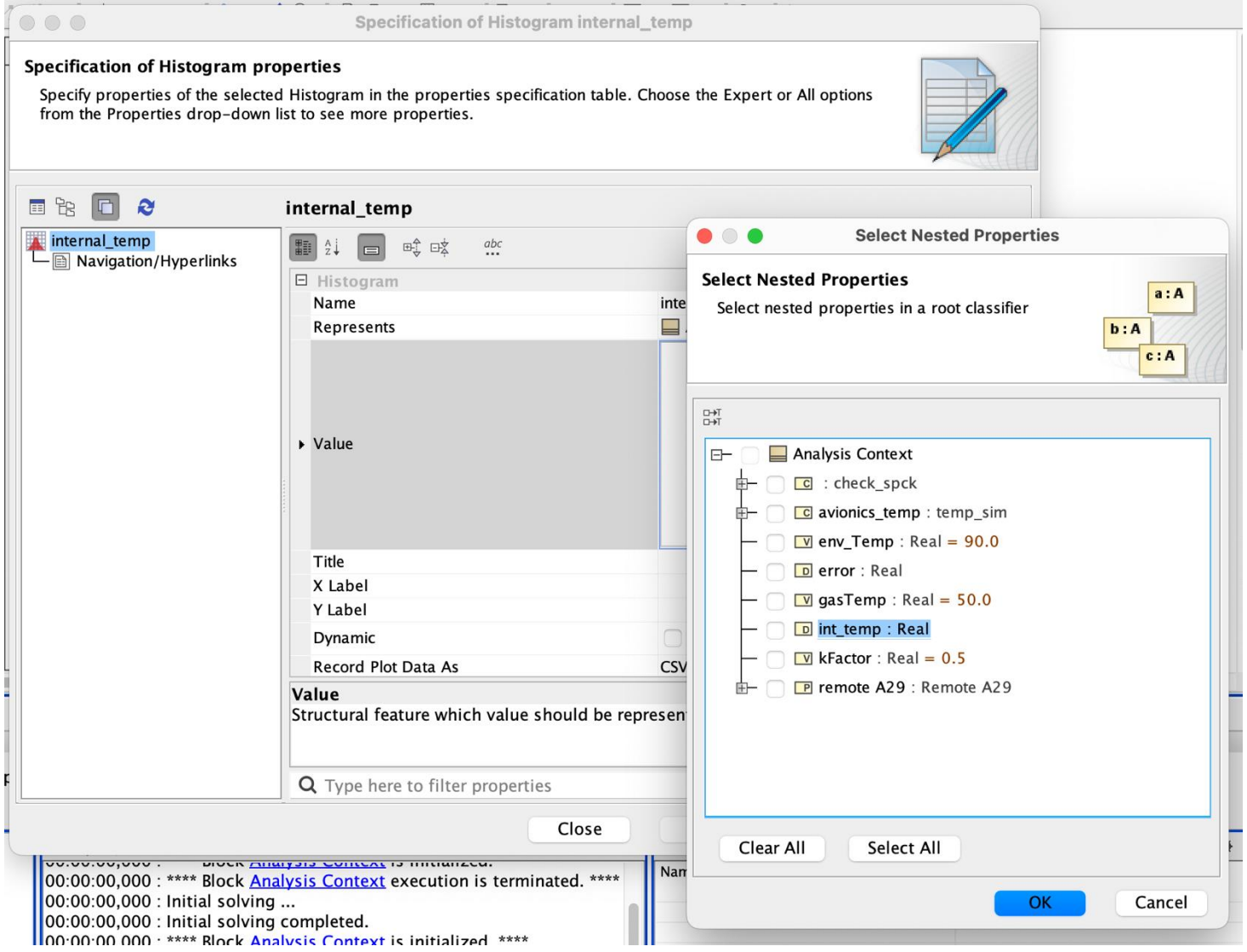
3. Set the number of runs



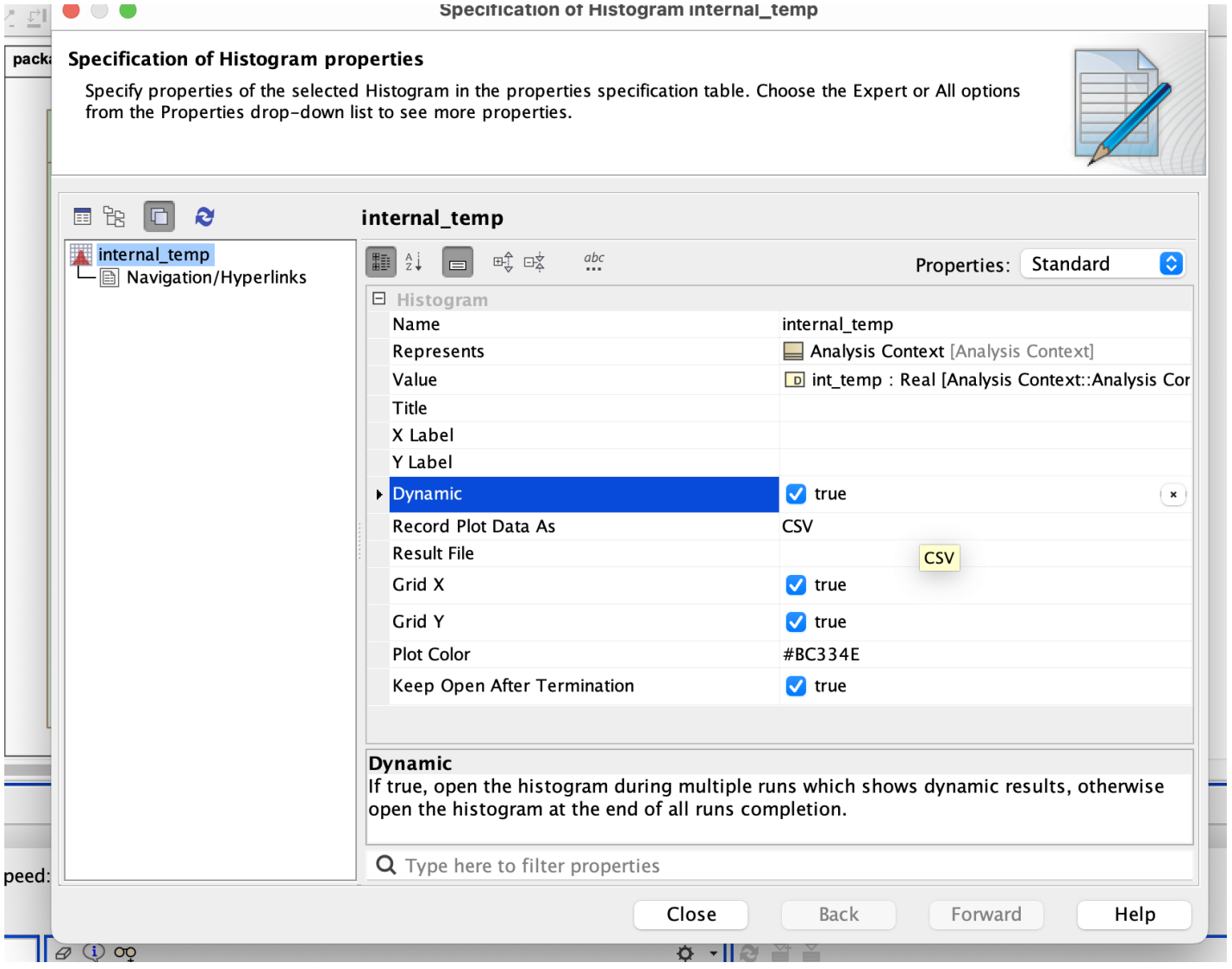
3. Set some UI w/ Histograms (bring the context)



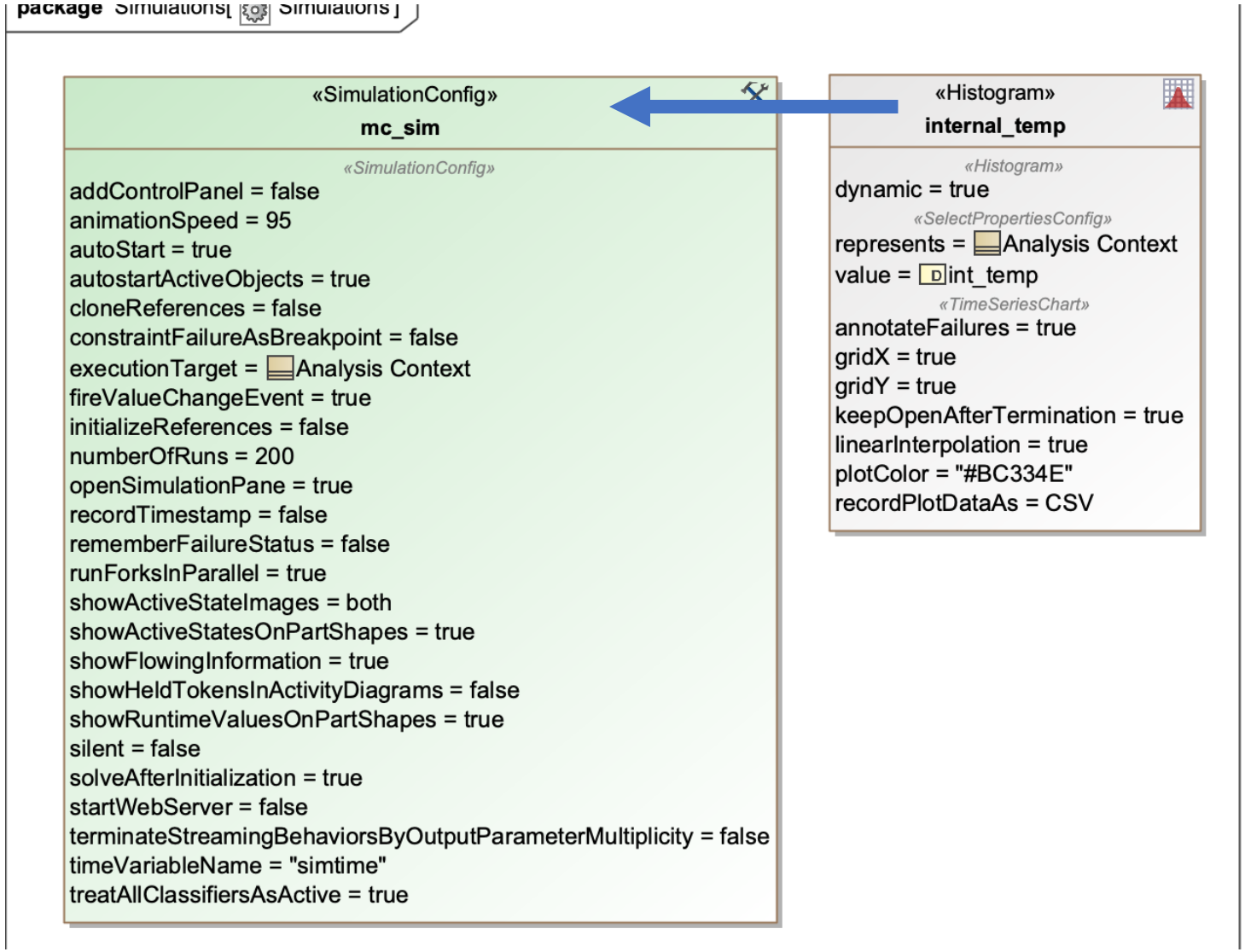
3. Configure Histogram value = int_temp



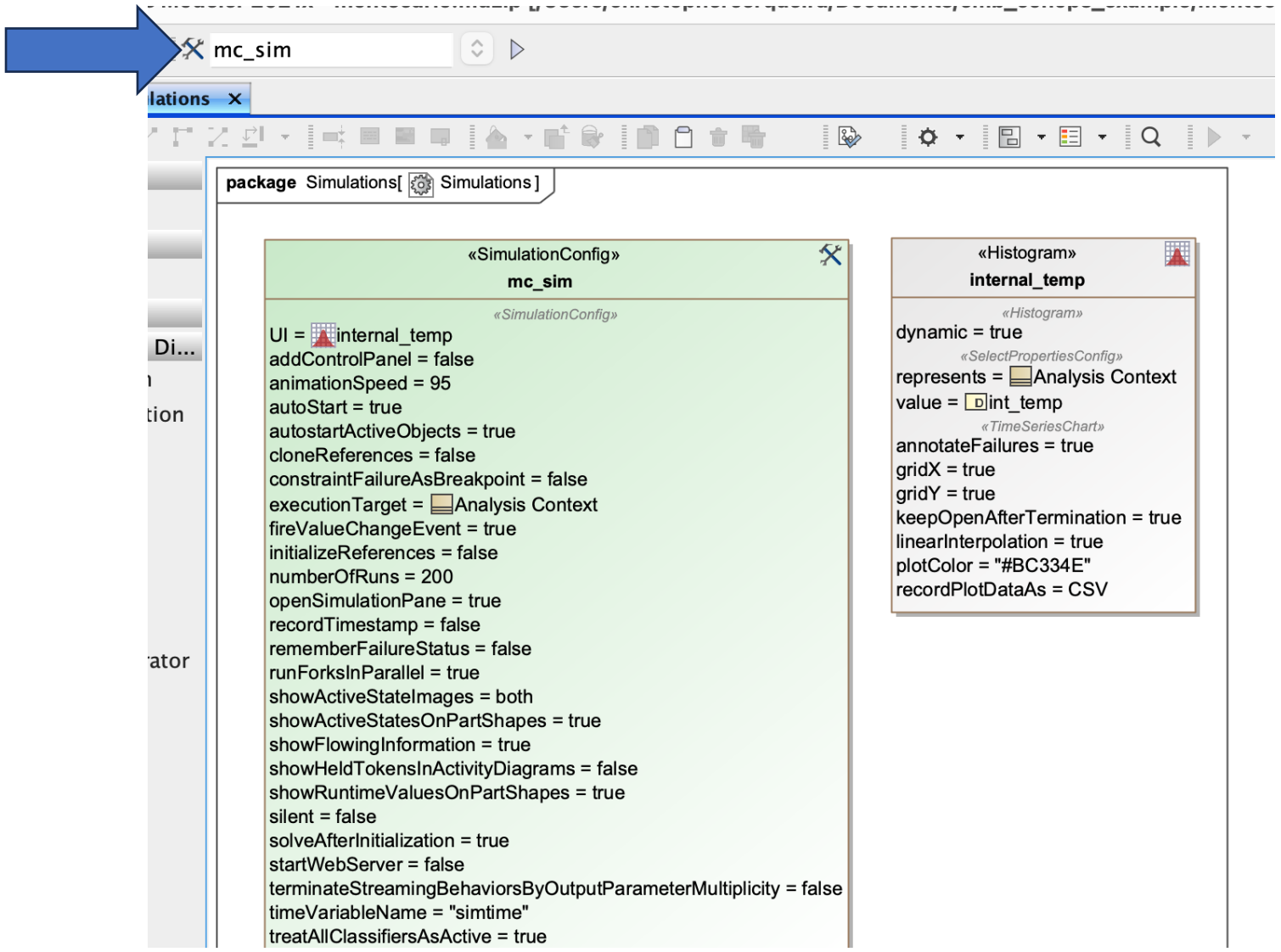
3. Set “keep open” and “dynamic”



3. Add histogram as UI

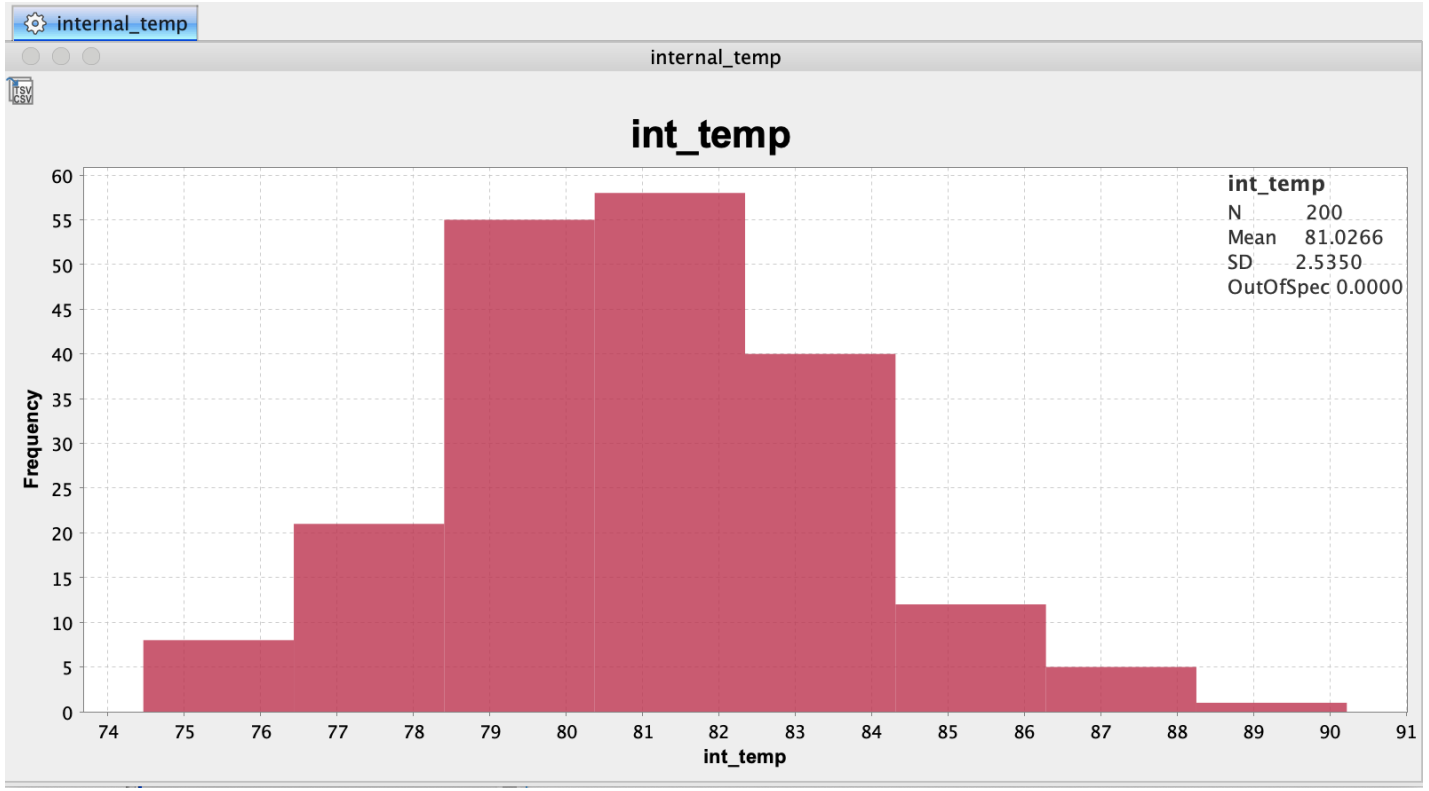


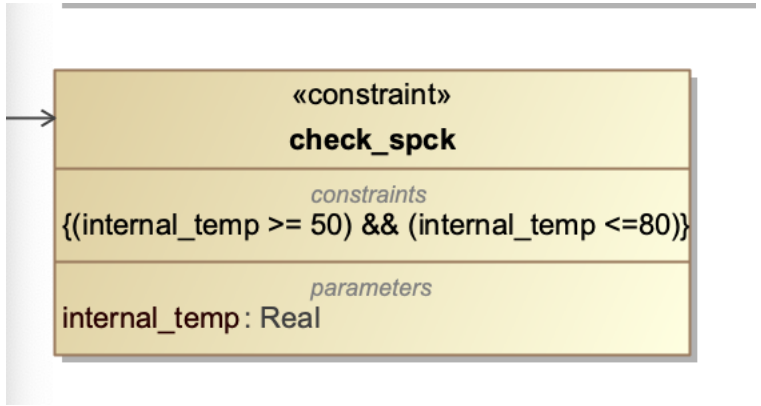
4. Run


mc_sim
package Simulations[Simulations]
«SimulationConfig»
mc_sim
«SimulationConfig»
UI = internal_temp
addControlPanel = false
animationSpeed = 95
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionTarget = Analysis Context
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 200
openSimulationPane = true
recordTimestamp = false
rememberFailureStatus = false
runForksInParallel = true
showActiveStateImages = both
showActiveStatesOnPartShapes = true
showFlowingInformation = true
showHeldTokensInActivityDiagrams = false
showRuntimeValuesOnPartShapes = true
silent = false
solveAfterInitialization = true
startWebServer = false
terminateStreamingBehaviorsByOutputParameterMultiplicity = false
timeVariableName = "simtime"
treatAllClassifiersAsActive = true
«Histogram»
internal_temp
«Histogram»
dynamic = true
«SelectPropertiesConfig»
represents = Analysis Context
value = int_temp
«TimeSeriesChart»
annotateFailures = true
gridX = true
gridY = true
keepOpenAfterTermination = true
linearInterpolation = true
plotColor = "#BC334E"
recordPlotDataAs = CSV

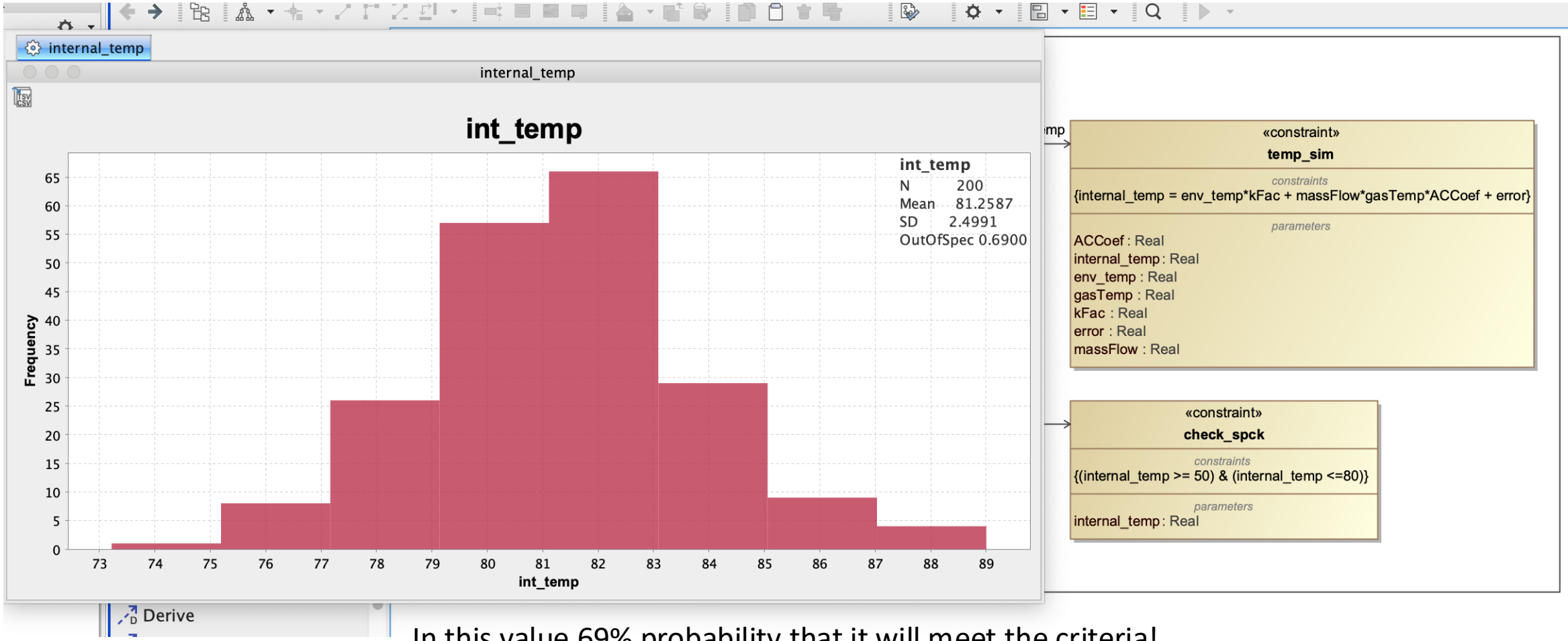
4. First Analysis

- The value of outspec accept all..
- Why?
- Because the constraint is wrong: it is just um &





4. With the correct constraint formula and “opened” accepted value ☺



In this value 69% probability that it will meet the criteria!

Considerações finais

Considerações

- A modelagem paramétrica tem vantagens.
- Nada mais é do que conseguir indicar “parâmetros” “restrições” à outros blocos.
- O diagrama paramétrico é um diagrama de blocos dos parâmetros (Constraints Blocks).

- Mais pra frente veremos que podemos usar essa técnica para fazer análises, trades, co-engineering..