



Graduate Program in Science and Space Technologies (PG-CTE)

*SPACE SYSTEMS, TESTING AND LAUNCHING (CTE-E)*

# PARAMETRICO E TRADES

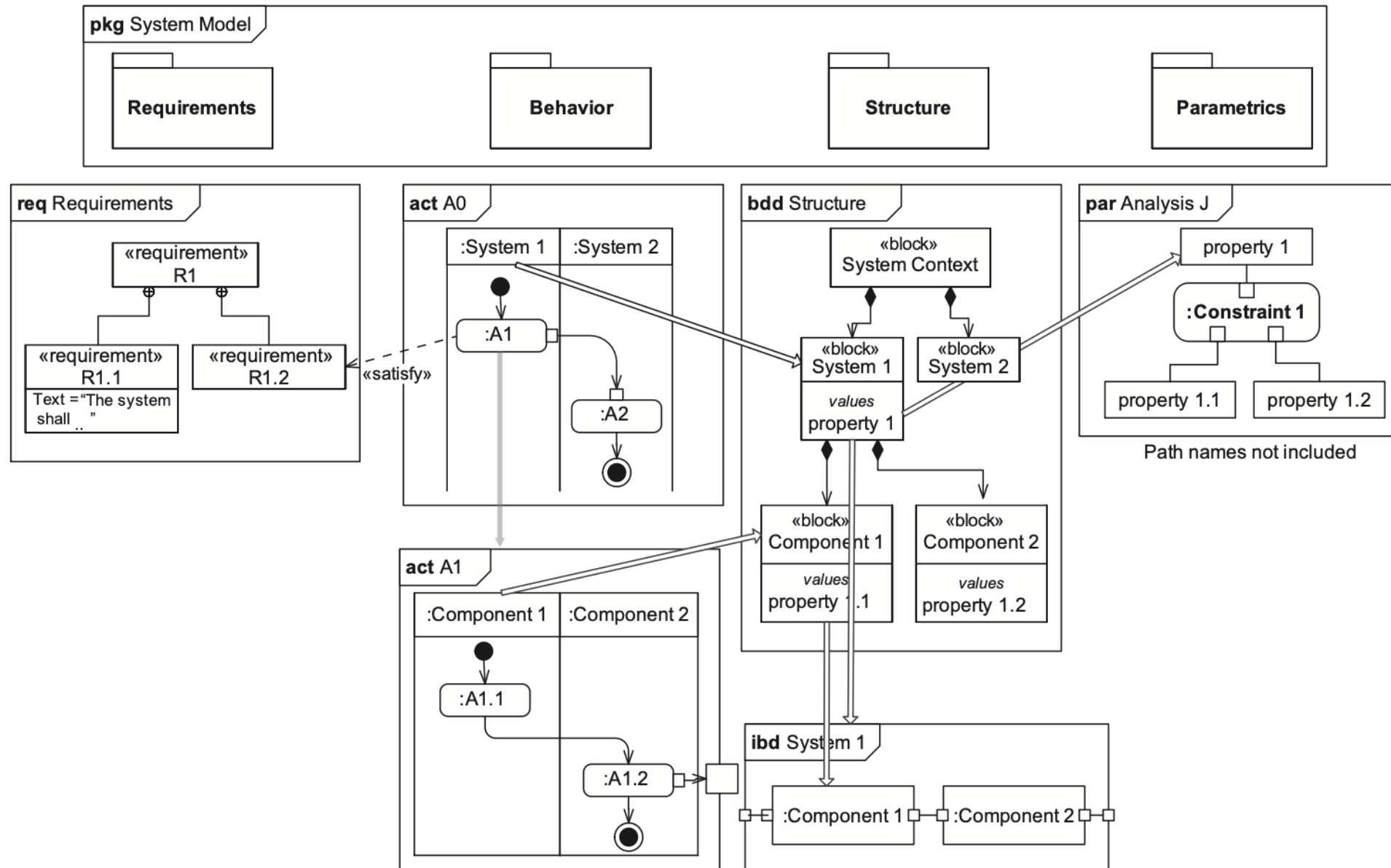
2025 – Preparado por Prof. Dr. Christopher Shneider Cerqueira



WEEK	CLASS ACTIVITY	REF	INDIVIDUAL	W	GROUP	W
1	Course Structure and Initial Definitions					
28/Jul	Systems Engineering Review	[1][2][3][4]	IA-01 - Reading and Conceptual Questions (10)	10%		0%
04/Aug	2 Classical Systems Engineering Diagrams (IDEF-0/N2/eFFBD/DFD)	[4]	IA-02 - Exercises	0%	GA-02 - Preparation of representation of your system using classical Diagrams	50%
11/Aug	3 Transition from Legacy to MBSE MBSE Methodologies MBSE Languages	[5][7]	IA-03 - Reading and Conceptual Questions (10)	10%		0%
18/Aug	4 OPM - Basic	[6]	IA-04 - Exercises	10%		0%
25/Aug	5 OPM - Extended	[6]	IA-05 - Exercises	10%		0%
01/Sep	6 OPM - Group Presentation		IA-06 - Exercises	0%	G6 - Prepare a presentation of your system using OPM	50%
08/Sep	7 SysML Introduction (bdd/ibd)	[7]	IA-07 - Exercises	10%		0%
15/Sep	8 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
<b>9</b> 29/Sep	SysML (act/stm)	[7]	IA-9 - Exercises	10%	GA-09 -	0%
<b>10</b> 06/Oct	SysML (seq/uc)	[7]	IA-10 - Exercises	10%	GA-10 -	0%
<b>11</b> 13/Oct	Simulation on SysML		IA-11 -	0%	GA-11 -	0%
<b>12</b> 20/Oct 21/Oct	SysML (pkg/req)	[7]	IA-12 - Exercises	10%	GA-12 -	0%
<b>13</b> 27/Oct	Arcadia process applied into the SysML	[5]	IA-13 -	0%	GA-13 -	0%
<b>14</b> 03/Nov	Some System Analysis on SysML SysML V2 Perspectives	[8]	IA-14 -	0%	GA-14 -	0%
<b>15</b> 10/Nov	SysML Group Presentation		IA-15 -	0%	GA-15 - Prepare a presentation of your system using SysML	100%
	Course Ending					
<b>16</b> 17/Nov	P2 - Conceptual Questions and Case	[5] [7]	IA-16 - Questions and a mini-case	70%	GA-16 -	
				<b>100%</b>		<b>100%</b>
<b>EXAM</b>						
24/Nov 08/Dec	If necessary: Writing an article (min 6pgs / max 10pgs) reporting the case of their group in the SIGE standard.					<b>100%</b>



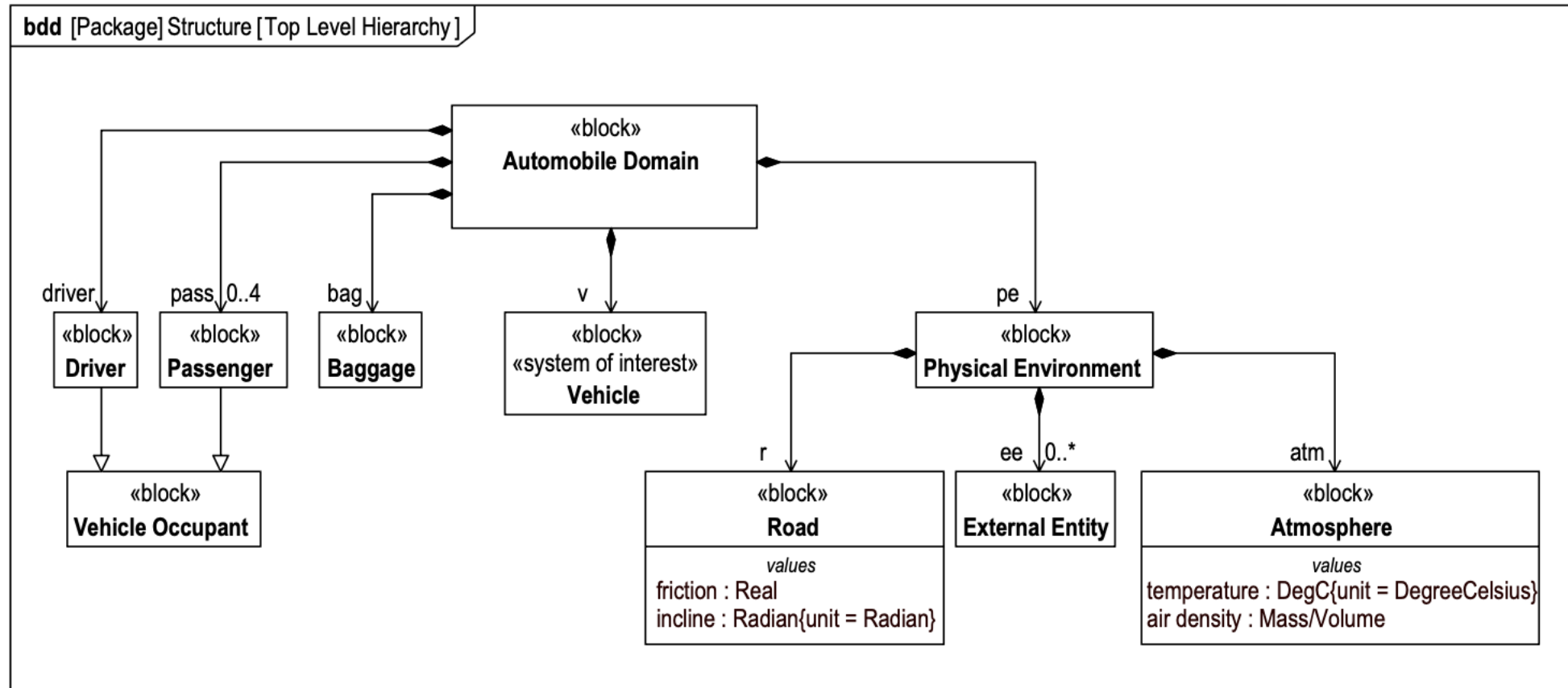
**FIGURE 3.3**

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

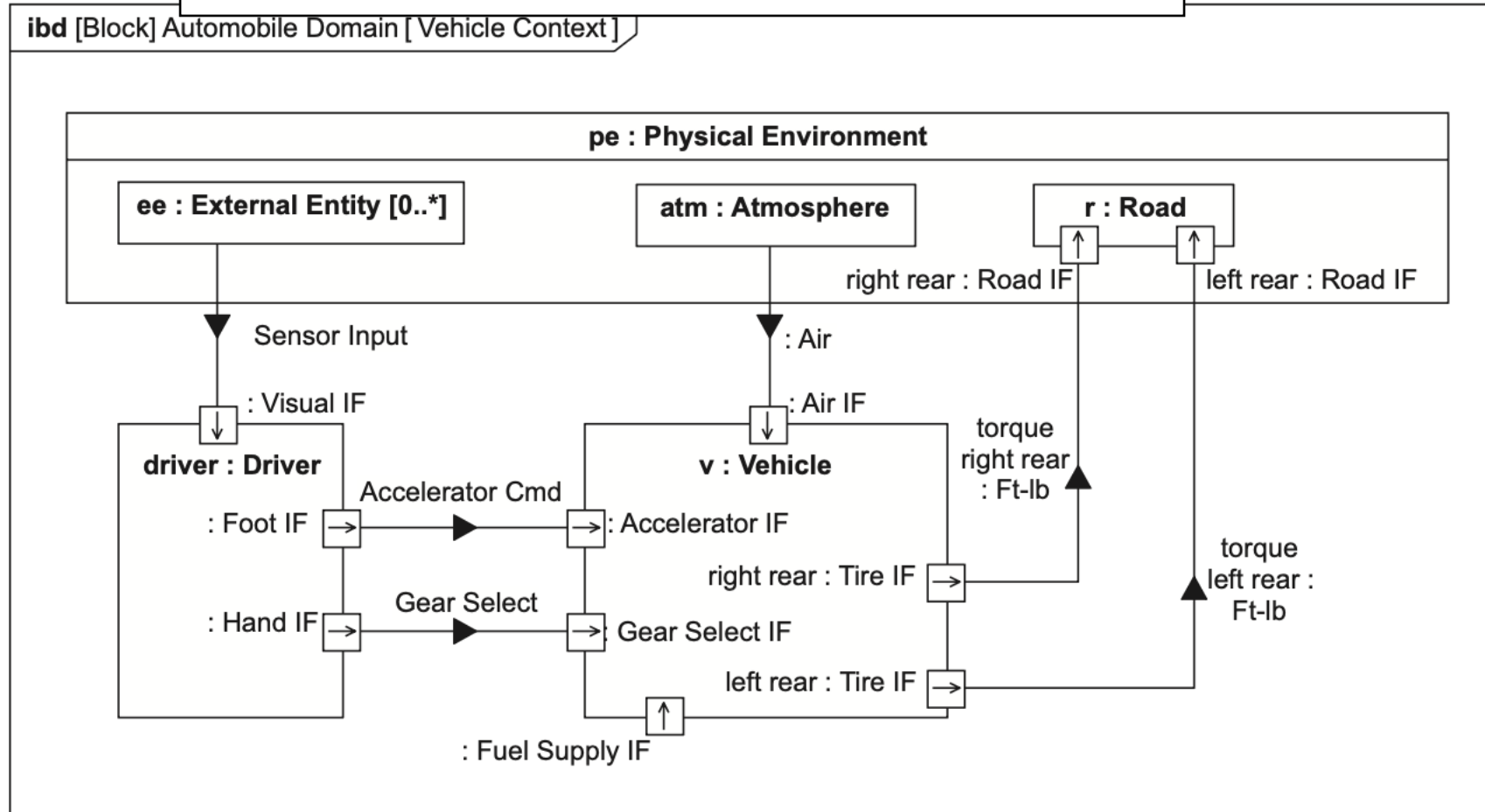


**FIGURE 4.3**

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

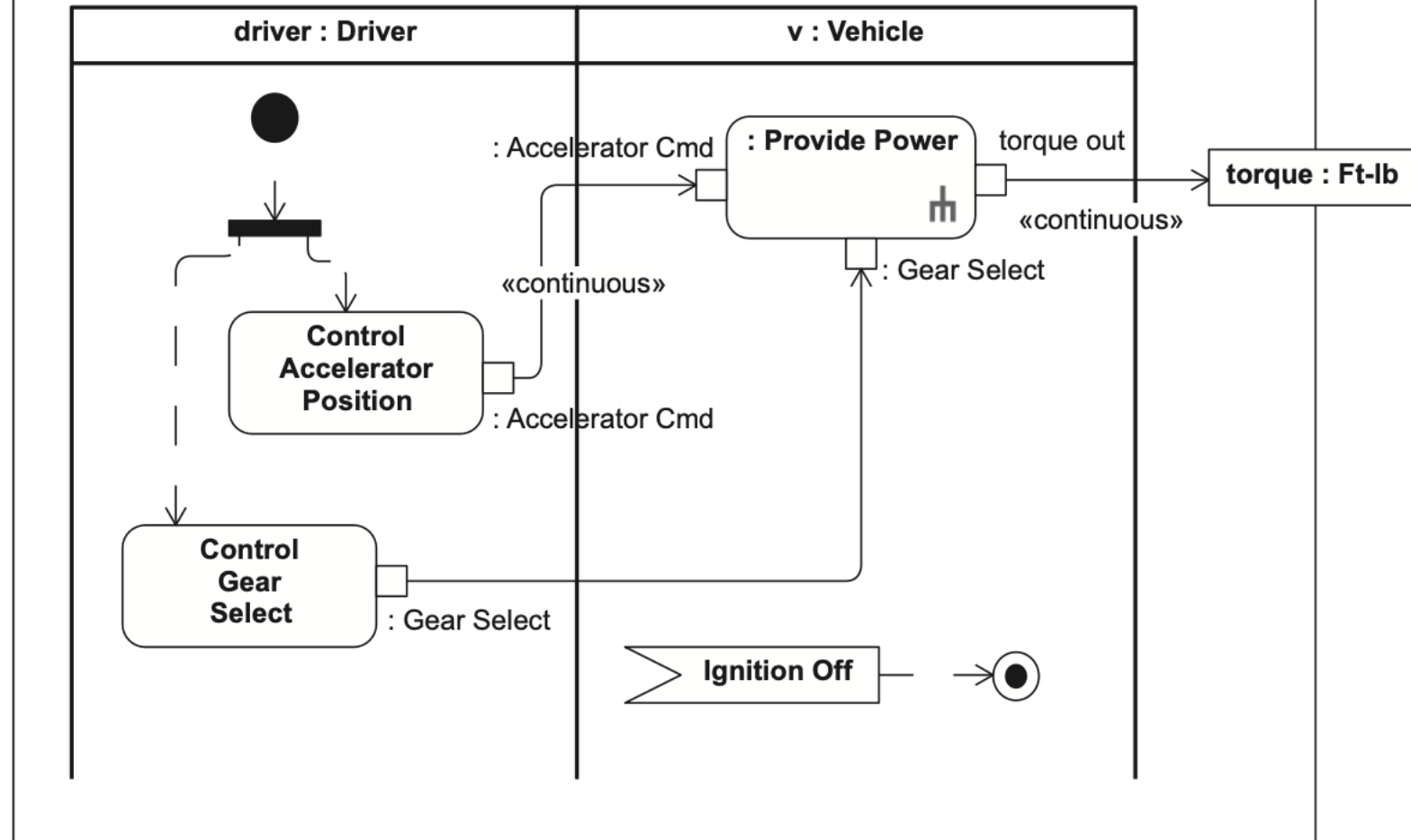


**FIGURE 4.9**

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](#).

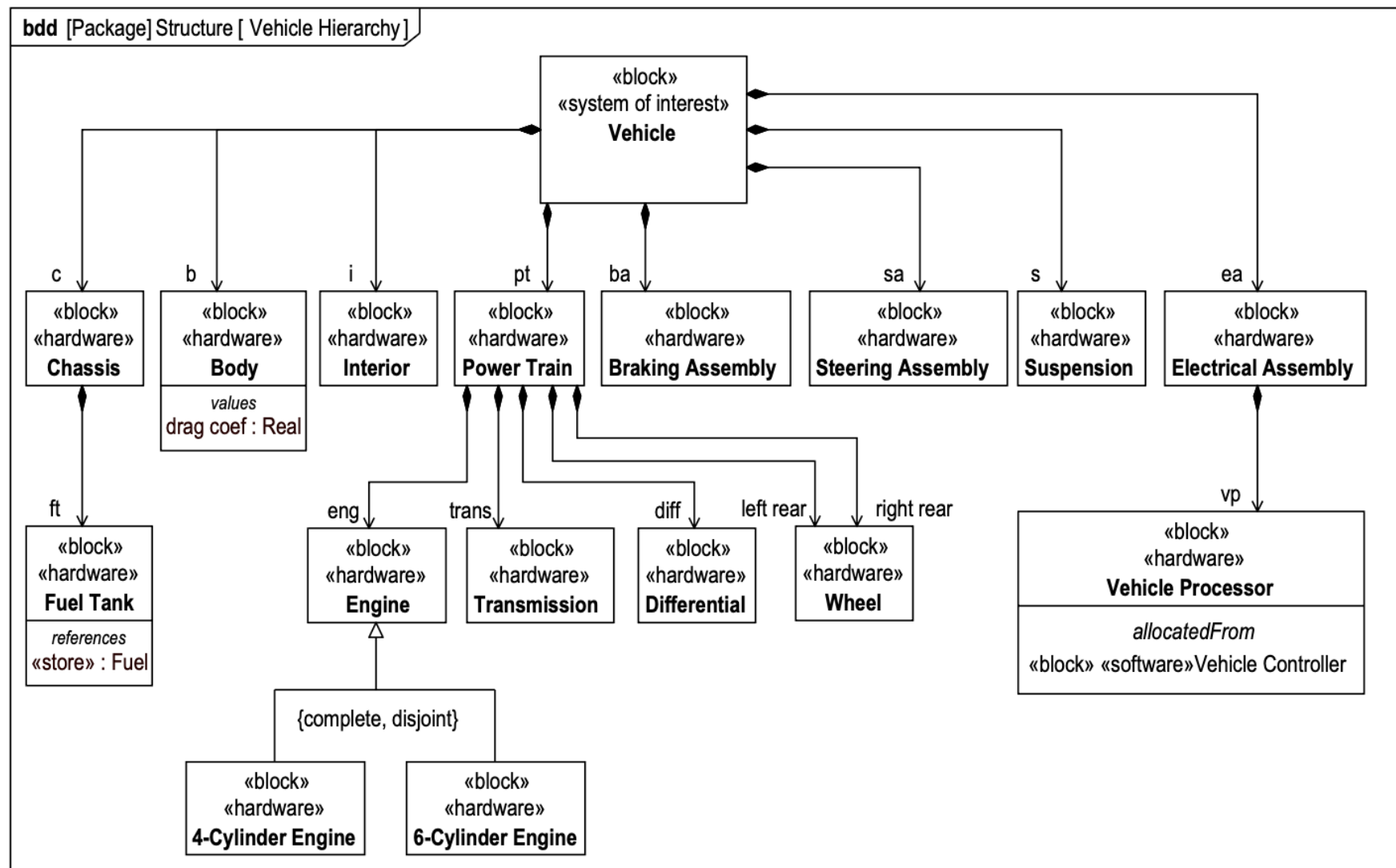


# Descrever a sequencia funcional dos elementos principais



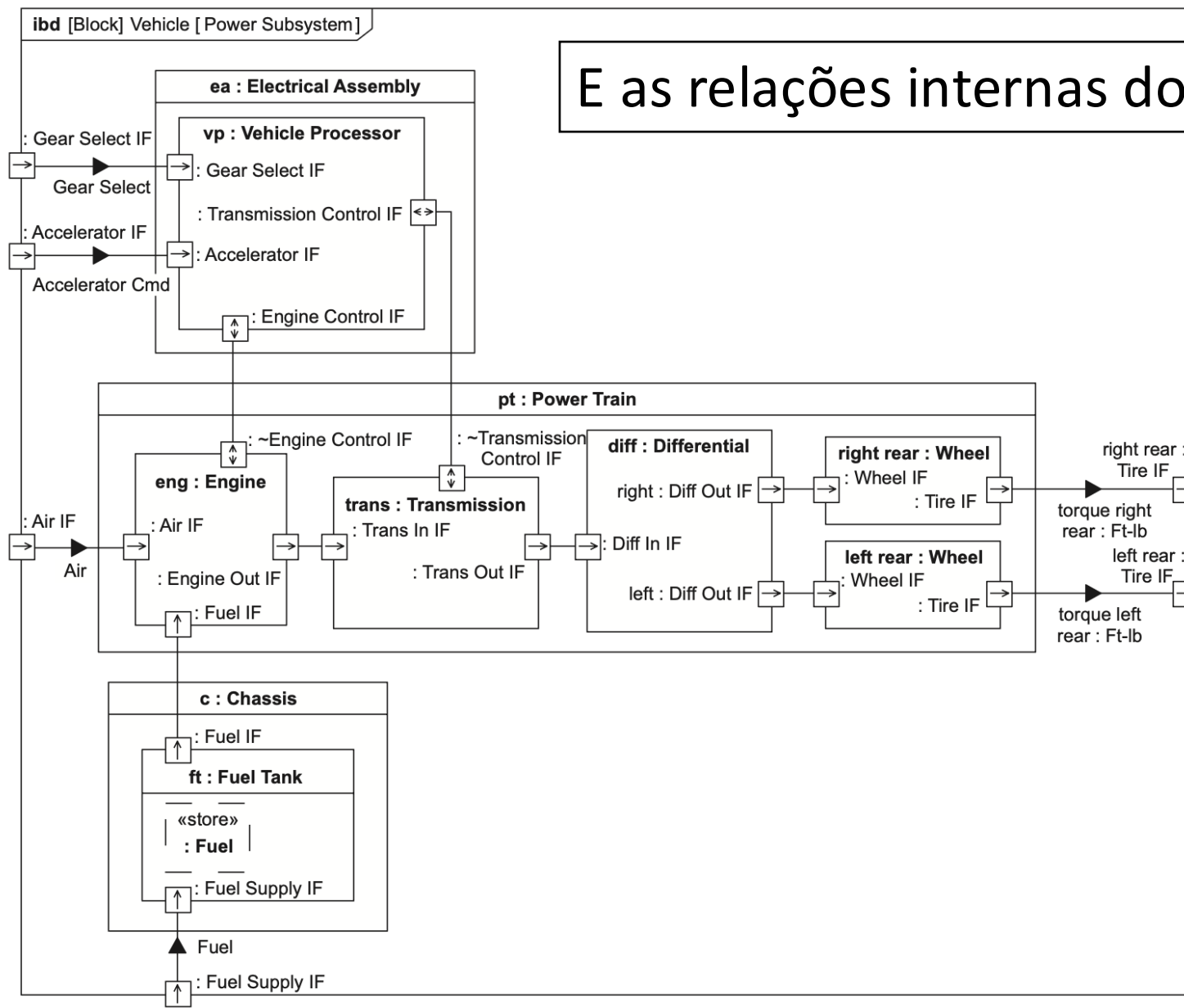
**FIGURE 4.7**

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

e

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

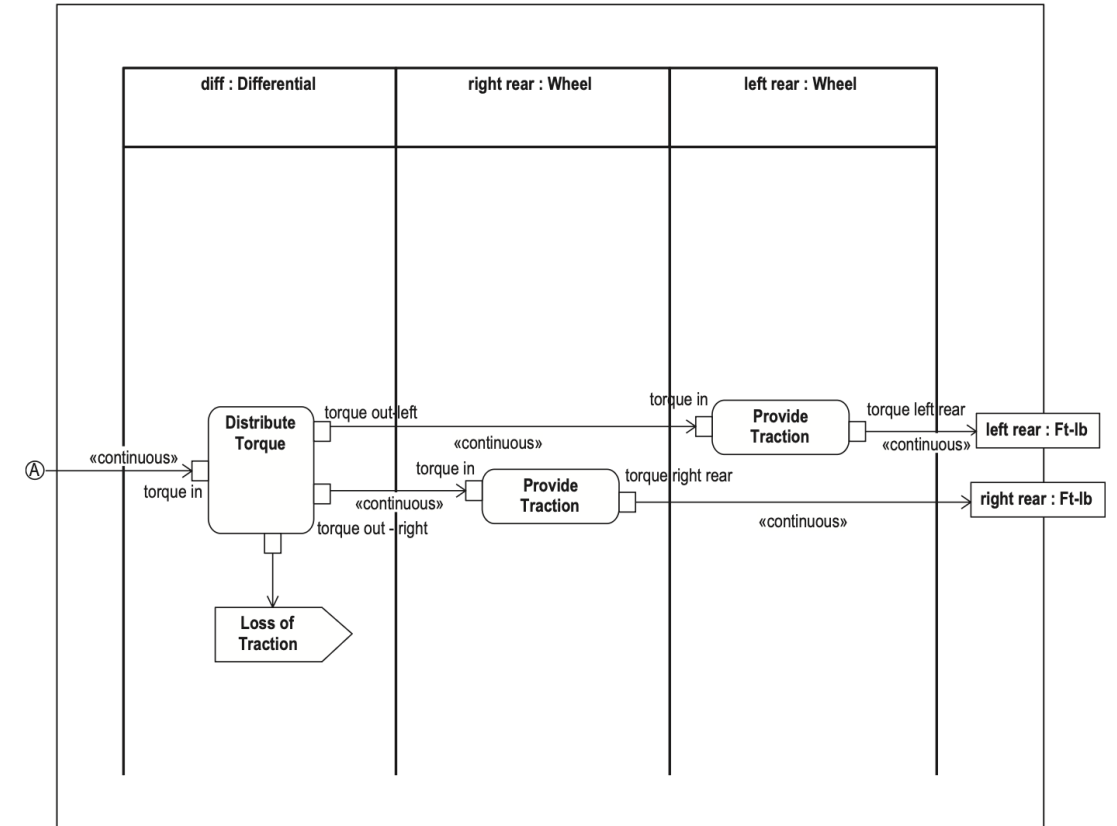
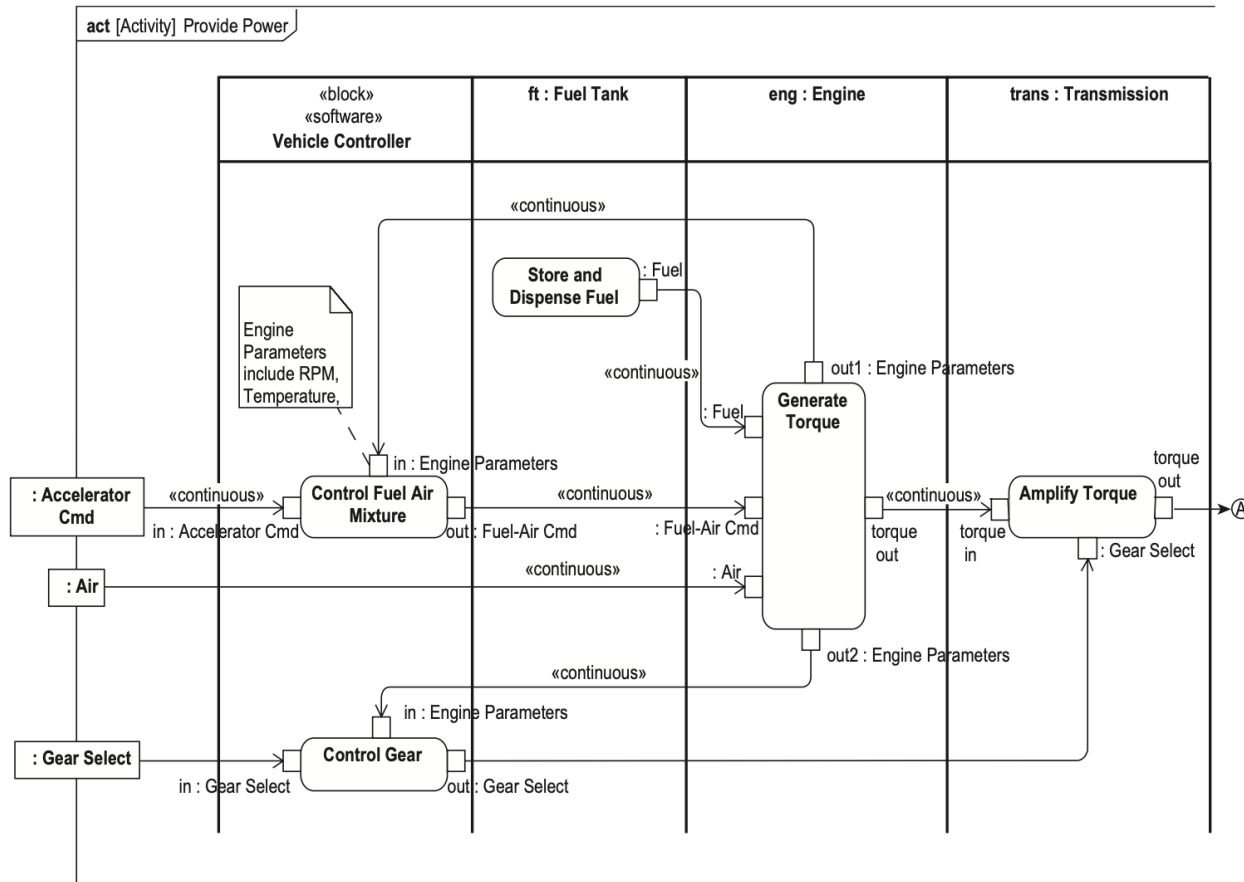


FIGURE 4.11

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.

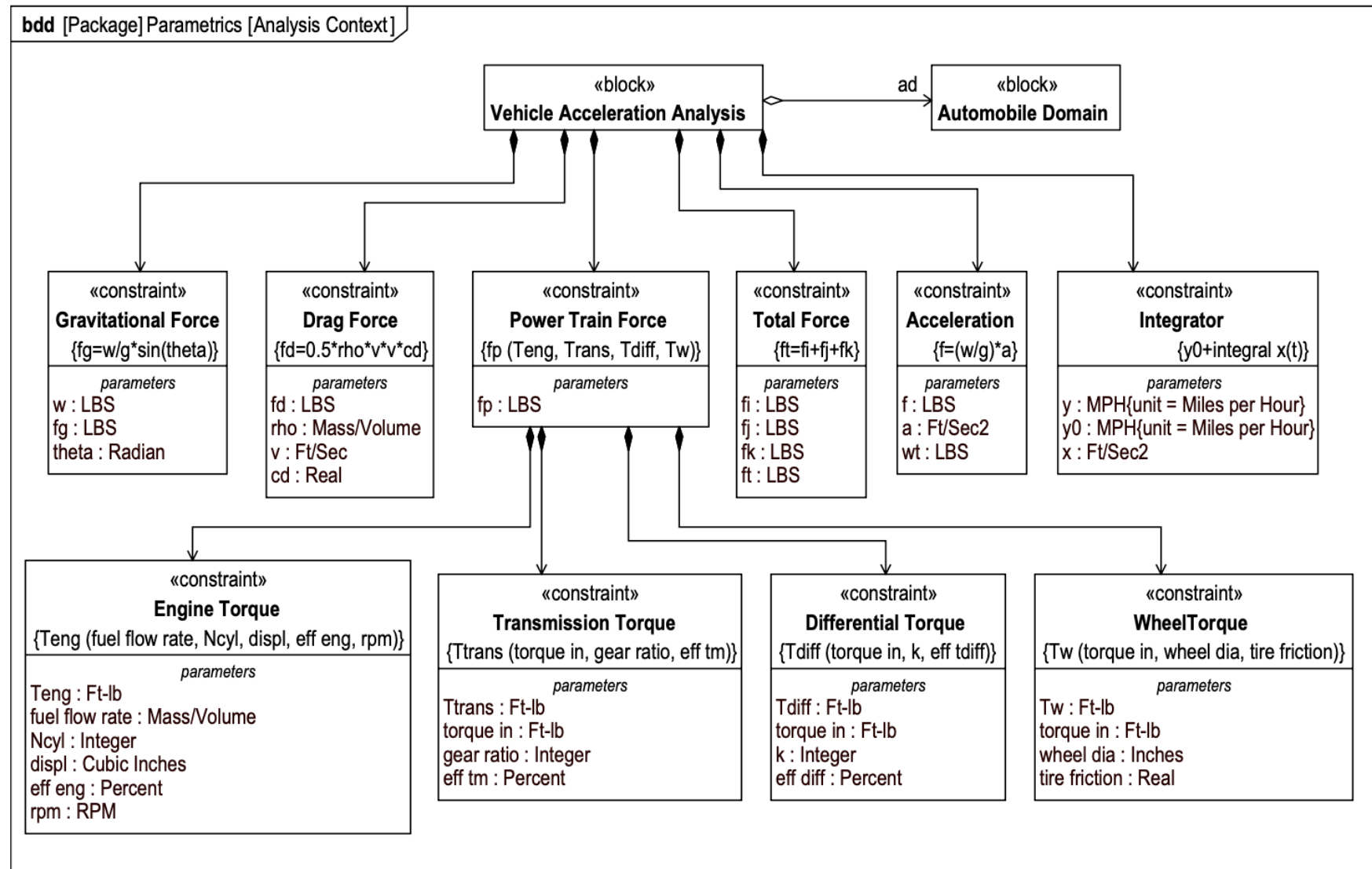


FIGURE 4.13

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.

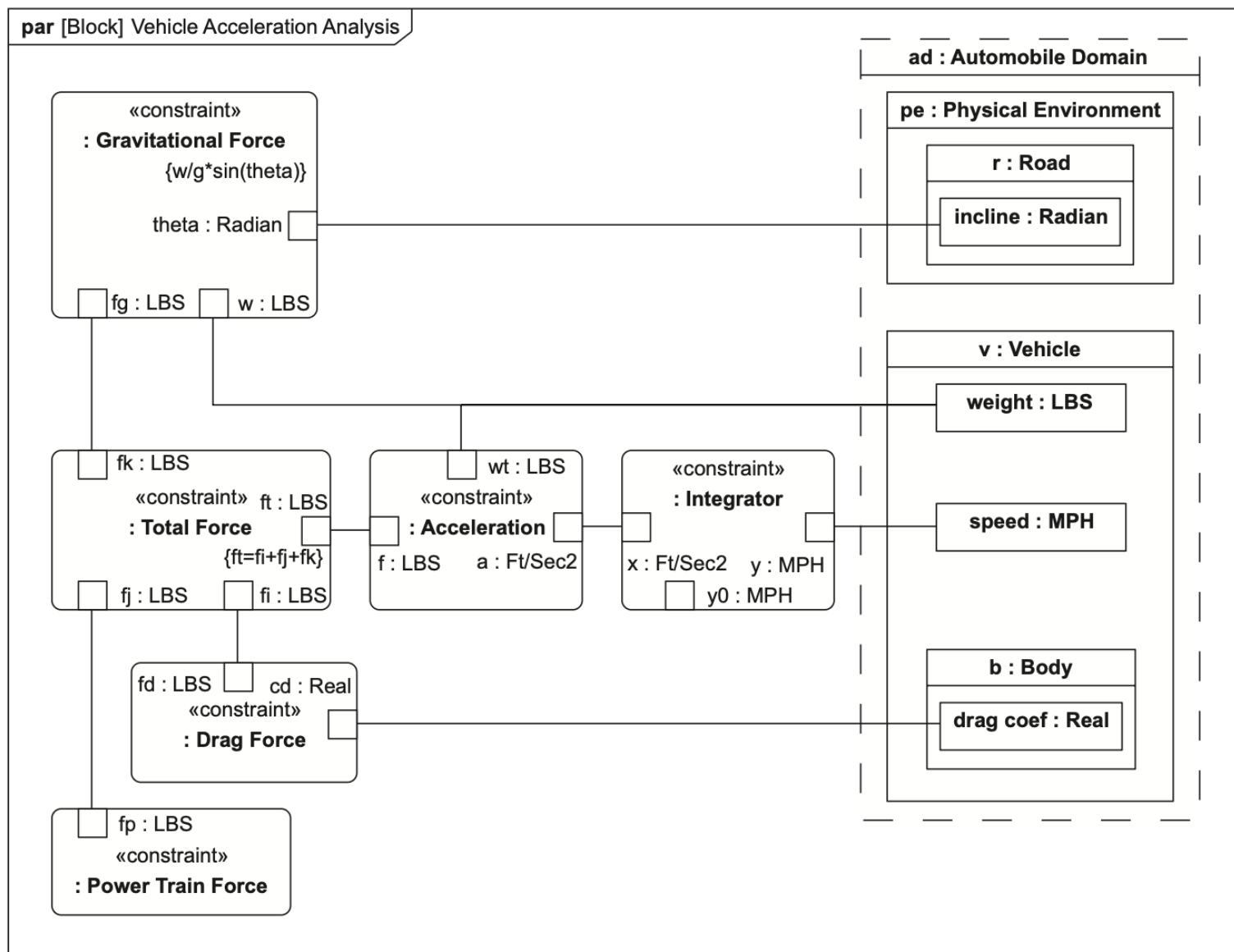


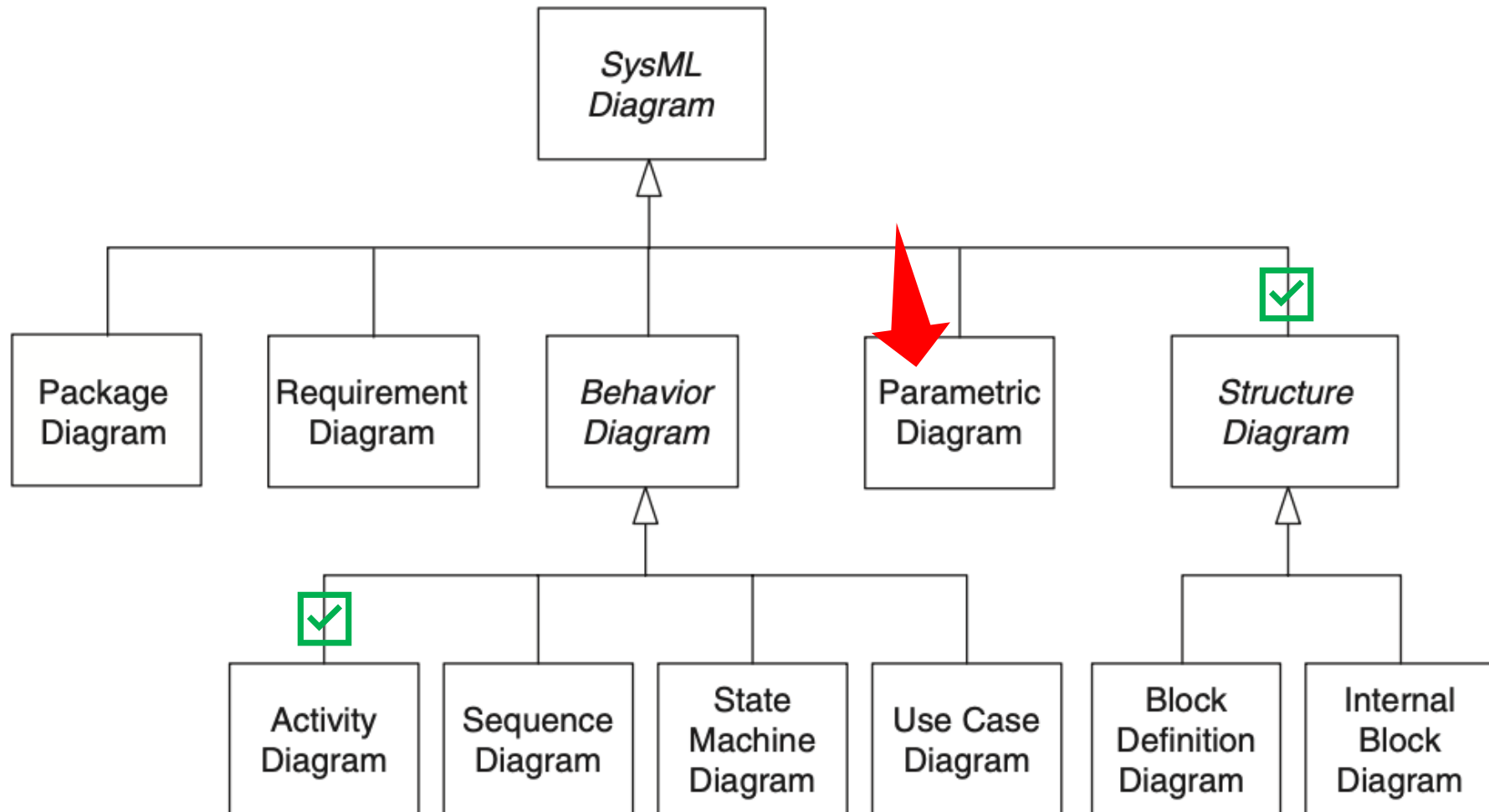
FIGURE 4.14

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.



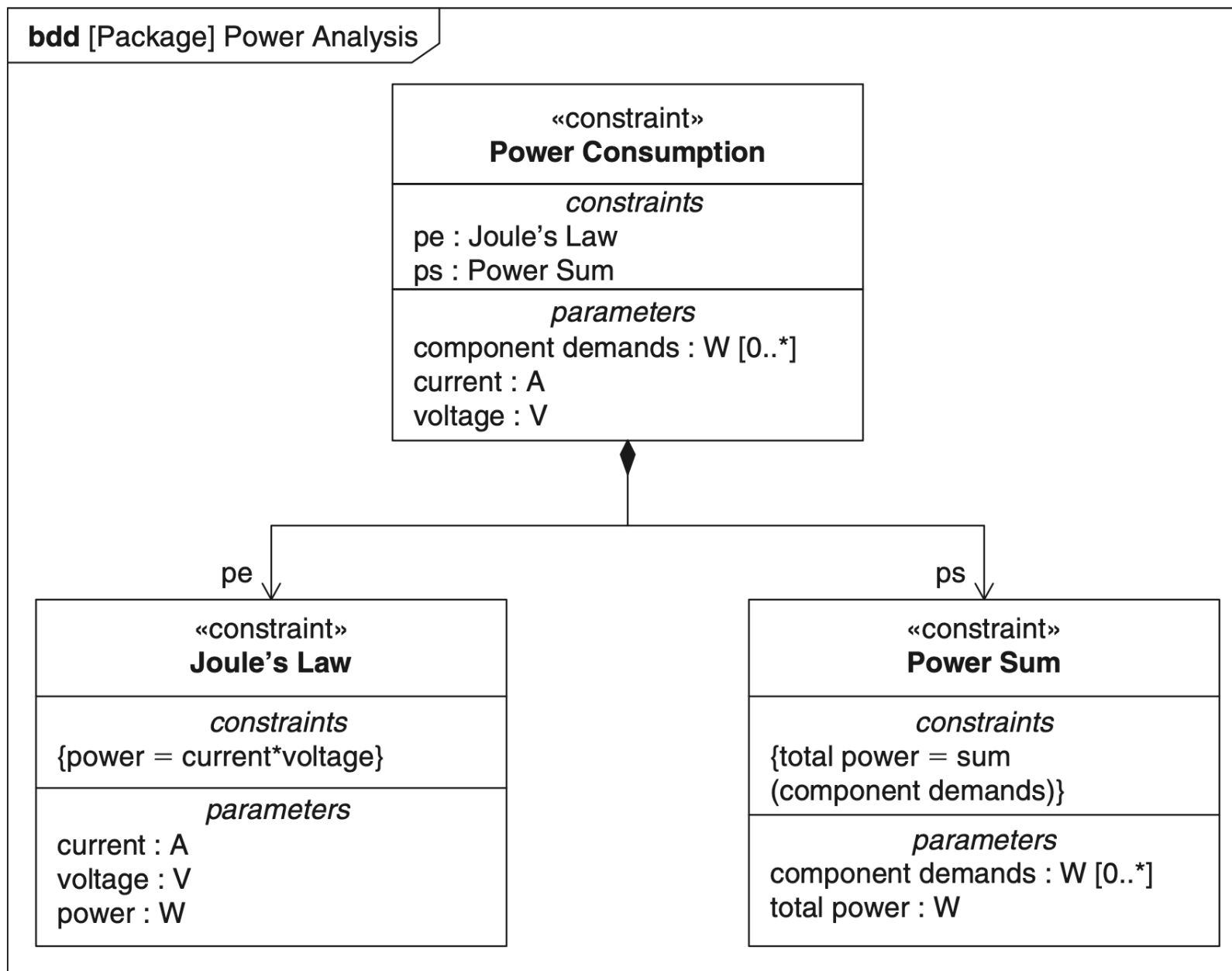
# Intro: Parametros são propriedades/restrições

- 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]*

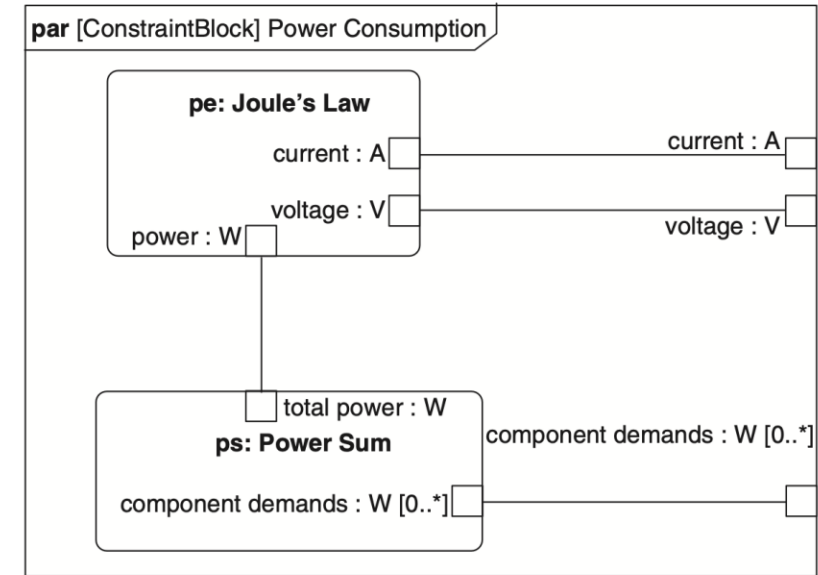


FIGURE 8.2

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.**

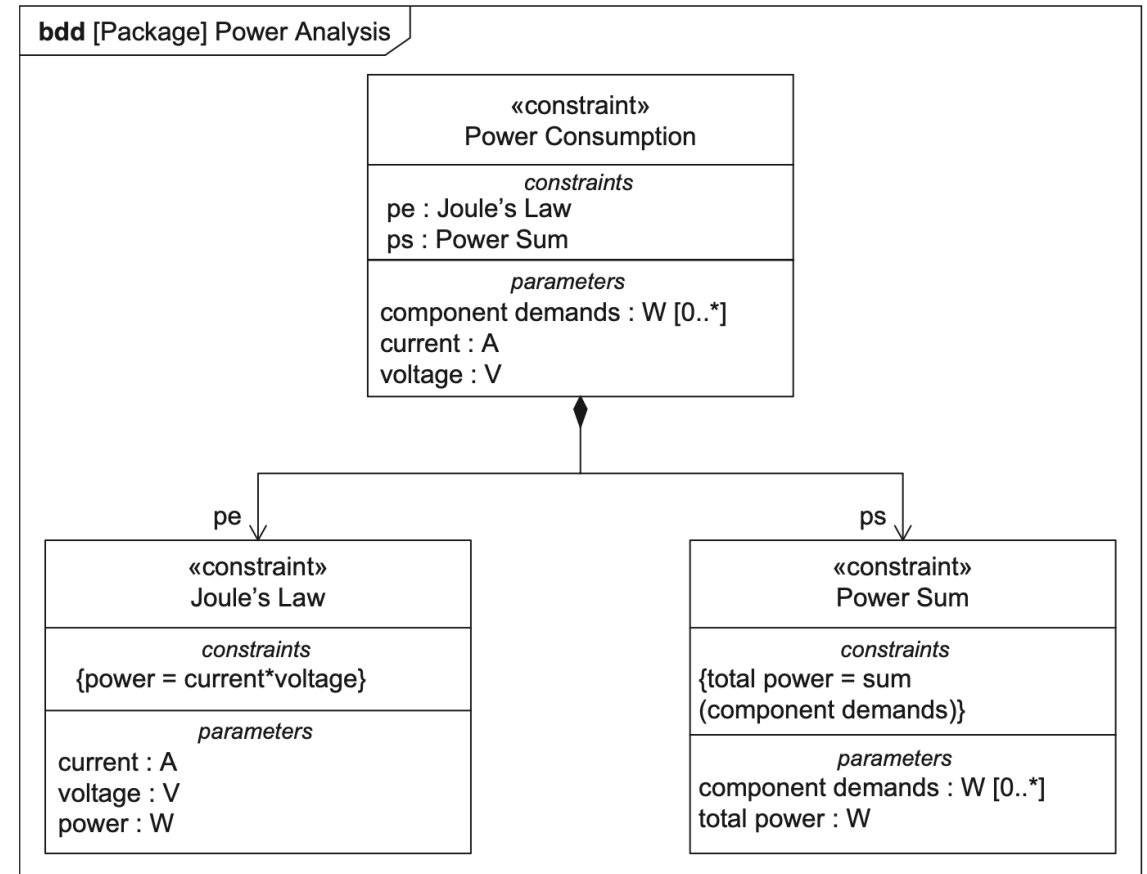
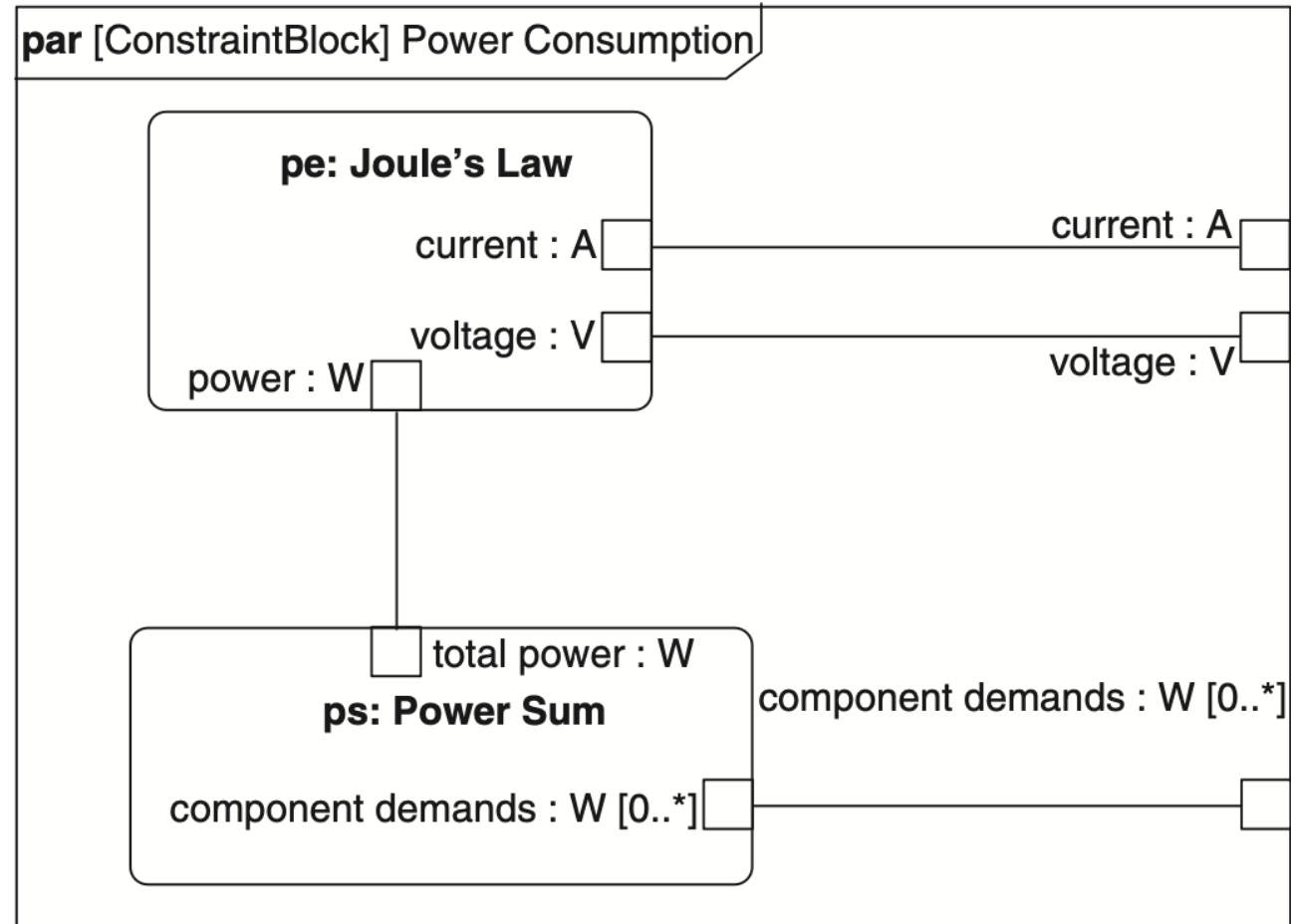


FIGURE 8.5

A hierarchy of constraints on a block definition diagram.



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

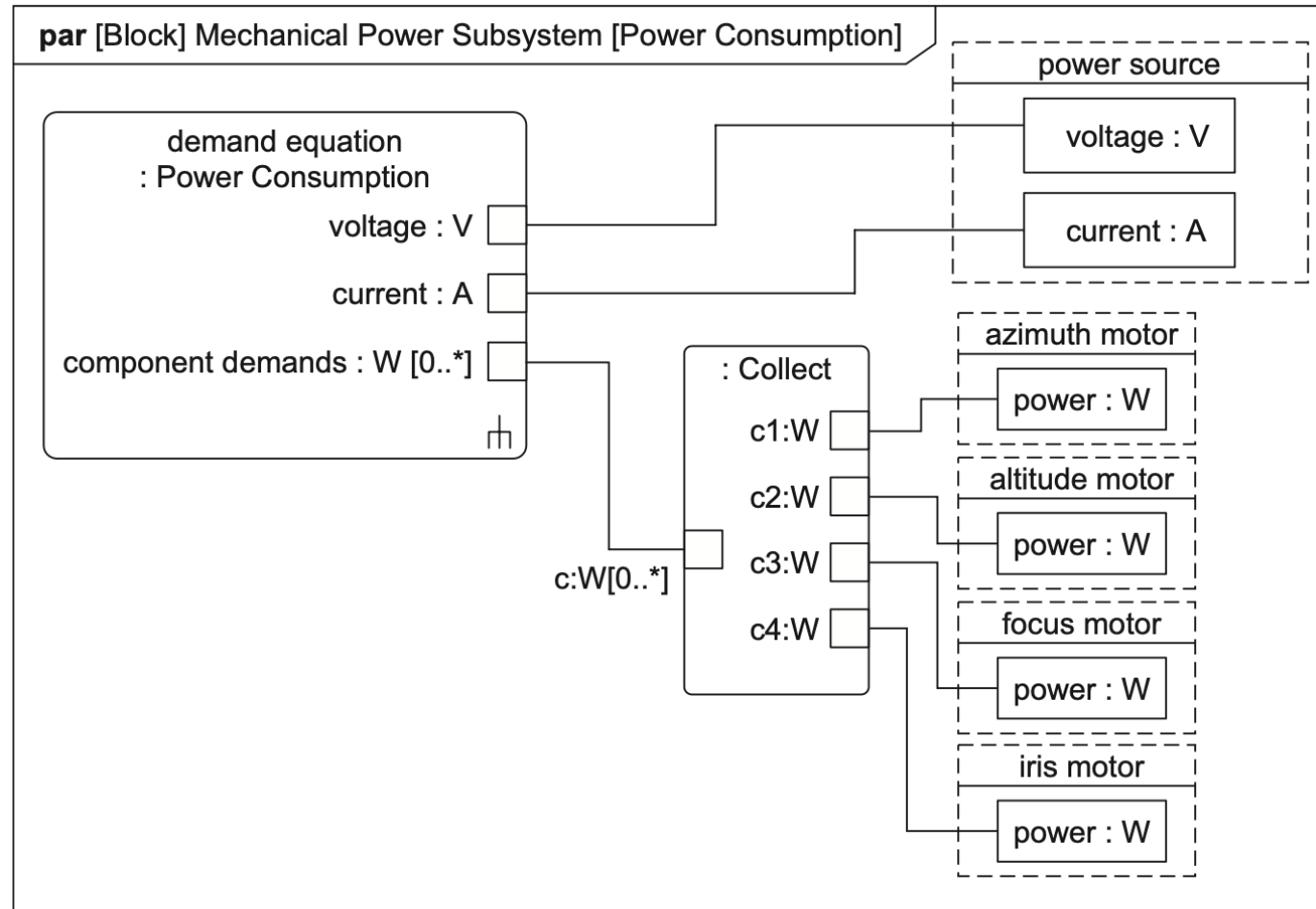


**FIGURE 8.6**

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.*

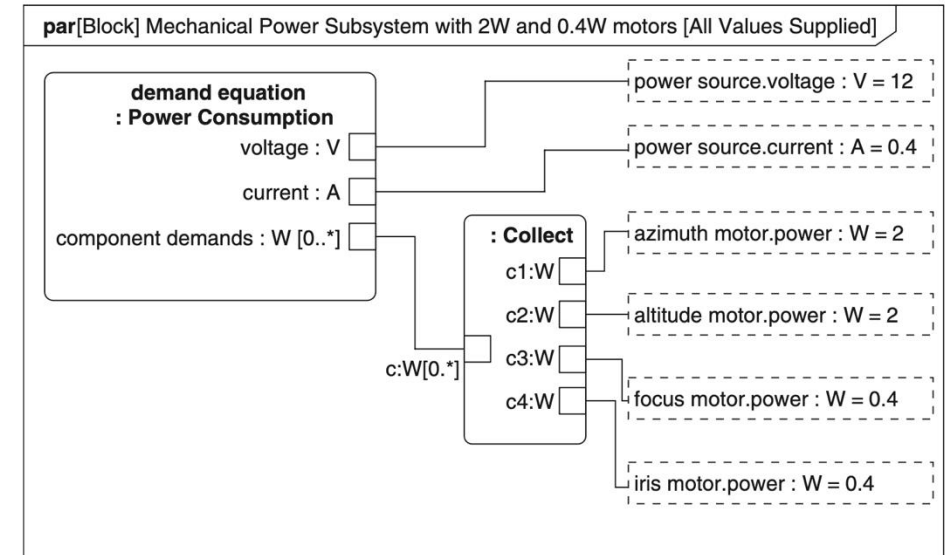


FIGURE 8.8

Describing a specific analysis configuration.

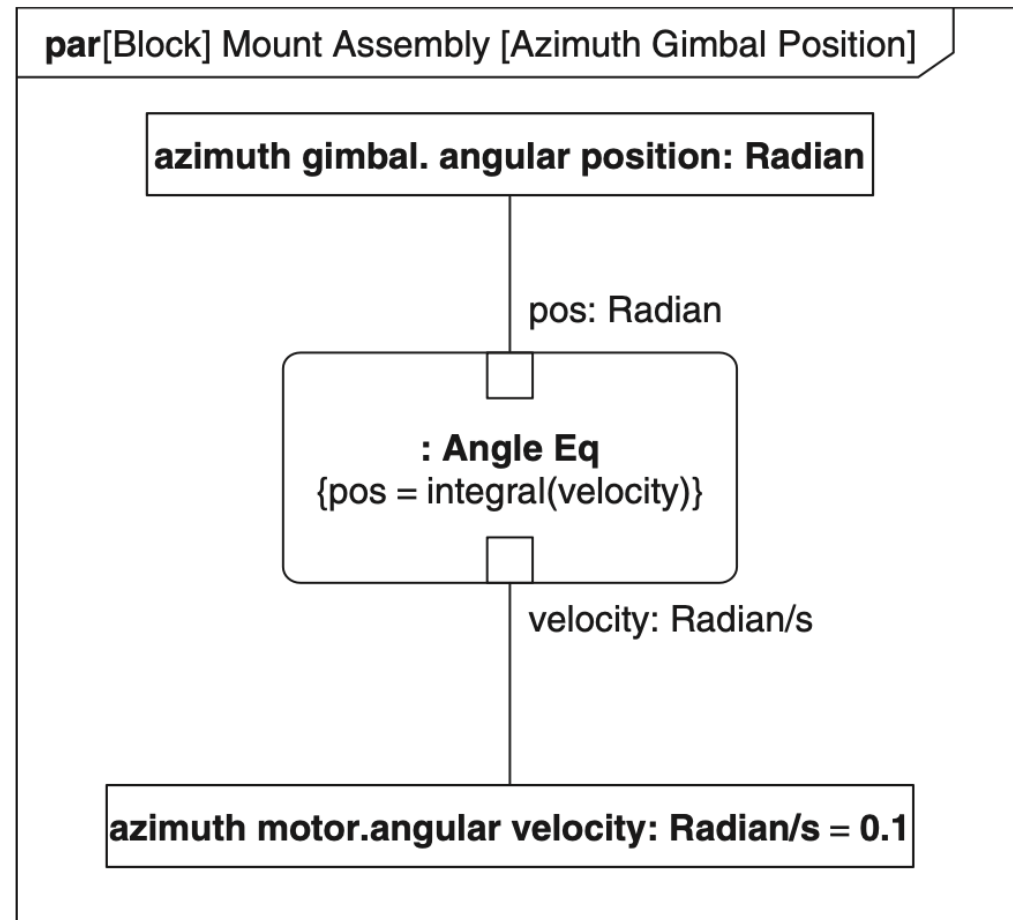


# Descrevendo propriedades temporais

- 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



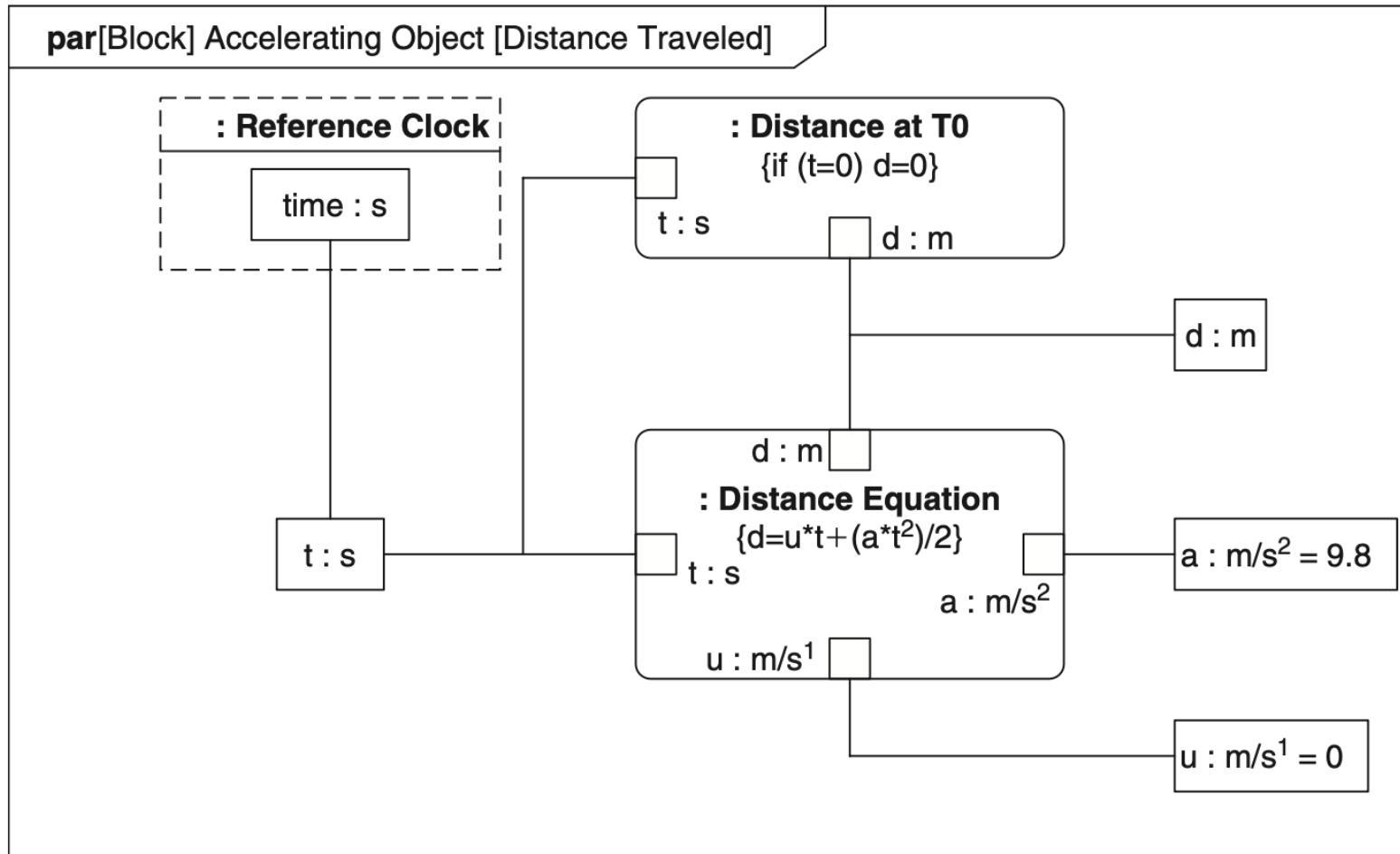
**FIGURE 8.9**

Using a time-dependent constraint.





# Tempo explicito entrando nos blocos

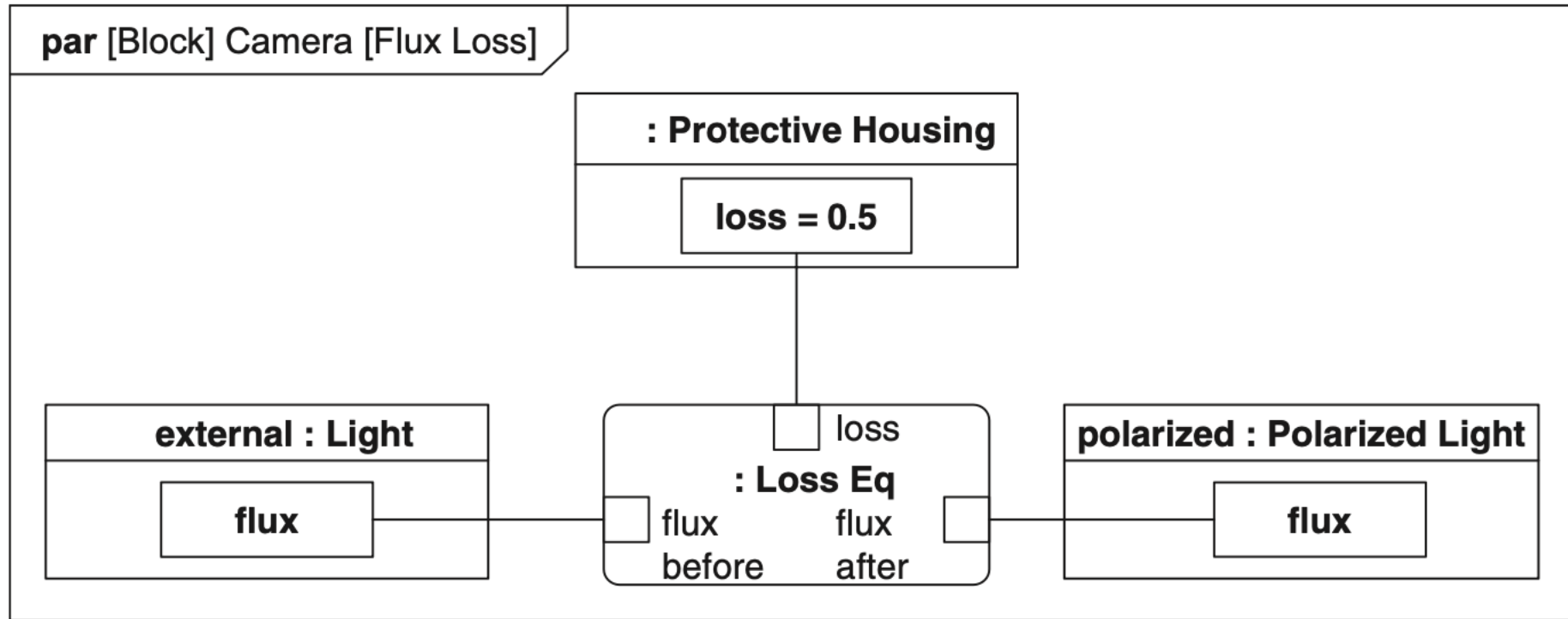


**FIGURE 8.10**

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.

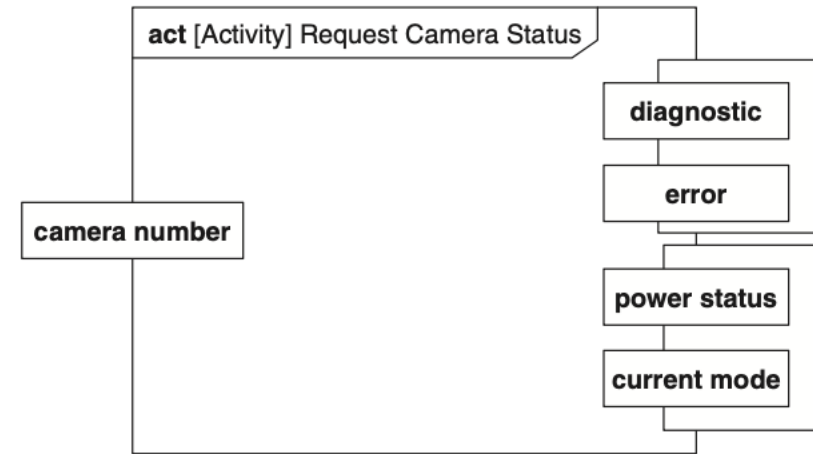


FIGURE 9.9

An activity with parameter sets.

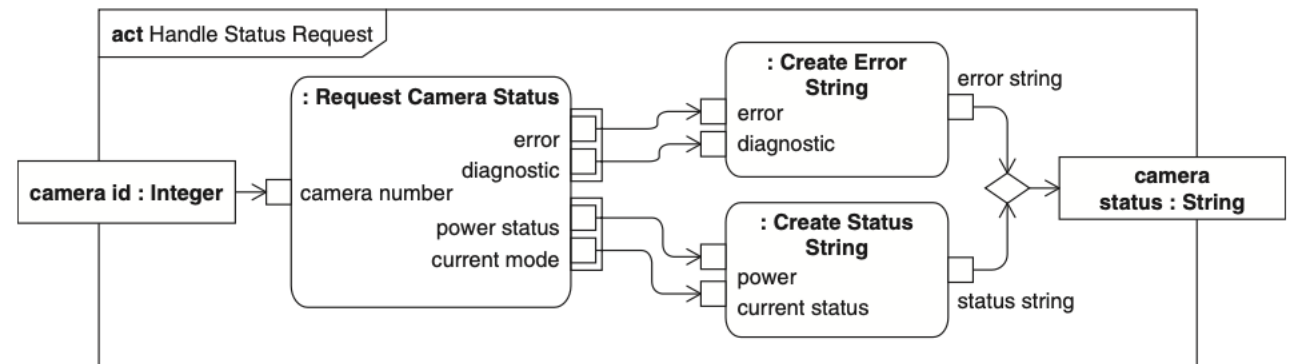


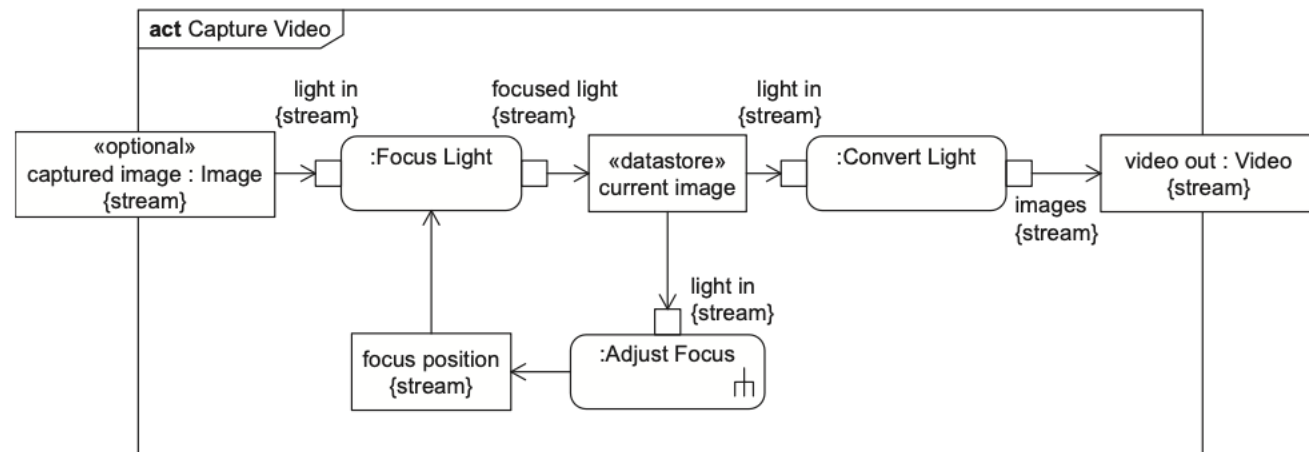
FIGURE 9.10

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.





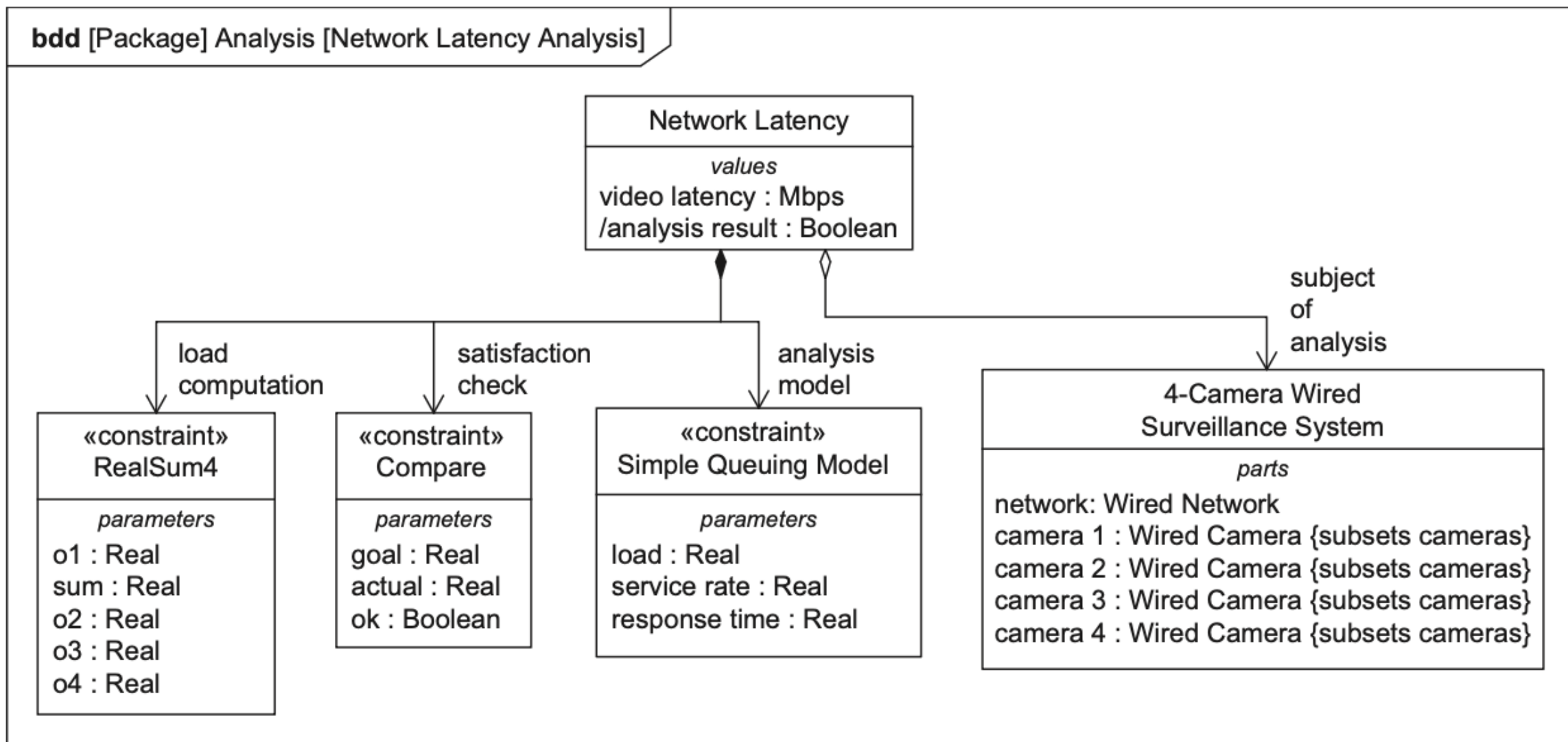
# Analysis Context



# 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.





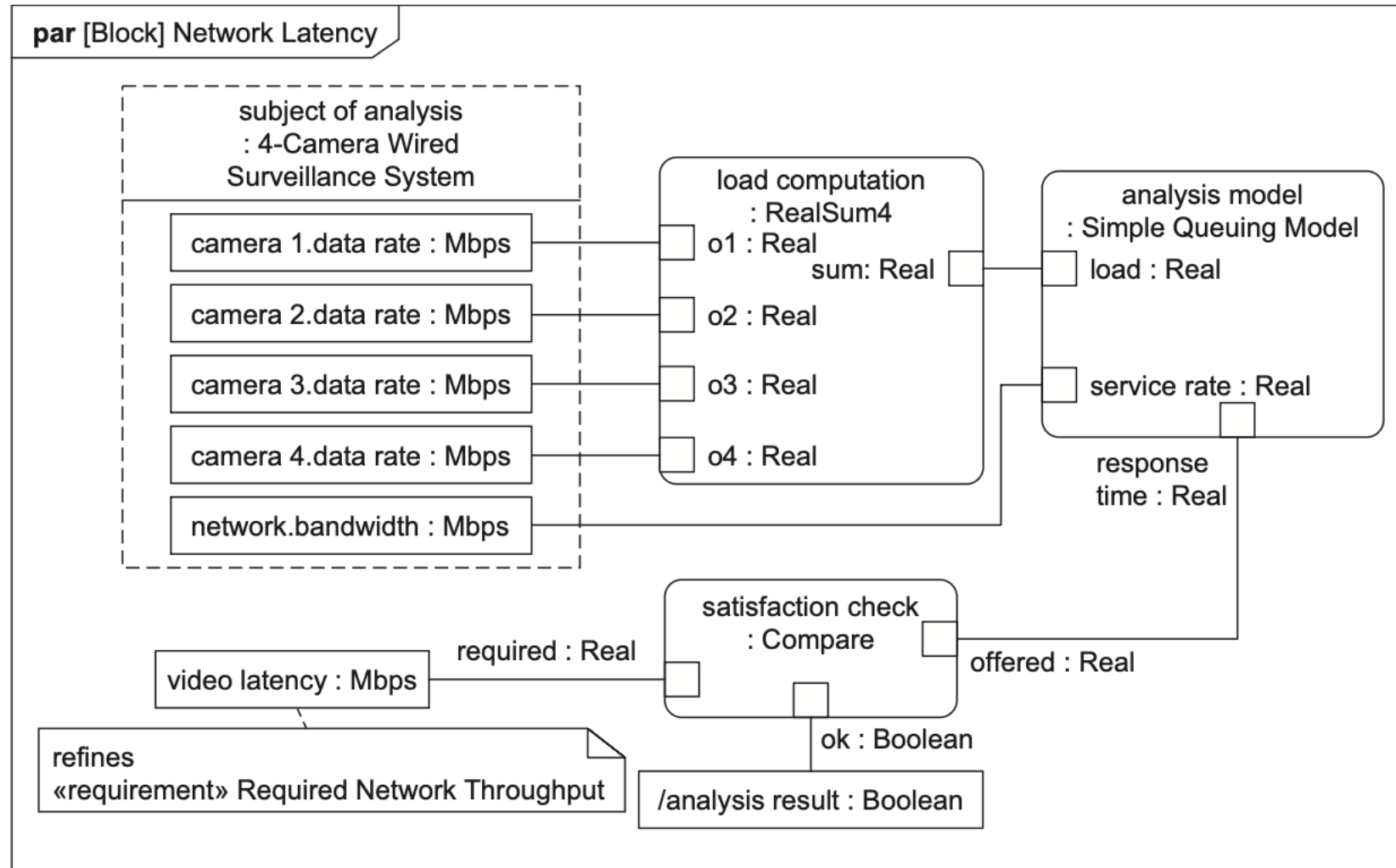
**FIGURE 8.12**

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.



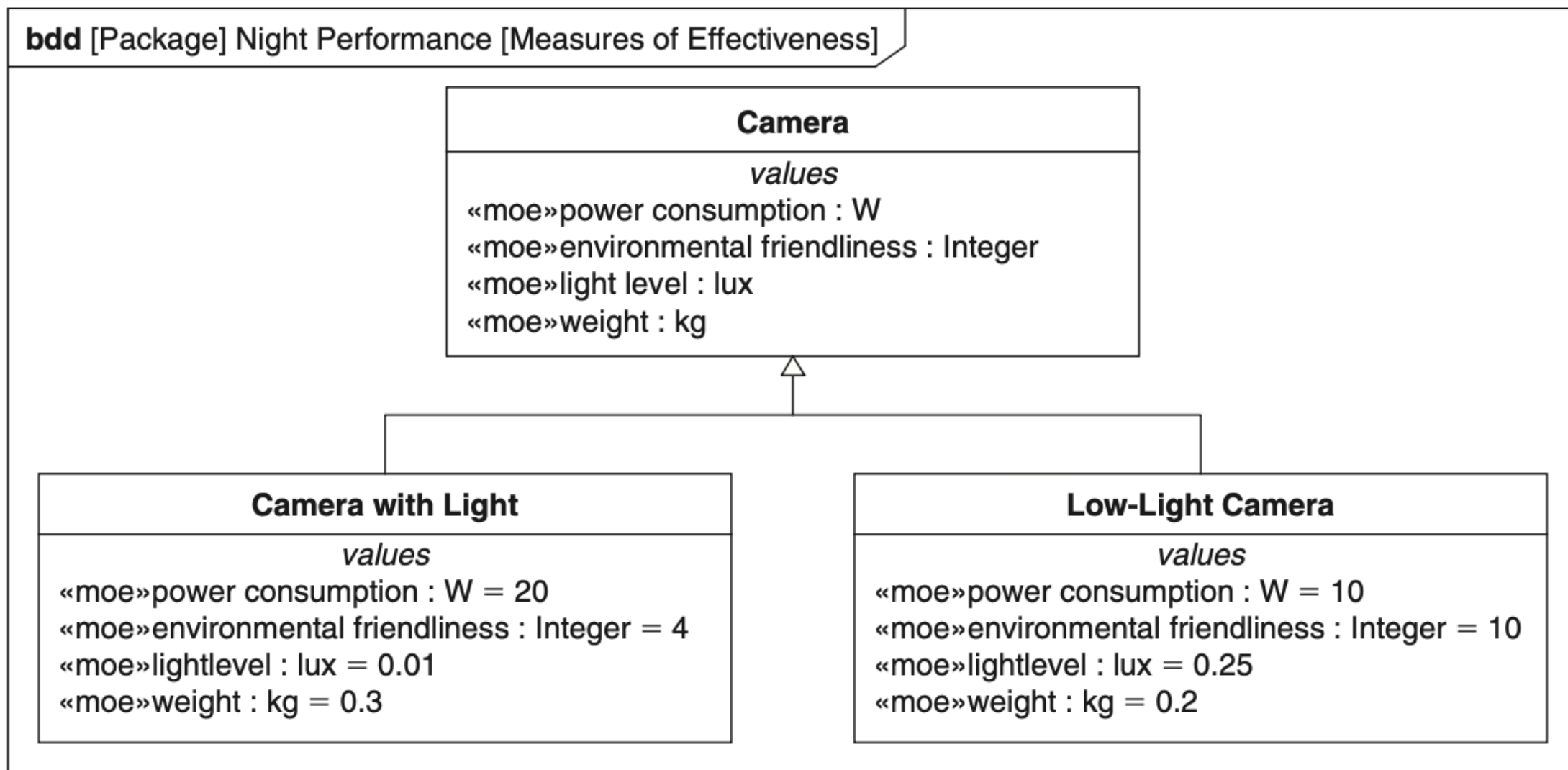
**FIGURE 8.13**

Binding values in an analysis context.



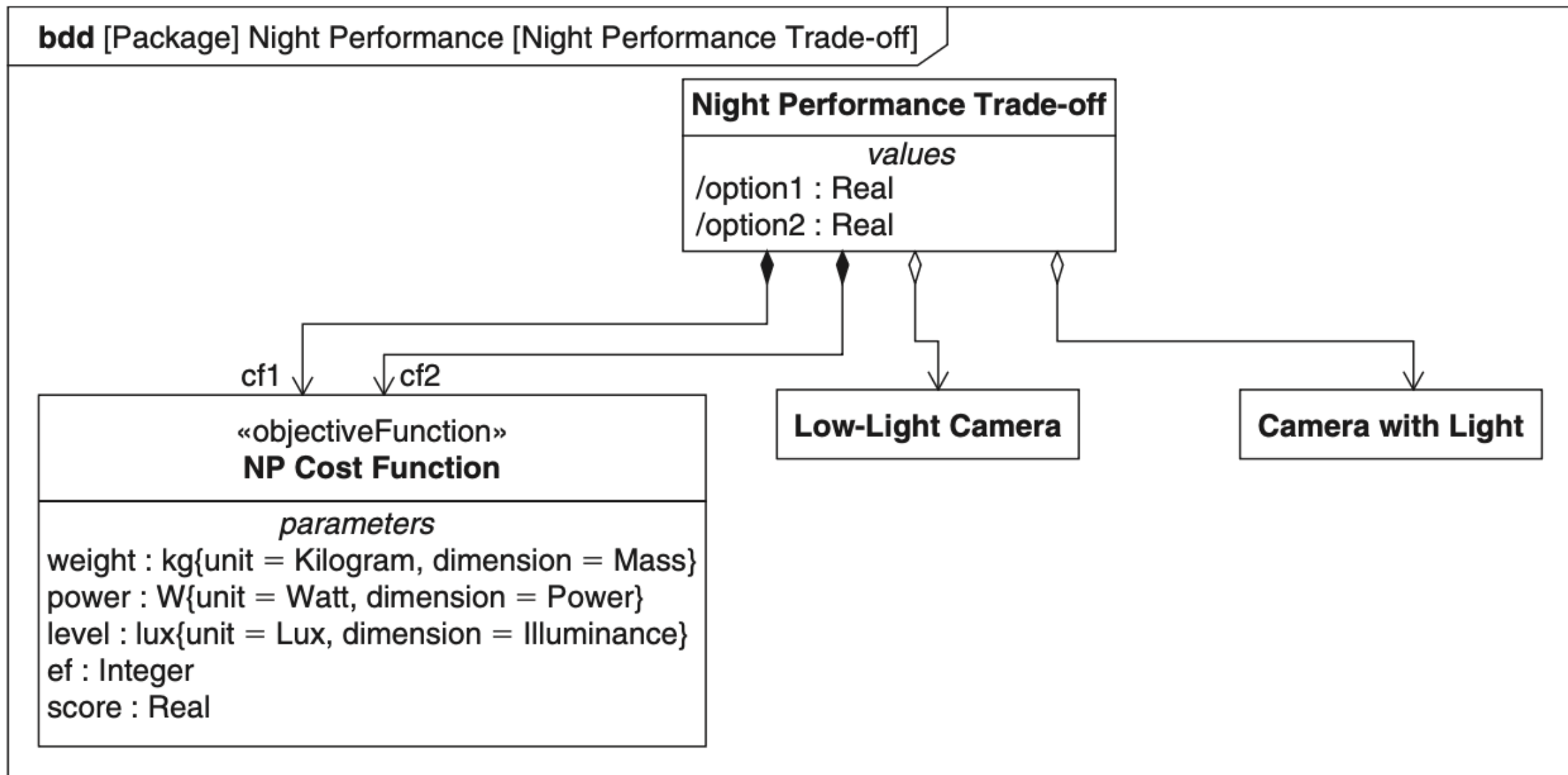
# Trade

- 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.



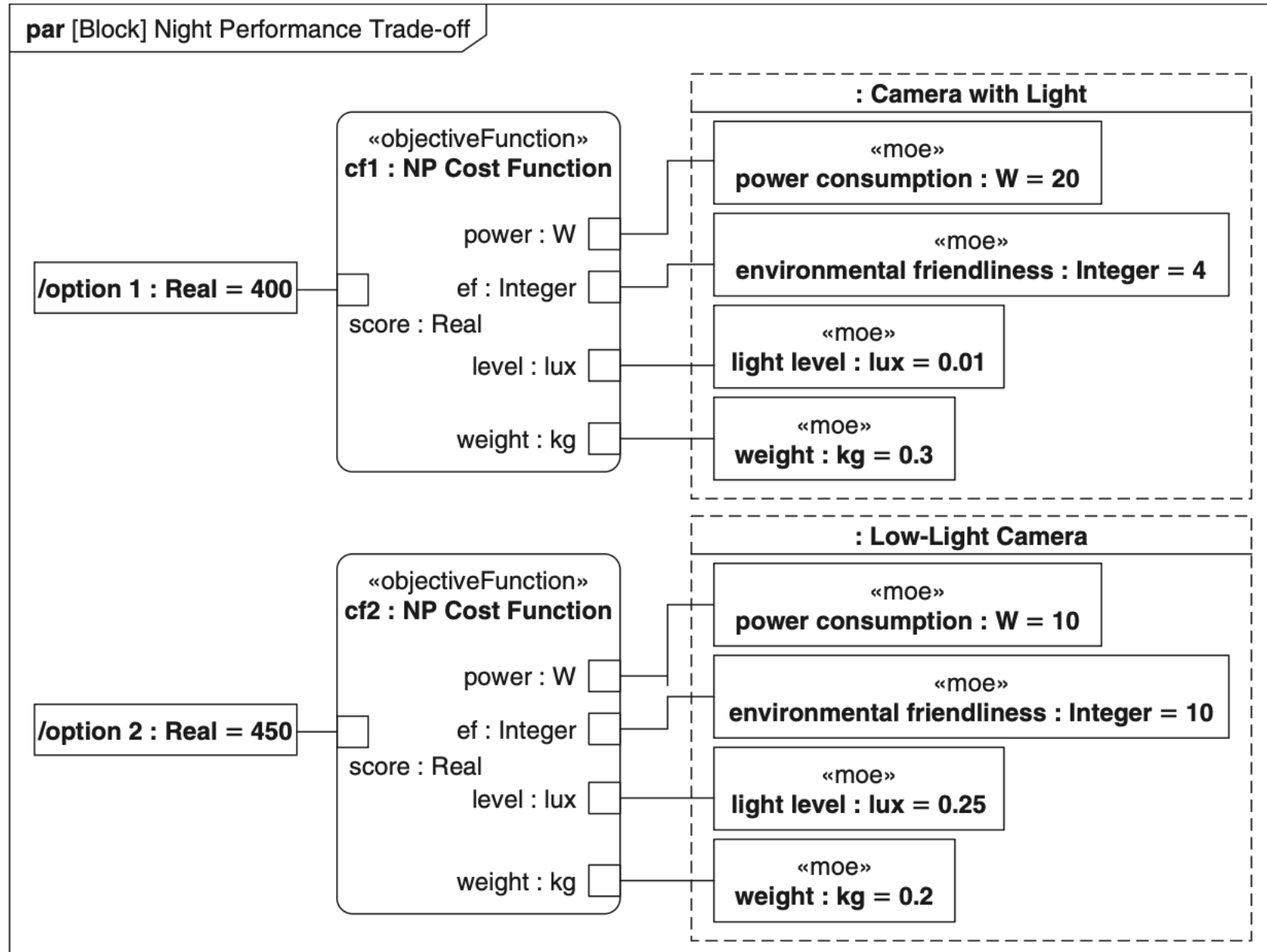
**FIGURE 8.14**

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!?



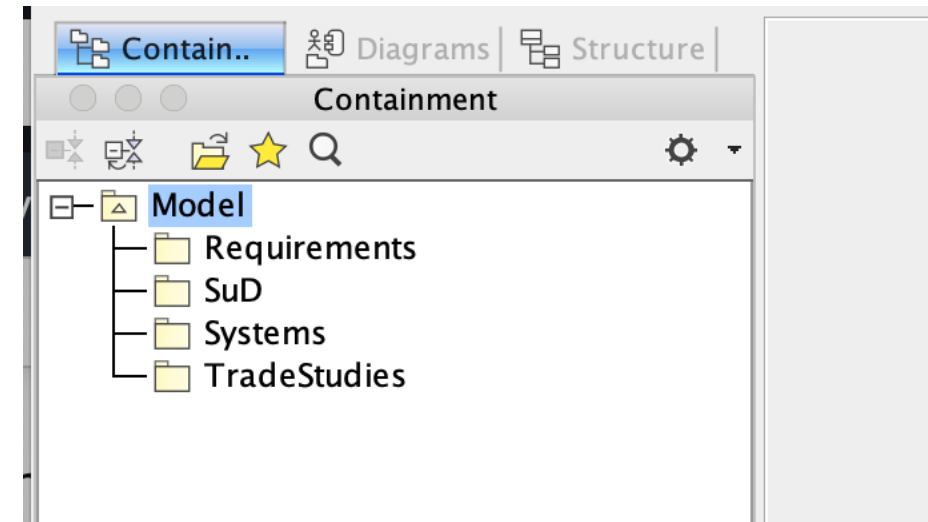
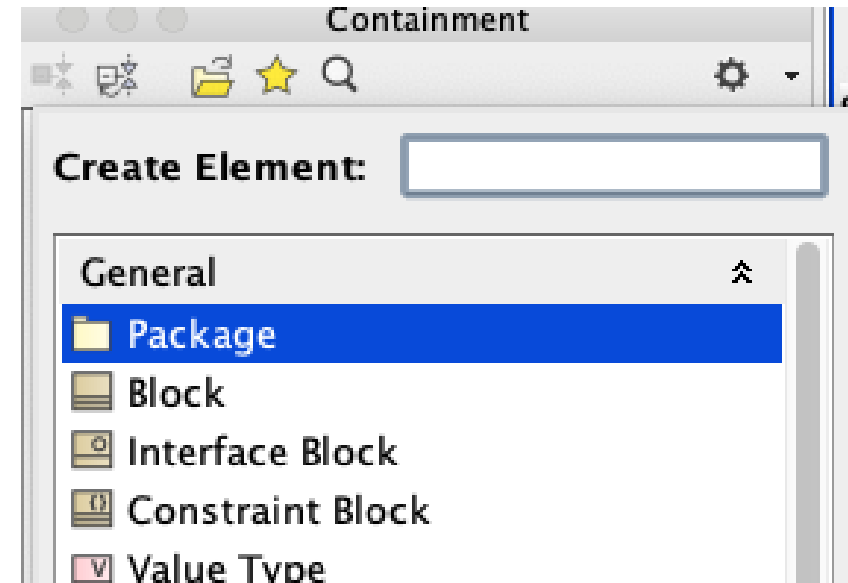


Step by step...



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

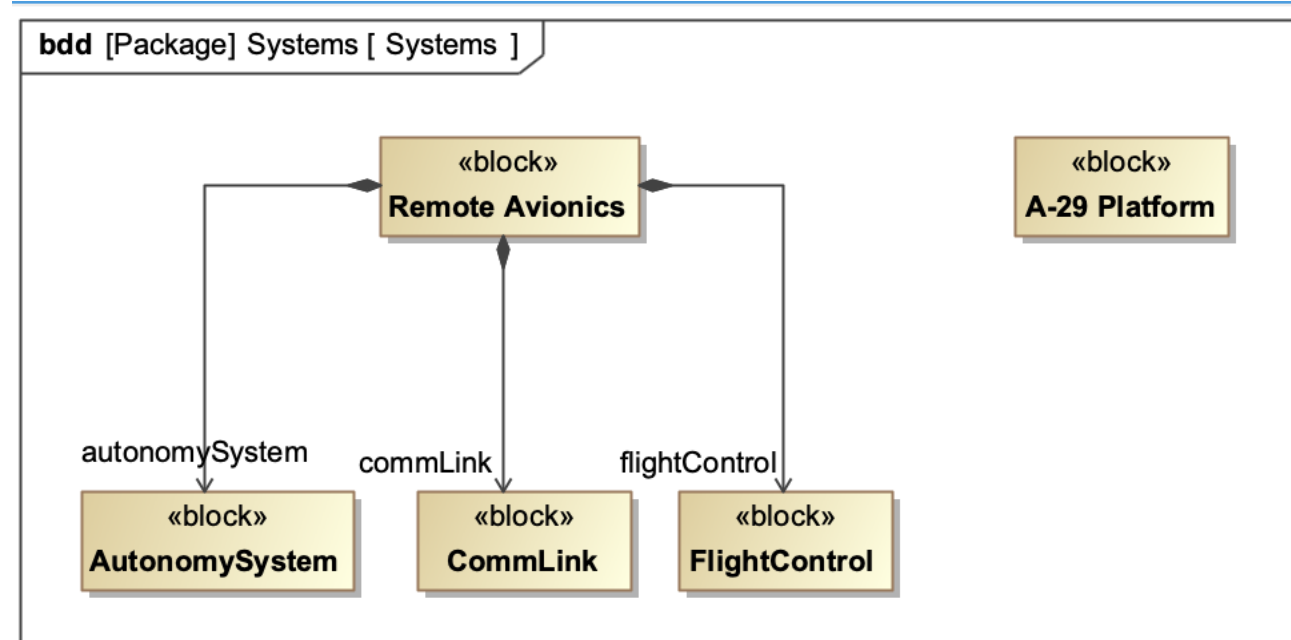
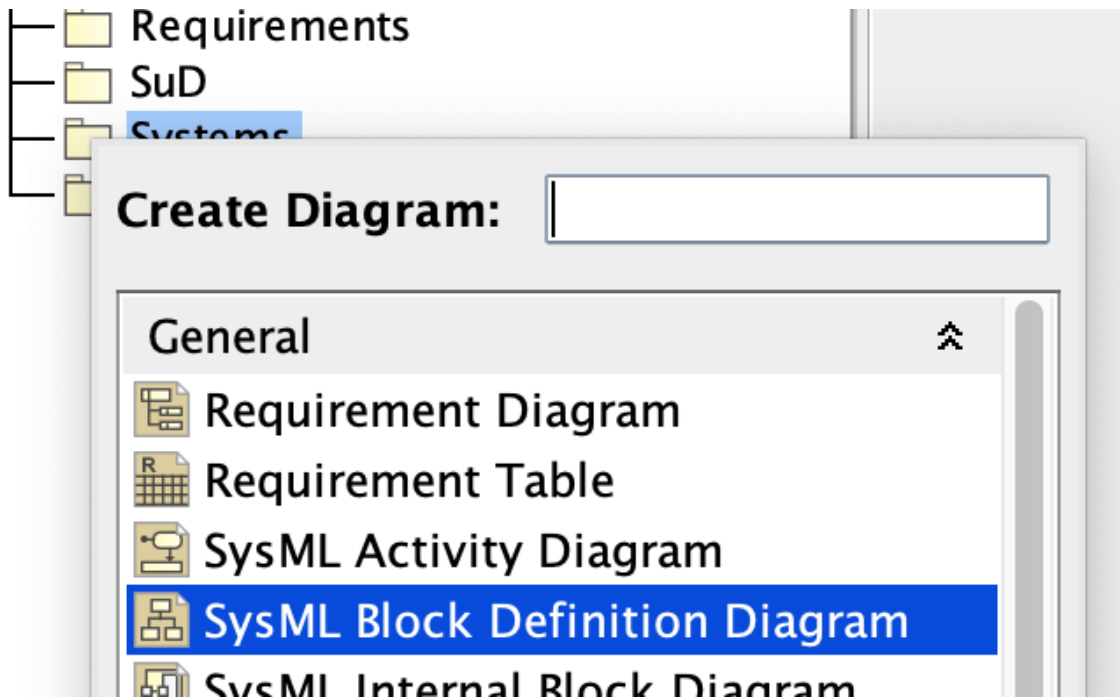
- First Create the basic packages
  - Click on model, and create two packages:
    - **Requirements** and
    - **SuD** and
    - **Systems** and
    - **TradeStudies**





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

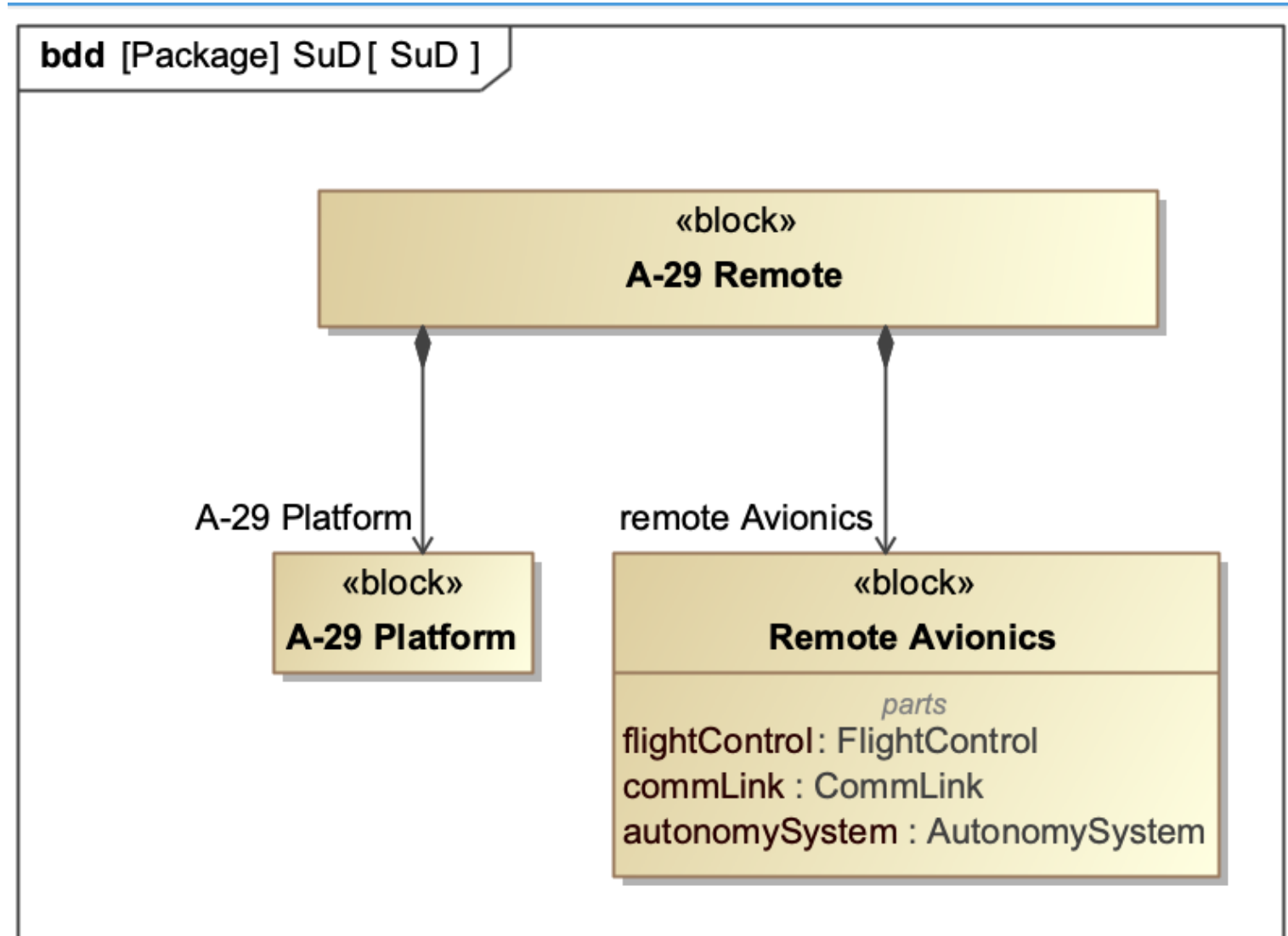
- Create the “systems” and a BDD





# 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)

The screenshot illustrates the process of creating value alternatives (roll-up patterns) in a software environment. The main window displays a diagram with a package named "bdd [Package] SuD [ SuD ]" containing a block labeled "«block»". A blue arrow points to this block, and another blue arrow points to the "Tools" menu item in the context menu.

The context menu is open, showing the following options:

- Specification
- Symbol Properties
- Element Group
- Create Diagram
- Create Relation
- Select in Containment Tree  B
- Select in Structure Tree
- Go To
- Display
- Related Elements
- Refactor
- Tools**
- Edit Compartments
- Stereotype
- Simulation

The "Tools" menu is highlighted, and a blue arrow points to the "Rollup Pattern Wizard" dialog box.

The "Rollup Pattern Wizard" dialog box is open, showing the following options:

- Apply the rollup pattern**  
Select one or more pattern Blocks to apply them recursively.
- Select Pattern Block:**  
CostRollUpPattern
- ☒ Apply Recursively
- ☐ Set Role Name
- ☒ Set Subsetted Properties
- ☒ Create Value Properties and Redefine
- OK Cancel Help

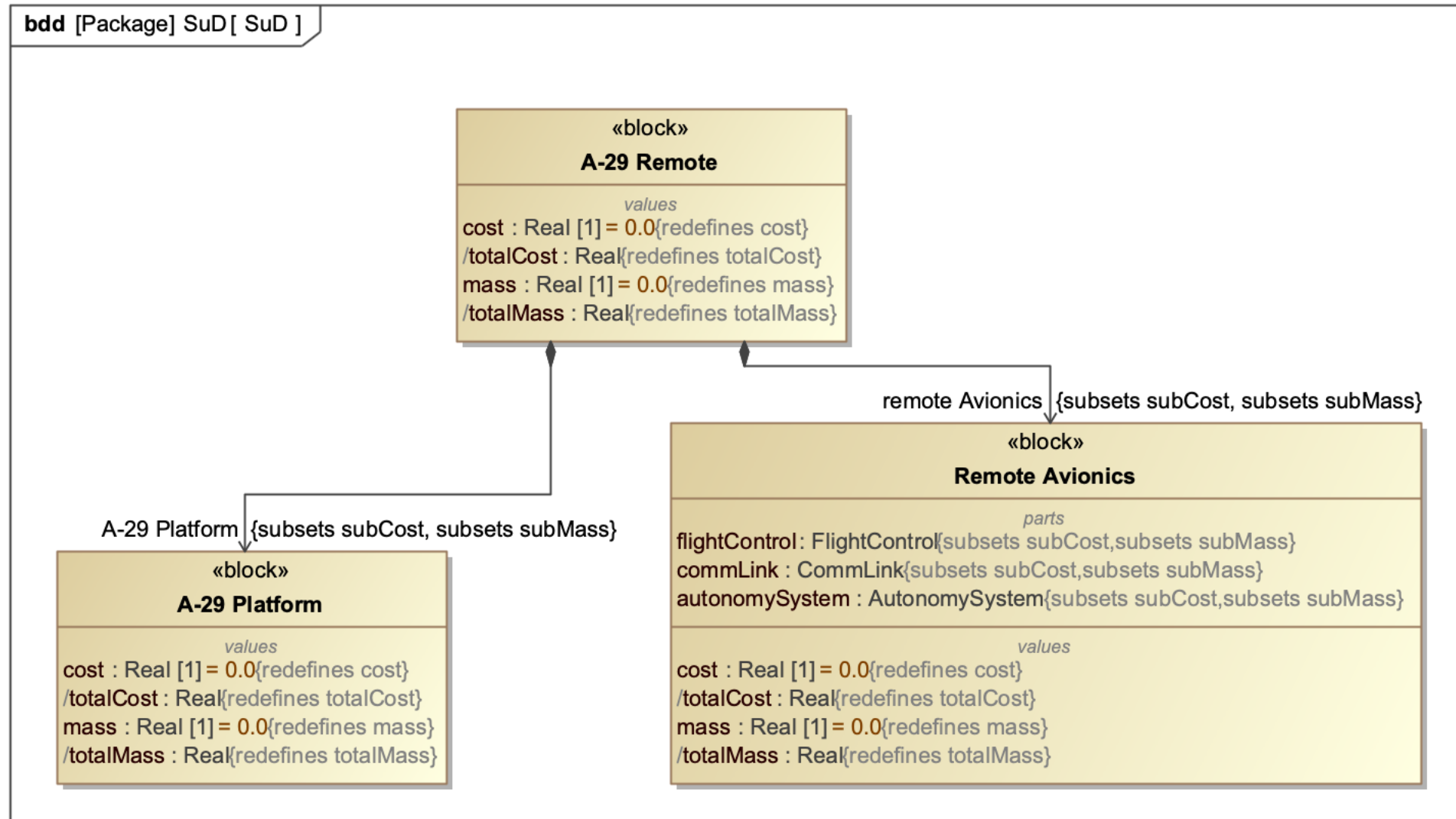
Below the wizard, the text "Create cost//mass" is displayed. A search bar labeled "Search by Name" shows 3 matches:

- CostRollUpPattern
- MassRollUpPattern
- PowerRollUpPattern

The "Apply Rollup Pattern..." option is highlighted in the context menu.

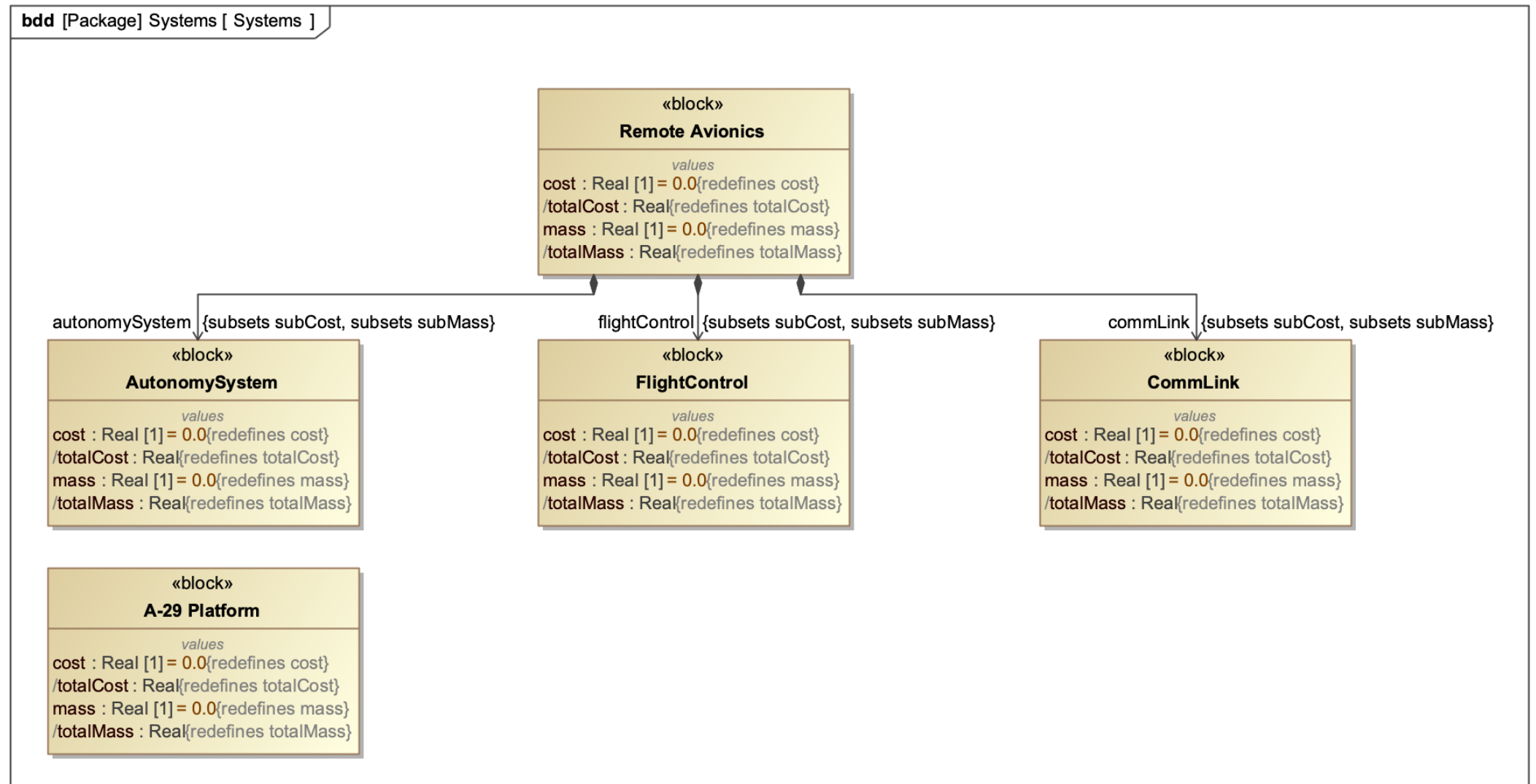


# Result of the rollup pattern



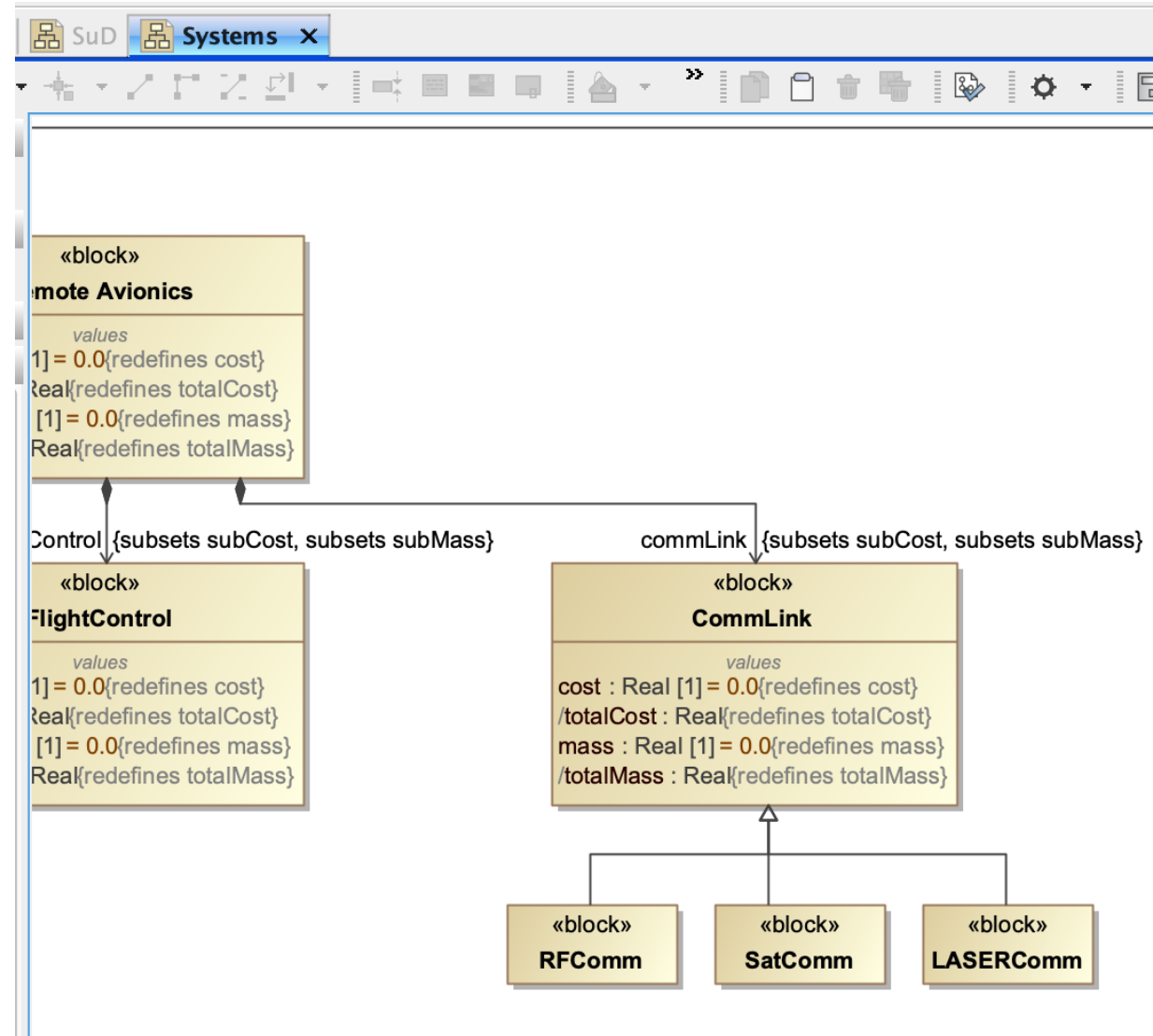


# Result of the rollup pattern





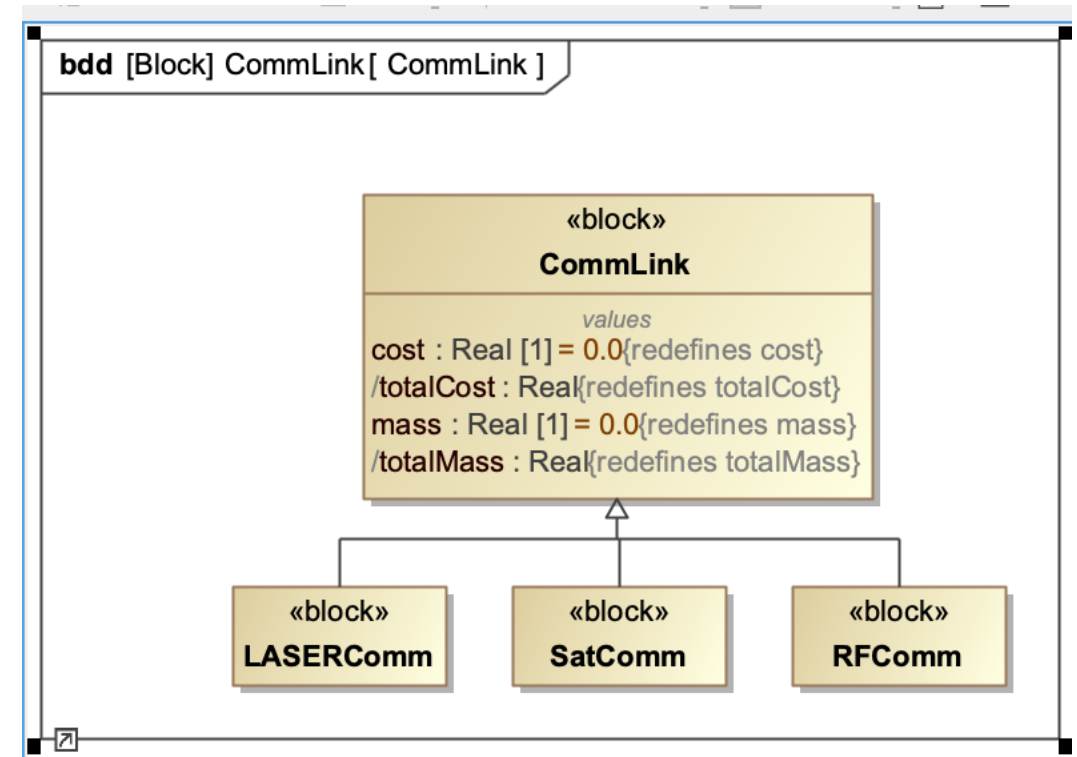
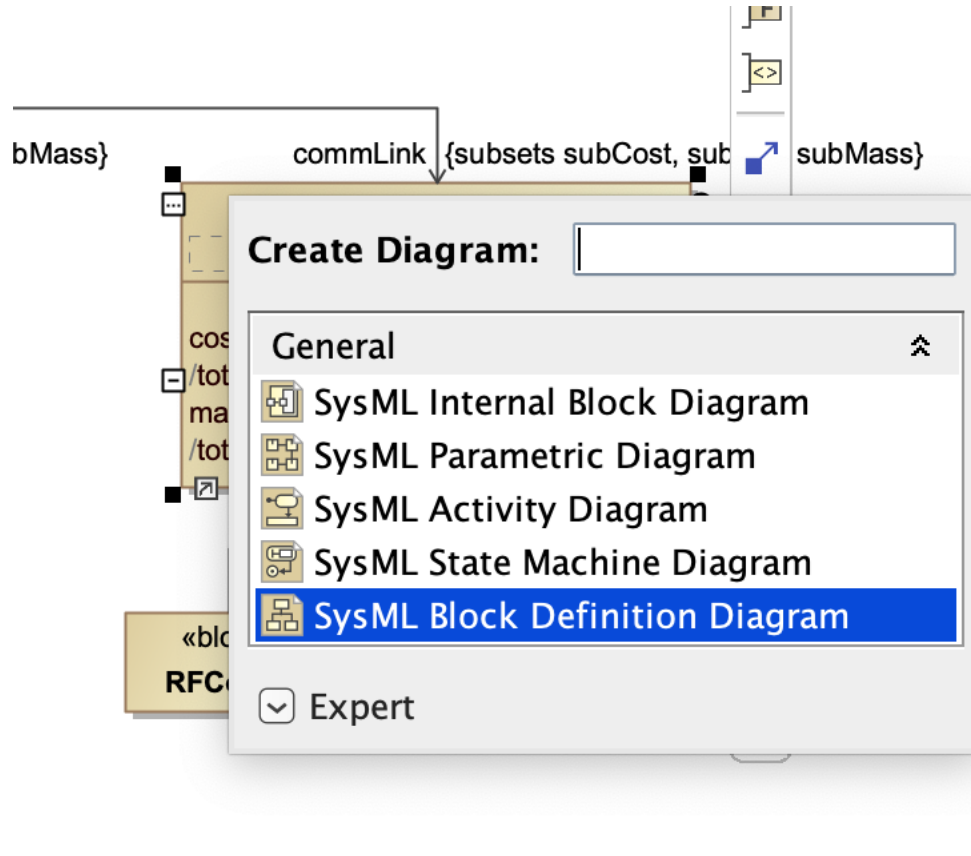
# Return into the Systems BDD and add 3 CommLink alternatives







# To easy things up create a bdd of the commlink





# Redefine all the property values (ONLY THE PV)

**Specification of Block LASERComm**

**Block attributes**  
The Attributes node contains a list of Block attributes. Create or delete attributes. Use the attribute specification button to edit properties of a specific attribute.

**Attributes**

Name	Type	Default Value	Classifier
<b>Inherited</b>			
+ cost	Real [SysML::Librari...	0	CommLink [Systems]
+ totalCost	Real [SysML::Librari...		CommLink [Systems]
+ mass	Real [SysML::Librari...	0	CommLink [Systems]
+ totalMass	Real [SysML::Librari...		CommLink [Systems]
+ sum	total [MD Customiz...		CostRollUpPattern [...]
+ sum	total [MD Customiz...		MassRollUpPattern ...
<b>Inherited Association End</b>			
+ subCost	CostRollUpPattern [...]		CostRollUpPattern [...]
+ subMass	MassRollUpPattern ...		MassRollUpPattern ...

Up Down Create Clone **Redefine** Delete

Close Back Forward Help

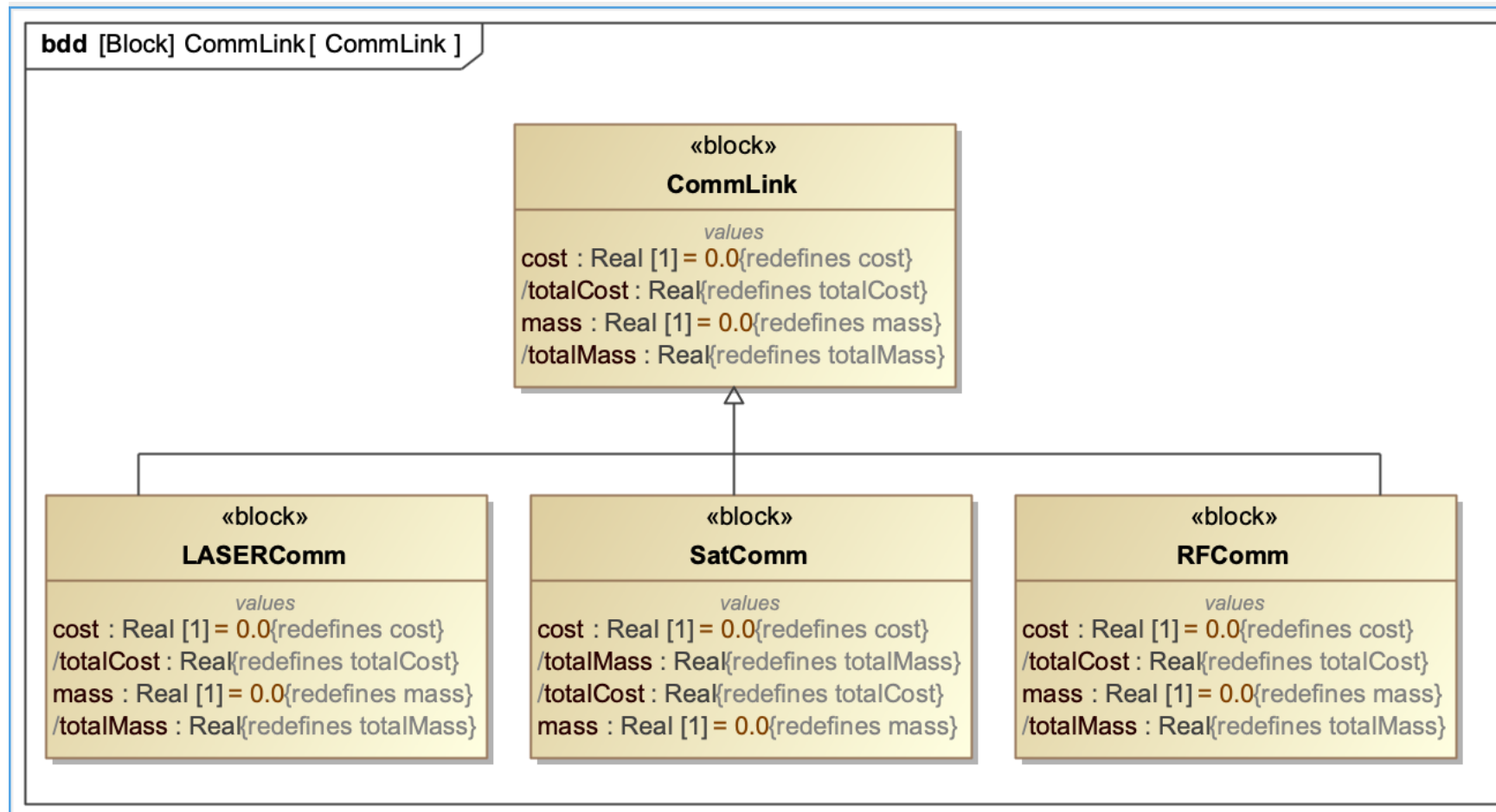
CommLink [ CommLink ]

```
graph TD
    subgraph "«block» CommLink"
        values["cost : Real [1] = 0.0(redefines cost)  
/totalCost : Real(redefines totalCost)  
mass : Real [1] = 0.0(redefines mass)  
/totalMass : Real(redefines totalMass)"]
    end
    subgraph "«block» LASERComm"
    end
    subgraph "«block» SatComm"
    end
    subgraph "«block» RFComm"
    end
    LASERComm --> CommLink
    SatComm --> CommLink
    RFComm --> CommLink
```

Do the same for all three variations

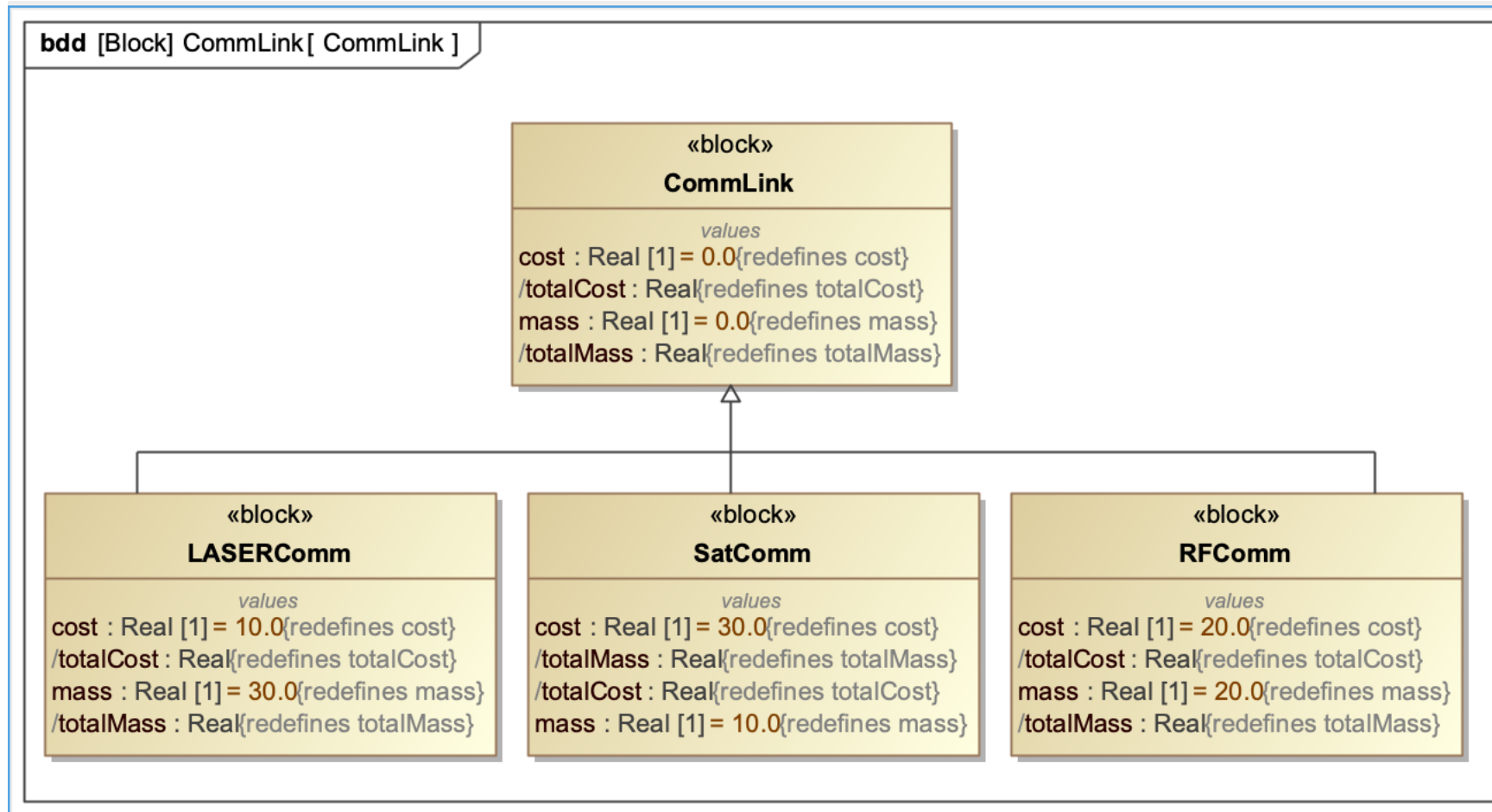


# Will result in a thing like this





# 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									

**Create Diagram:** ins

**Other Diagrams**

- Instance Table

**Model**

- Requirements
- SuD
- Systems
  - Relations
  - Instance Table
  - Systems
    - A-29 Platform
    - AutonomySystem
    - CommLink
    - FlightControl
    - LASERComm
    - Remote Avionics
    - RFCComm
    - SatComm

**Criteria**

Classifier: Drag elements from the Model Browse ... Scope (optional): Drag elements from the Model Browser {}xy ... Filter: Y



# Connect the reference to the file

The screenshot shows a software interface with a file explorer on the left and a table on the right. A red arrow points from the 'Systems' folder in the file explorer to the 'tradeAutonomy.xlsx' file in the table.

**File Explorer (Left):**

- Model
  - Requirements
  - SuD
  - Systems
    - Relations
    - Instance Table
    - Systems
    - A-29 Platform
    - AutonomySystem
    - CommLink
    - FlightControl
    - LASERComm
    - Remote Avionics
    - RFCComm
    - SatComm
  - TradeStudies

**Table (Right):**

#	Name	cost : Real	mass : Real
	[CE-MBSE][EMB][...s - Requisitos.pptx]		
	[CE-MBSE][EMB][...sos de Uso (1).pptx]		
	[CE-MBSE][EMB][...Casos de Uso.pdf]		
	[CE-MBSE][EMB][...Casos de Uso.pptx]		
	[CE-MBSE][EMB][...es e Requisitos.pdf]		
	[CE-MBSE][EMB][...s e Requisitos.pptx]		
	[CE-MBSE][EMB][...- SE e Modelos.pdf]		
	[CE-MBSE][EMB][...SE e Modelos.pptx]		
	[CE-MBSE][EMB][...ia Simplificada.pdf]		
	[CE-MBSE][EMB][...a Simplificada.pptx]		
	[CE-MBSE][EMB][...Ps e Interfaces.pdf]		
	[CE-MBSE][EMB][...s e Interfaces.pptx]		
	[CE-MBSE][EMB][...RADE STUDY.pptx]		
	tradeAutonomy.xlsx		

The diagram shows a file structure with a callout box. A blue arrow points from the 'Systems' folder to the 'Add Hyperlink Create Attached File' text.

**File Structure:**

- SuD
  - Systems
    - Relations
    - Instance Table
    - Systems
    - A-29 Platform

**Callout Box:**

Add Hyperlink  
Create Attached File

The screenshot shows a software interface with a file explorer on the left and a table on the right. A red arrow points from the 'tradeAutonomy.xlsx' file in the file explorer to the 'tradeAutonomy.xlsx' file in the table.

**File Explorer (Left):**

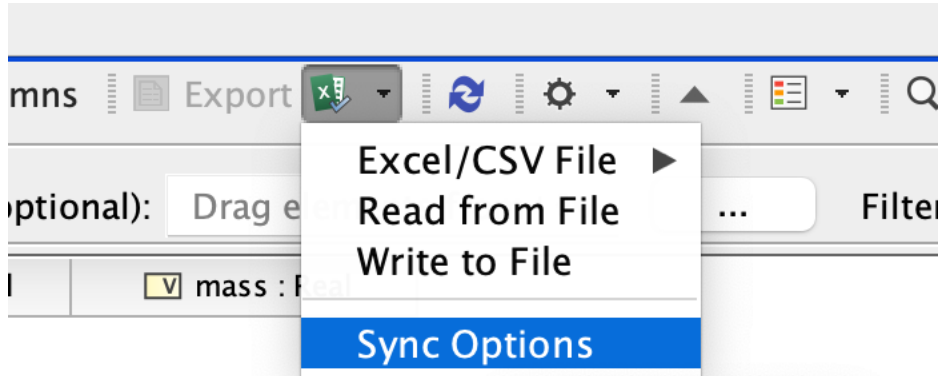
- Model
  - Requirements
  - SuD
  - Systems
    - Relations
    - Instance Table
    - Systems
    - A-29 Platform
    - AutonomySystem
    - CommLink
    - FlightControl
    - LASERComm
    - Remote Avionics
    - RFCComm
    - SatComm
  - TradeStudies

**Table (Right):**

#	Name	cost : Real	mass : Real
	[CE-MBSE][EMB][...s - Requisitos.pptx]		
	[CE-MBSE][EMB][...sos de Uso (1).pptx]		
	[CE-MBSE][EMB][...Casos de Uso.pdf]		
	[CE-MBSE][EMB][...Casos de Uso.pptx]		
	[CE-MBSE][EMB][...es e Requisitos.pdf]		
	[CE-MBSE][EMB][...s e Requisitos.pptx]		
	[CE-MBSE][EMB][...- SE e Modelos.pdf]		
	[CE-MBSE][EMB][...SE e Modelos.pptx]		
	[CE-MBSE][EMB][...ia Simplificada.pdf]		
	[CE-MBSE][EMB][...a Simplificada.pptx]		
	[CE-MBSE][EMB][...Ps e Interfaces.pdf]		
	[CE-MBSE][EMB][...s e Interfaces.pptx]		
	[CE-MBSE][EMB][...RADE STUDY.pptx]		
	tradeAutonomy.xlsx		



# Check the sync



Excel/CSV Sync Options

**Excel/CSV Sync Options**

**Set Excel/CSV sync options**

This tool allows you to specify settings before syncing content between an Excel/CSV file and a table. Select the Excel or CSV file to sync with the table. To define syncing or mapping options, click the Details button.

Select Excel/CSV file

☐ From file system

☒ From model

Excel/CSV File: tradeAutonomy.xlsx

Sync Options

If rows in the file are deleted: Mark as obsolete

Mapping Options

Sheet: Sheet1

First Cell: A1

CSV Delimiter: ,

Identification Property: Default

☒ First row contains headings

Table Columns

Name
cost : Real
mass : Real

Excel/CSV Columns

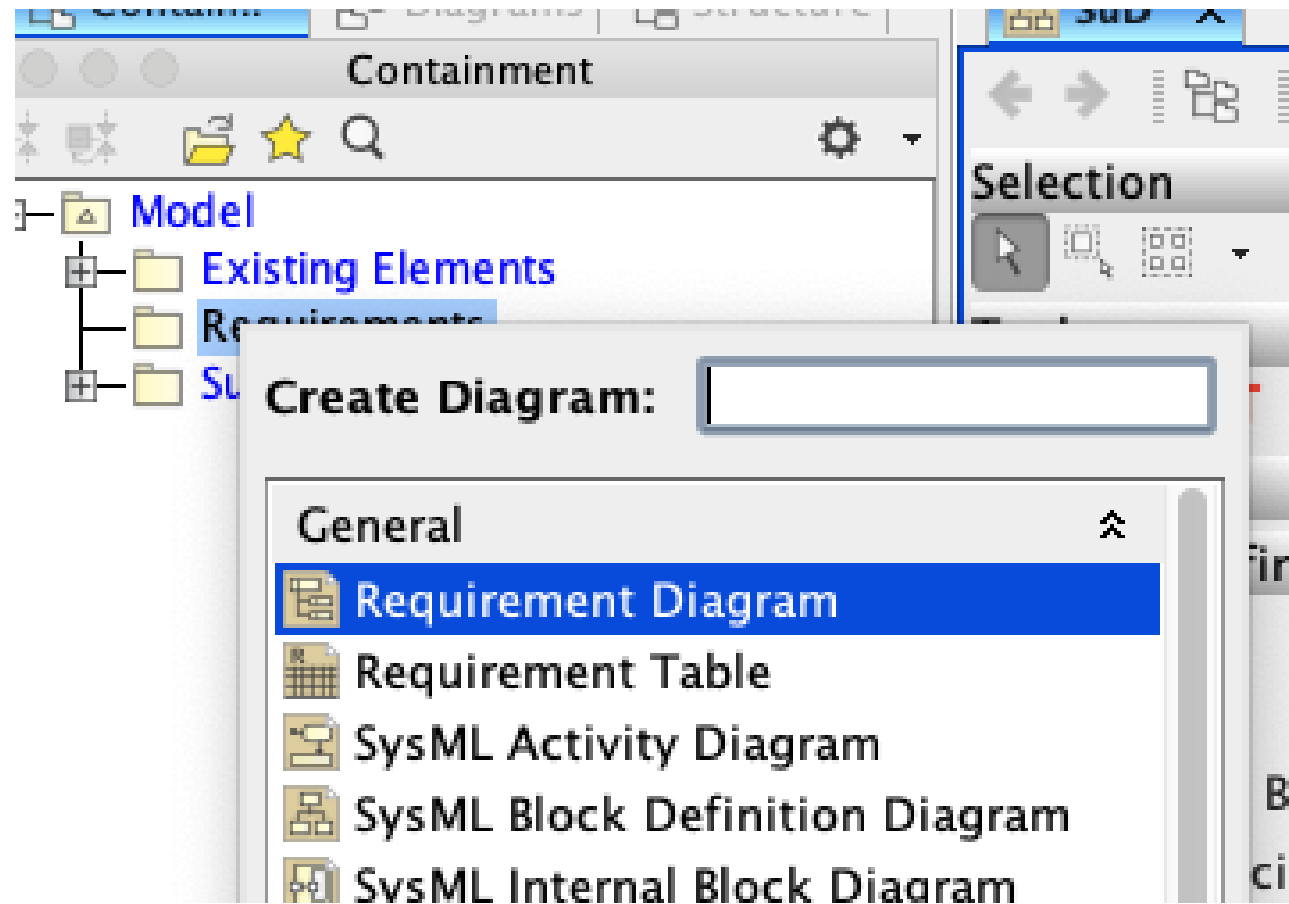
name (A1)
cost (B1)
mass (C1)

Details OK Cancel Help



### 3) Requirements

- Create a requirement diagram

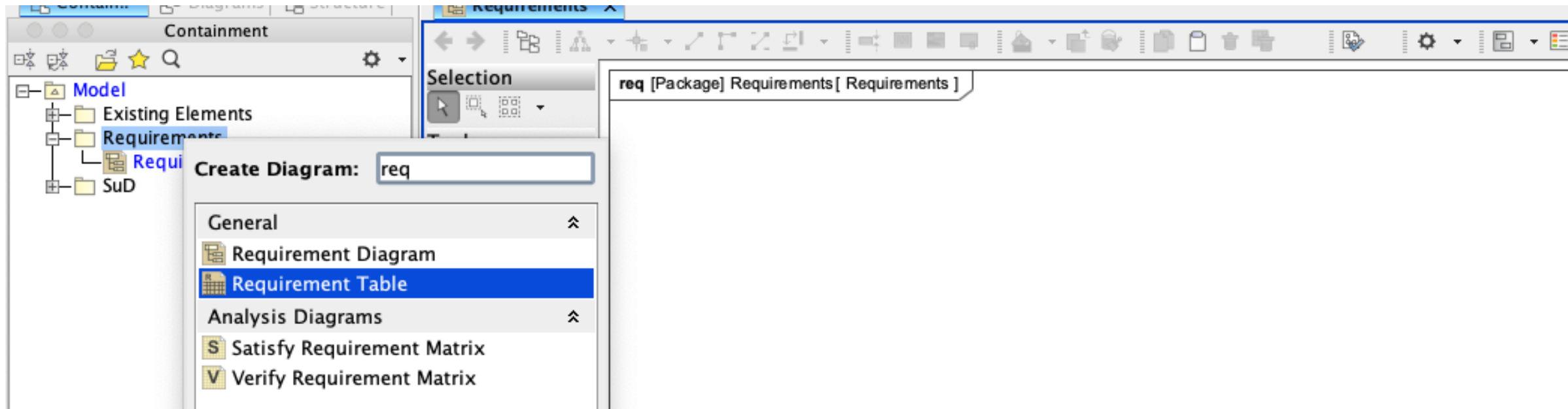






### 3) Requirements

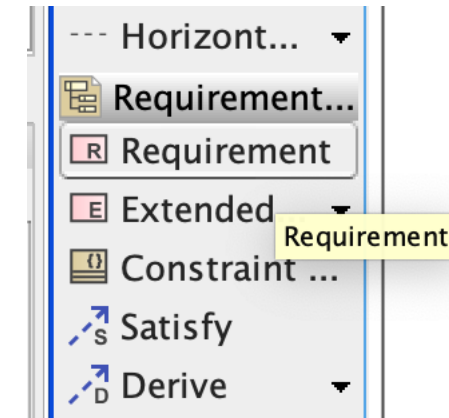
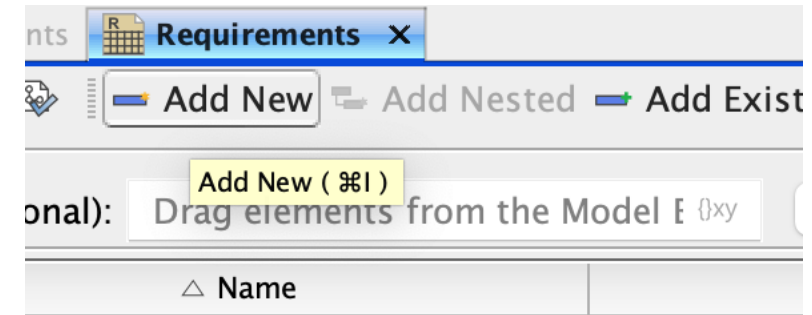
- Create a Requirement Table





# 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.

Requirements		
Requirements		
Criteria		
Scope (optional): Drag elements from the Model E {xy} ... Fil		
#	Name	△ Text
1	R 3 Aircraft mass	The Remote Aircraft mass shall <u>not exceed 5400 kg</u> .
2	R 2 Overall Cost	The Remote Aircraft cost shall be <u>less than 100 M\$</u> .
3	R 1 RemoteSystem Cost	The RemoteSystem cost shall be <u>less than 4M\$</u> .



### 3) Requirements

- Check the requirements

req [Package] Requirements[ Requirements ]

«requirement»

**Aircraft mass**

Id = "3"  
Text = "The Remote Aircraft  
mass shall not exceed 5400  
kg."  
"

«requirement»

**Overall Cost**




Id = "2"  
Text = "The Remote Aircraft  
cost shall be less than 100  
M\$."  
"

«requirement»

**RemoteSystem Cost**

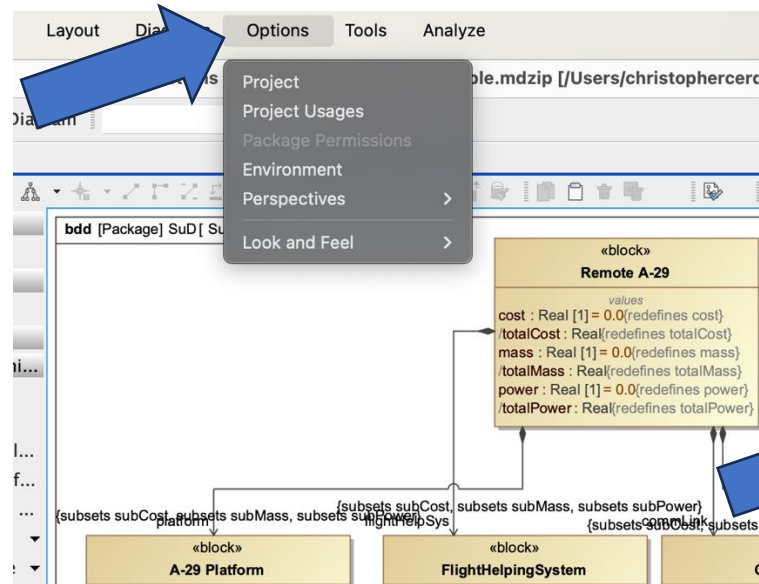
Id = "1"  
Text = "The RemoteSystem  
cost shall be less than 4M\$.  
"

Table [Package] Requirements[ Requirements ]

#	Name	△ Text
1	 3 Aircraft mass	The Remote Aircraft mass shall not exceed 5400 kg.
2	 2 Overall Cost	The Remote Aircraft cost shall be less than 100 M\$.
3	 1 RemoteSystem Cost	The RemoteSystem cost shall be less than 4M\$.



# Enable requirement natural language processing



The screenshot shows the 'Project Options' dialog box, which is used to specify general project properties. The 'General' tab is selected, and the 'Use Requirement Terms Glossary' option is checked. A blue arrow points to this option. The dialog box also shows a list of options on the left, including 'General', 'Browser', 'Dependency Checker', 'Diagrams', 'Element References', 'Legend Adorning', 'Numbering', 'ReqIF', 'Requirements', 'Simulation', 'Suspect Links', 'SysML', 'Validation', 'Diagram Info', 'Symbol styles', and 'Default model properties'. On the right, the 'General' section is expanded, showing various settings. A red box highlights the 'Requirement Terms Glossary' section, which contains three requirement terms: 'Aircraft mass', 'Overall Cost', and 'RemoteSystem Cost'. Each term has an ID, a text description, and a condition (e.g., 'not exceed 5400 kg.', 'less than 100 M\$', 'less than 4M\$').

**Project Options**

Specify general project properties

Specify the validation, project dependency checker options and other general project-specific options.

Type here to filter options

**General**

- ☒ Browser
- ☒ Dependency Checker
- ☒ Diagrams
- ☒ Element References
- ☒ General
- ☒ Legend Adorning
- ☒ Numbering
- ☒ ReqIF
- ☒ Requirements
- ☒ Simulation
- ☒ Suspect Links
- ☒ SysML
- ☒ Validation
- ☐ Diagram Info
- ☐ Symbol styles
- ☐ Default model properties

**General**

Path to Used Project

Enable Dot Notation for Associations

Change Ownership of Non-Navigable Asso...

Qualified Name Display Style

Synchronize Parameters and Arguments

Layout Template Creation Mode

Layout use case scenario activity diagram

Decimal Places

Eclipse UML2 XMI Output Location

Show Excel/CSV Import Migration Message

**Use Requirement Terms Glossary**

Use Glossary

Tooltips Style

Reset to Defaults

OK Cancel Help

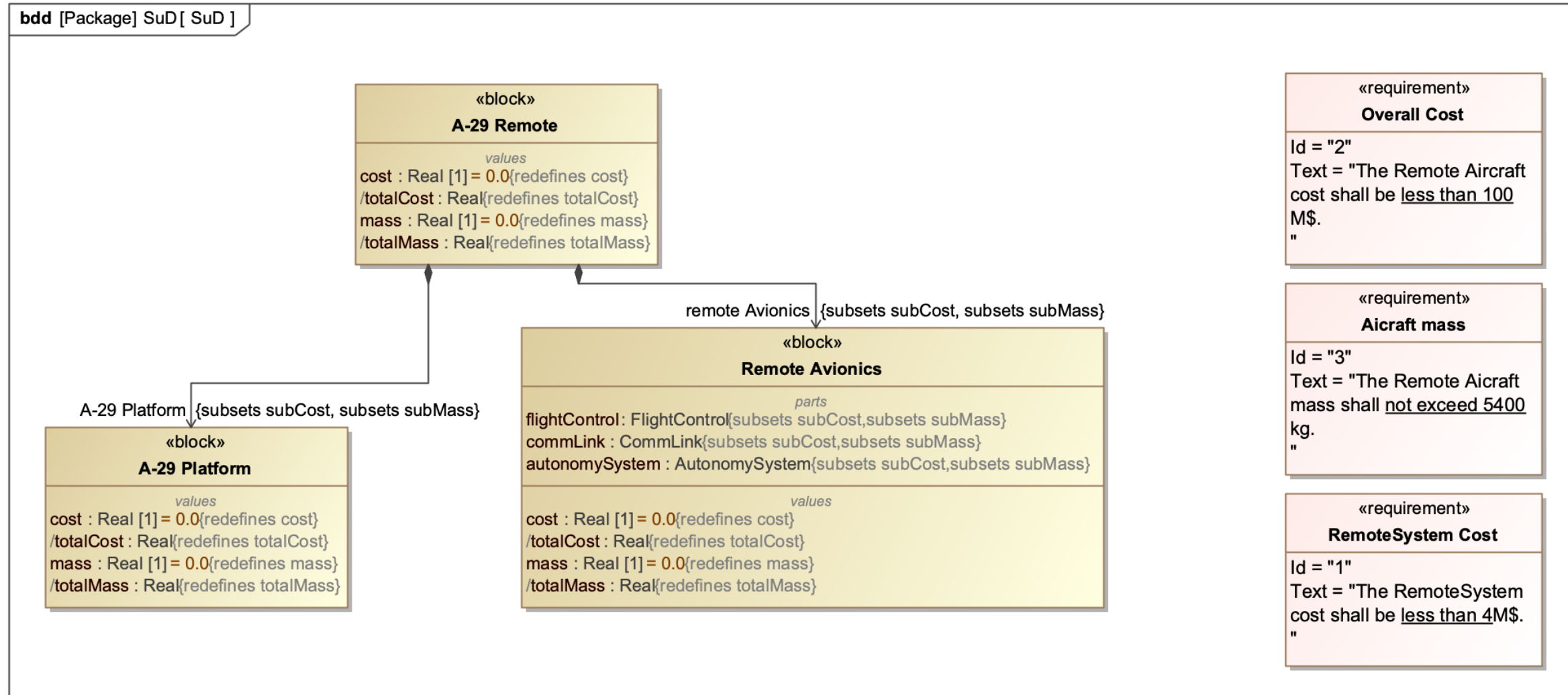
**Requirement Terms Glossary**

Set to true to use the Requirement Term glossary functionality which marks condition phrases (e.g. less than, at least) in text.

Requirement	Id	Text	Condition
«requirement» Aircraft mass	Id = "3"	Text = "The Remote Aircraft mass shall not exceed 5400 kg."	not exceed 5400 kg.
«requirement» Overall Cost	Id = "2"	Text = "The Remote Aircraft cost shall be less than 100 M\$."	less than 100 M\$.
«requirement» RemoteSystem Cost	Id = "1"	Text = "The RemoteSystem cost shall be less than 4M\$."	less than 4M\$.



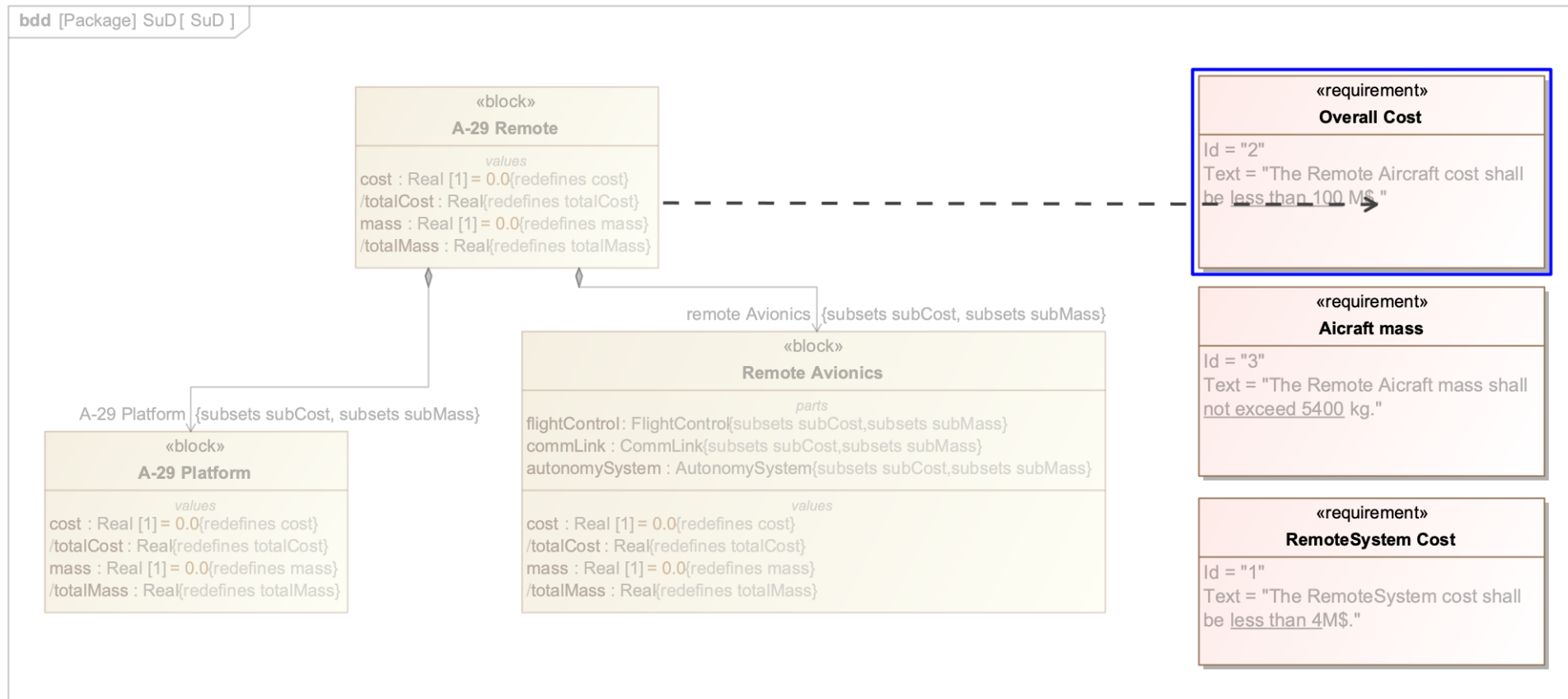
# Connect the requirement w/ the architecture





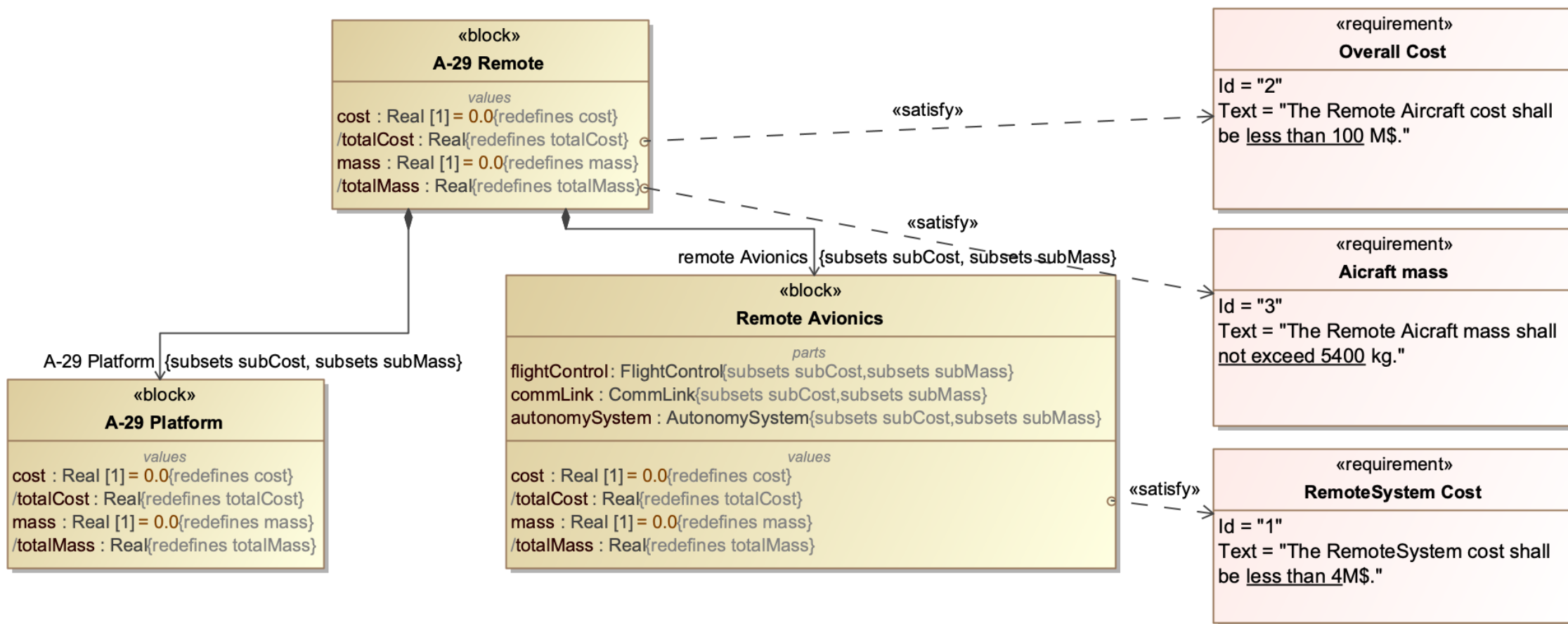
# Relate total<values> to the requirements

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





bdd [Package] SuD [ SuD ]

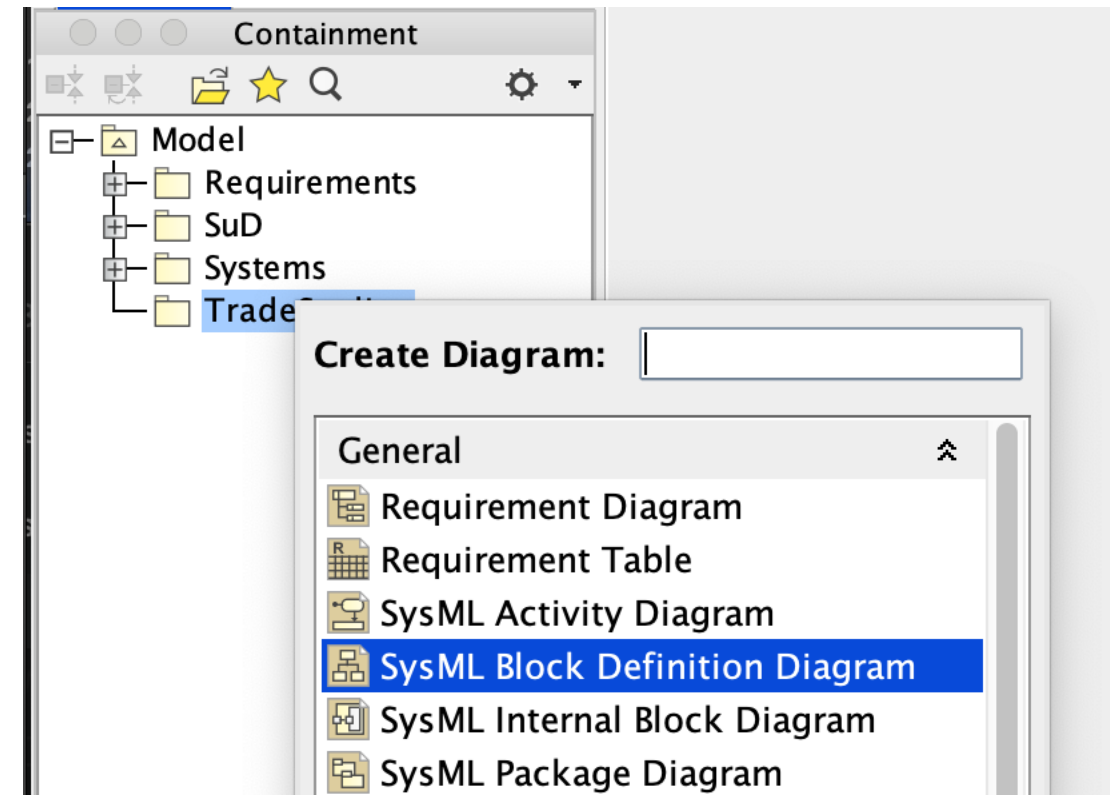






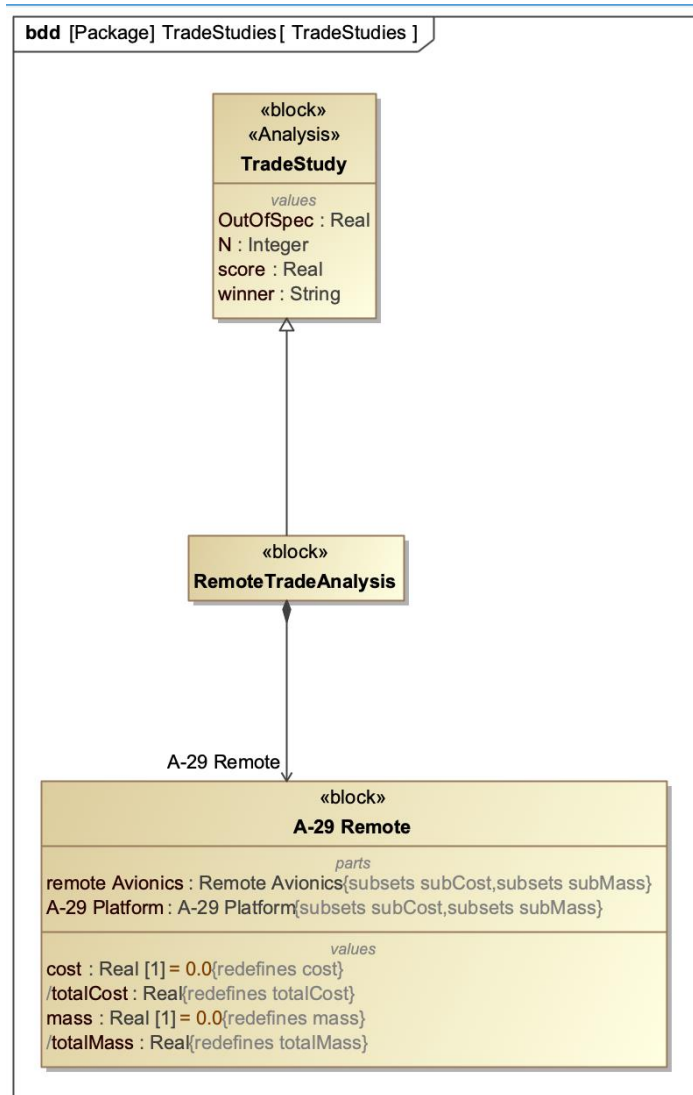
## 4) Start the trade study analysis

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





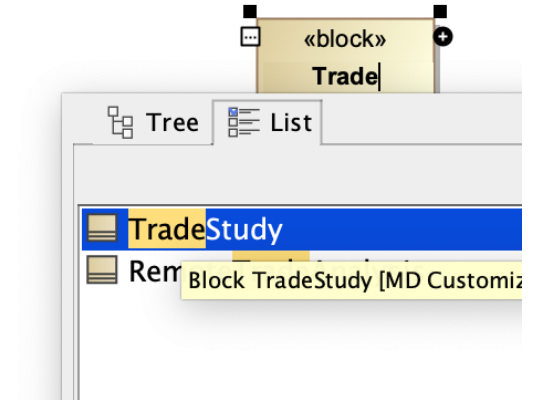
# Create the trade study



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

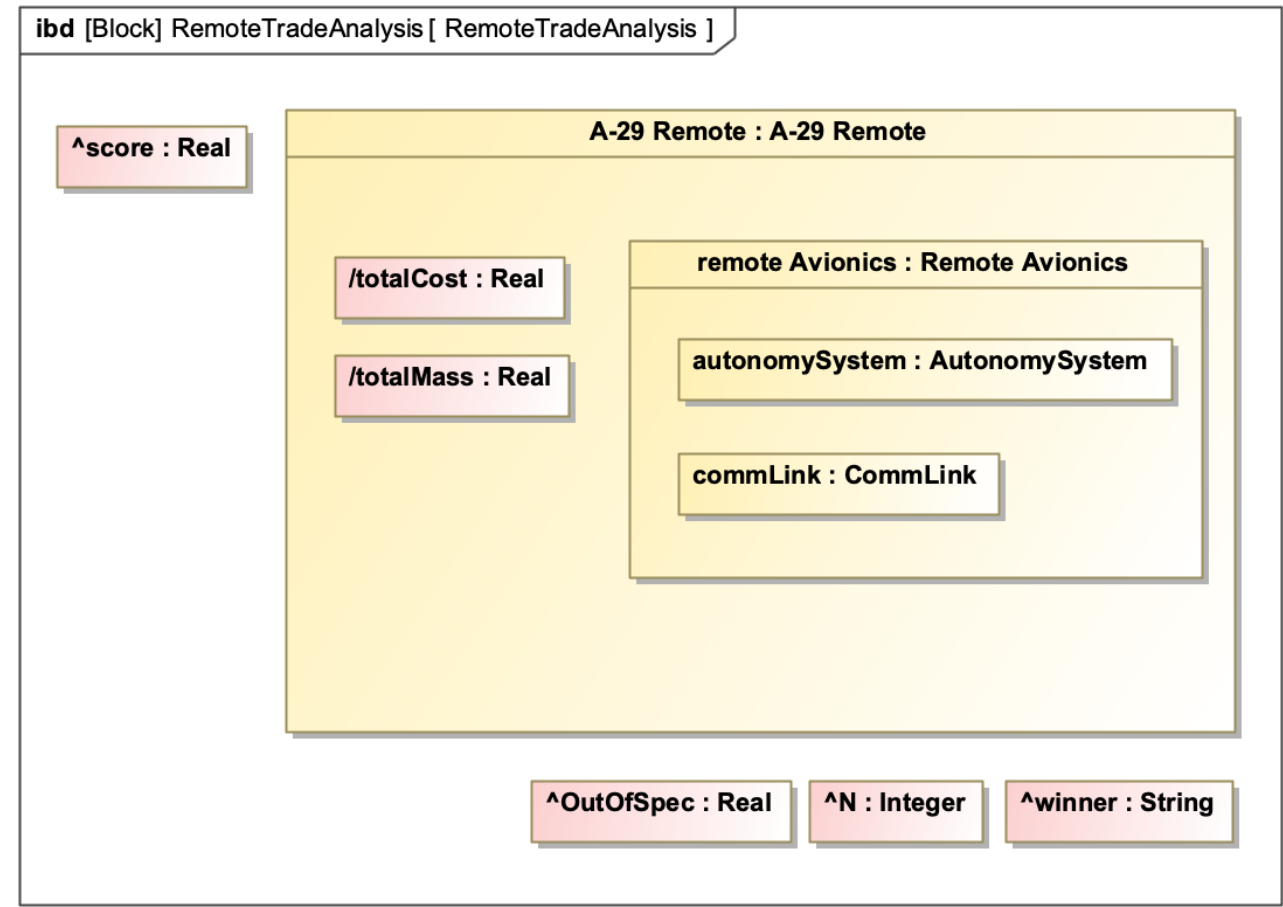
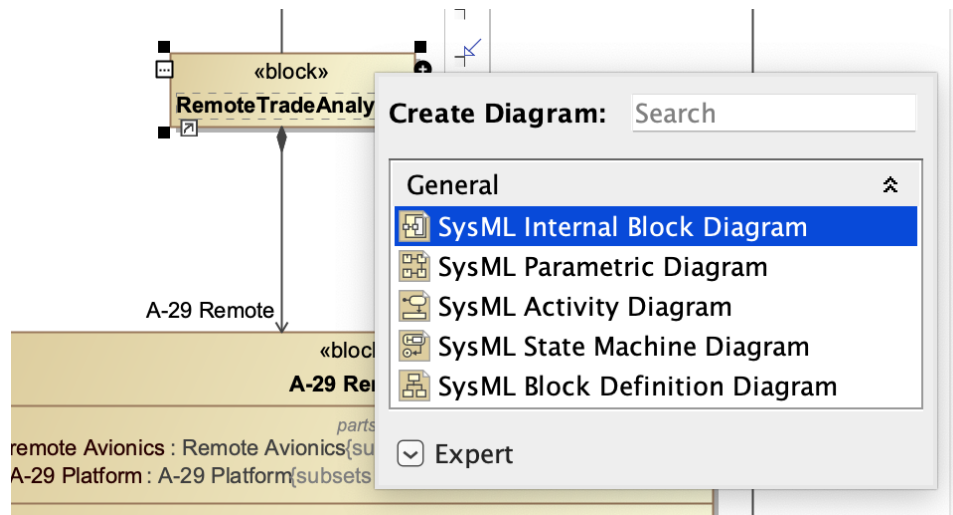
2) Create the analysis context

1) Drag the the system that will be trade



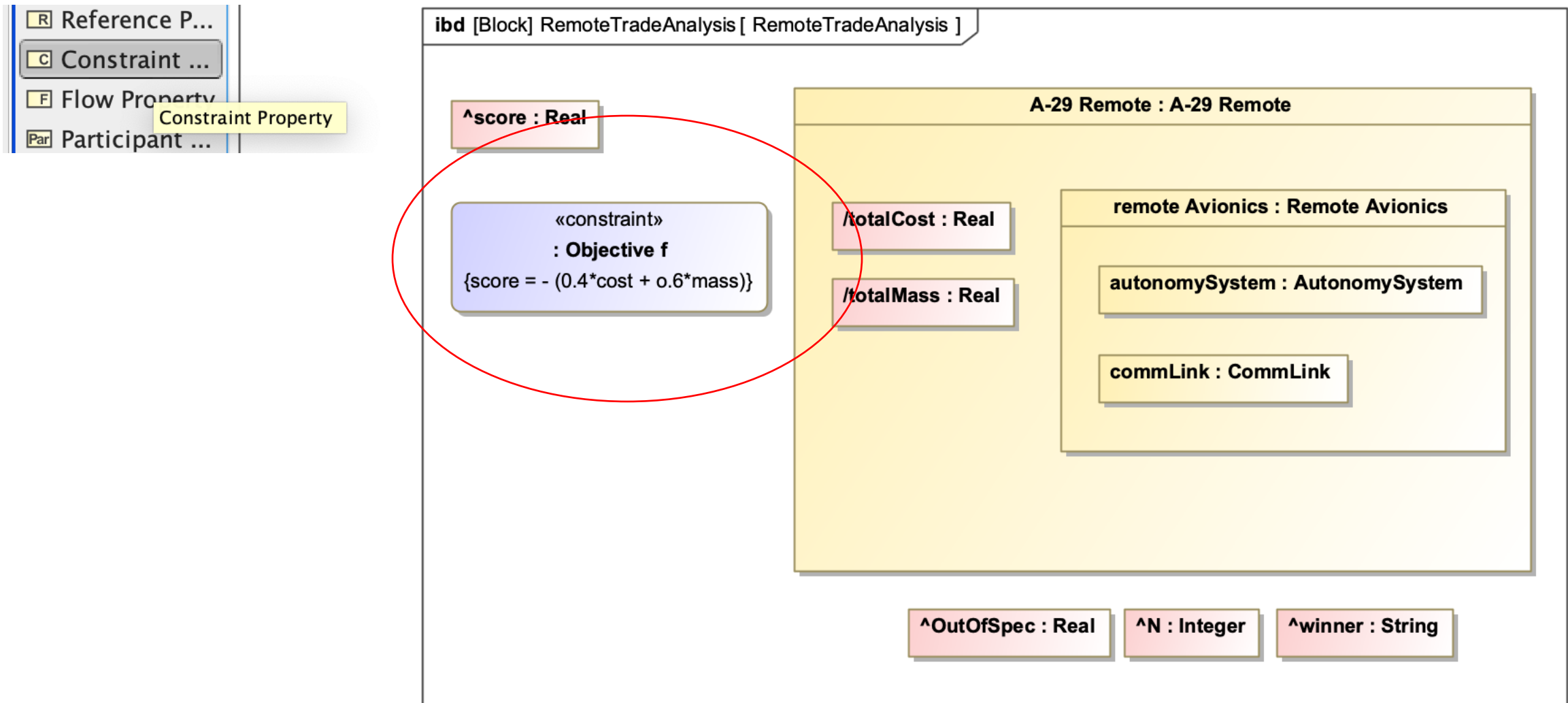


# Results of creating the IBD of the trade analysis



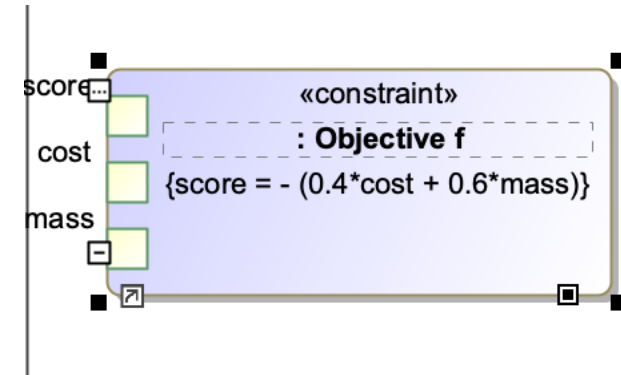
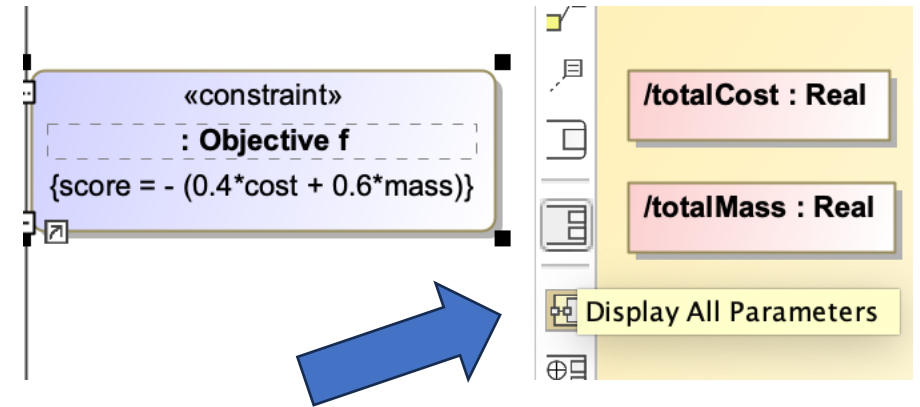
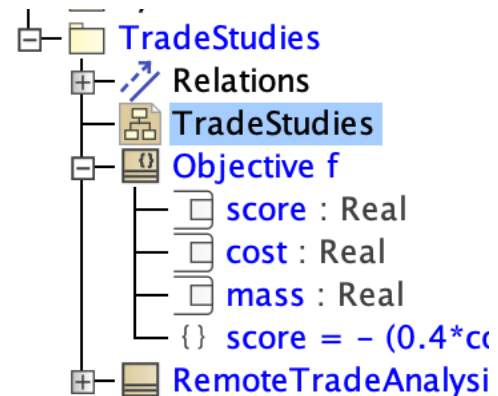
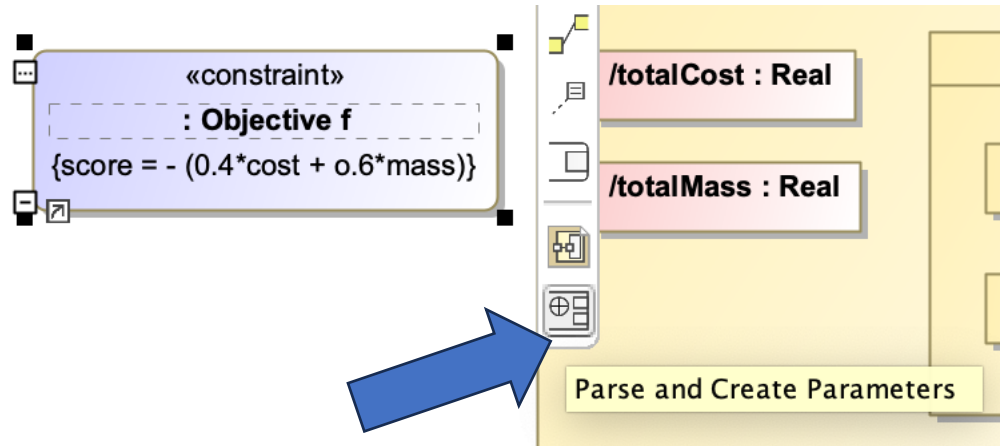


# Create a constraint of the object function



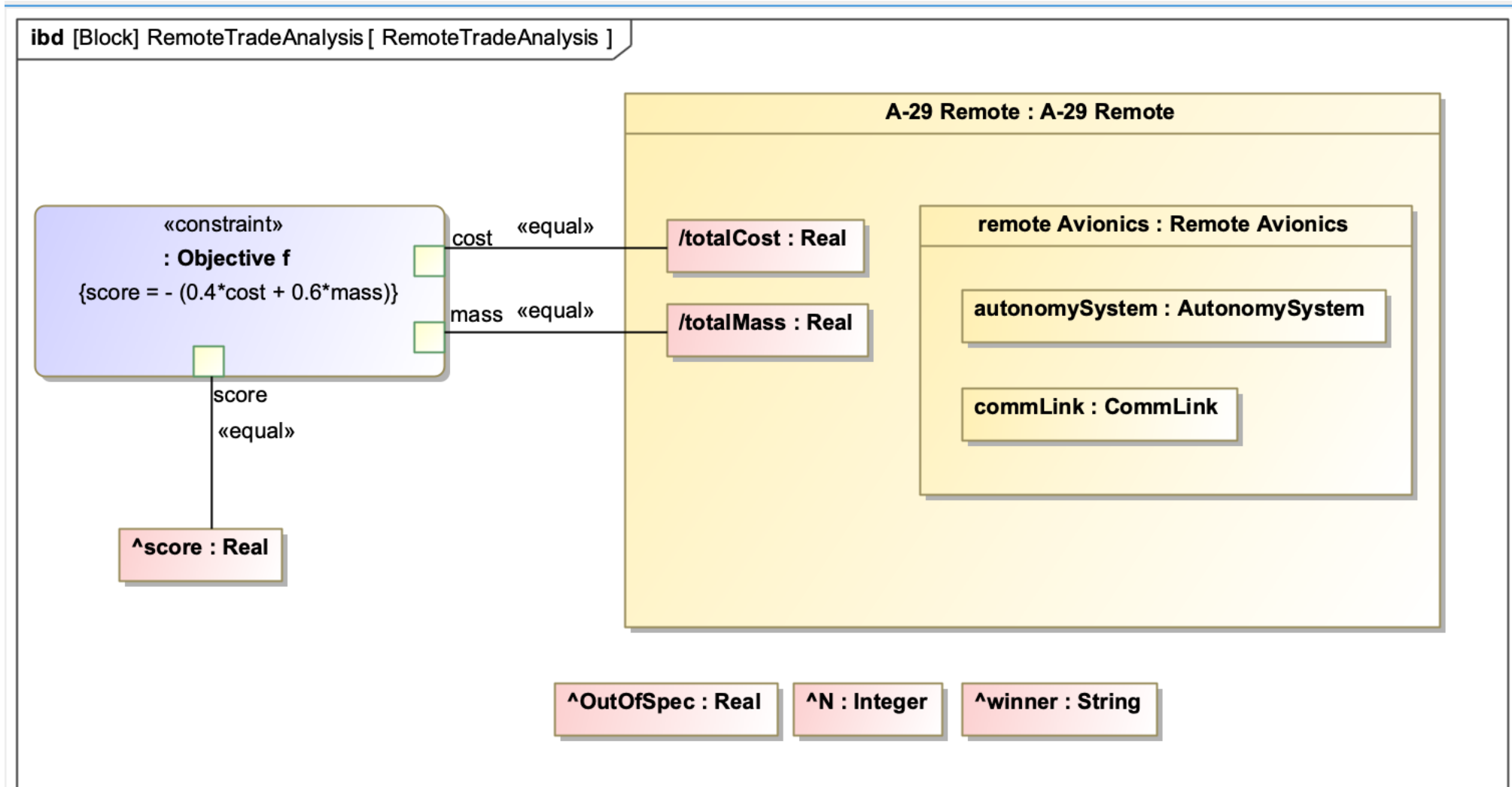


# Show the parameter ports





# Interconnect w/ the architecture elements



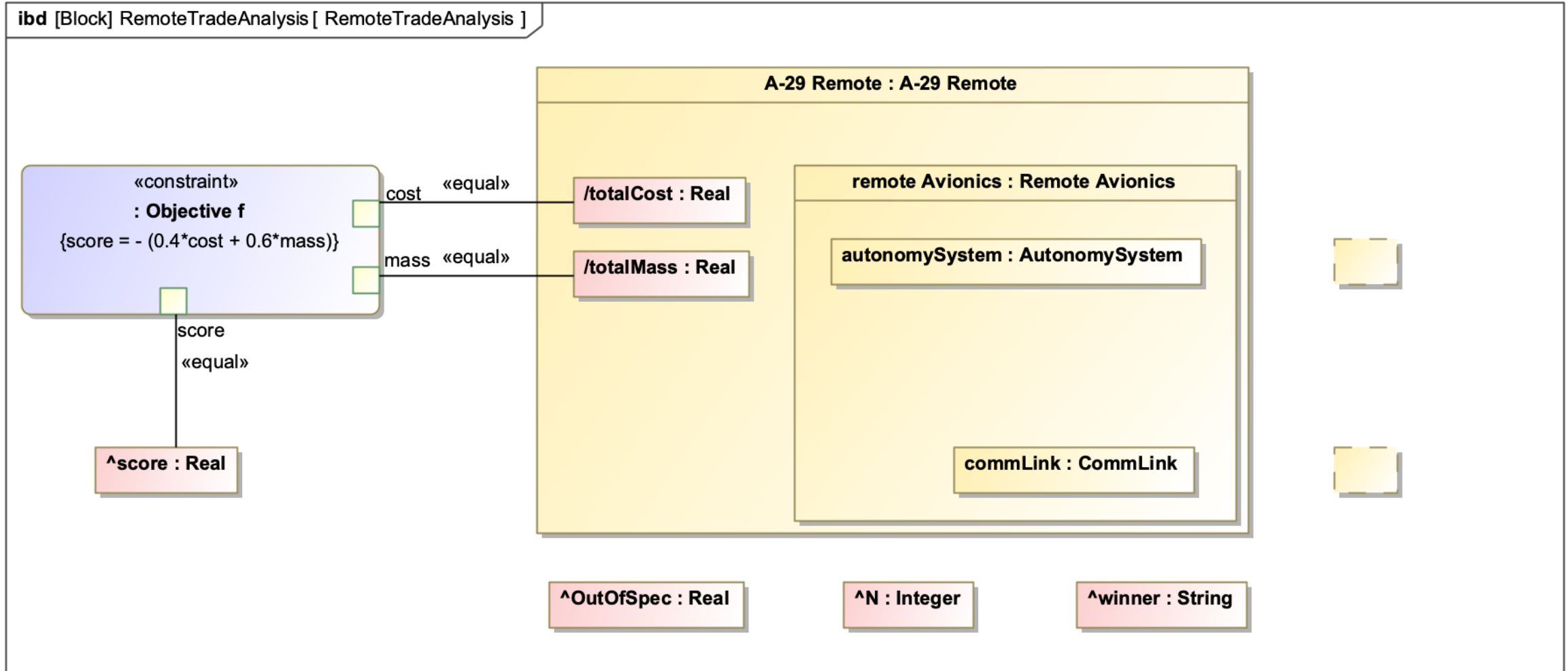
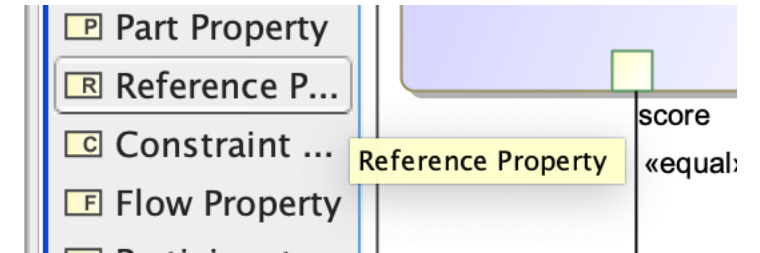


## 5) Setting up the alternatives

- Via the possible specializations
- Via a excel file



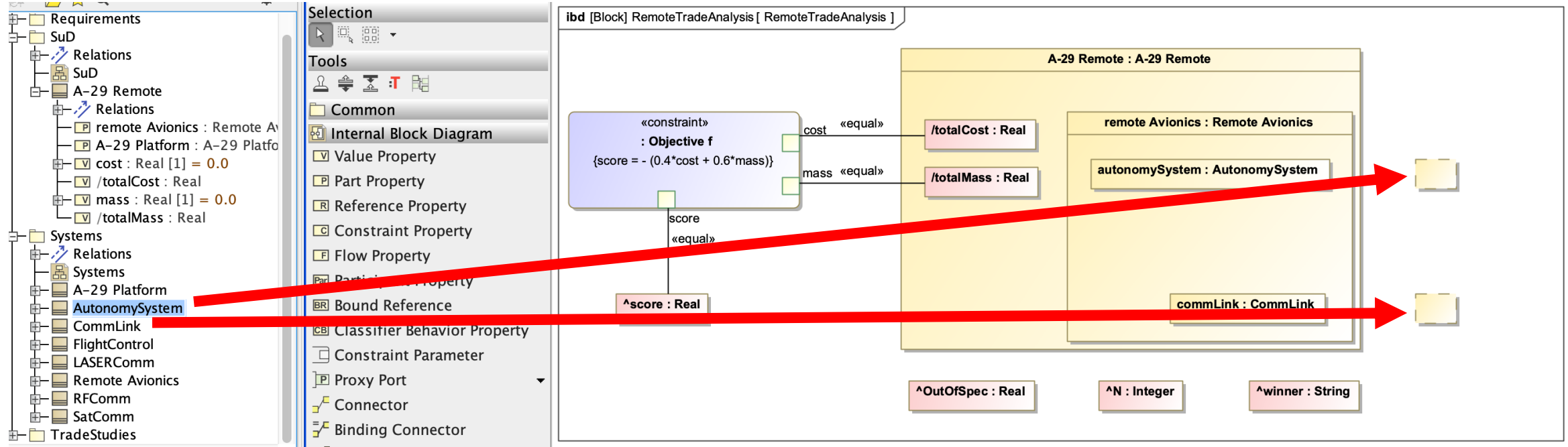
# Create source references..





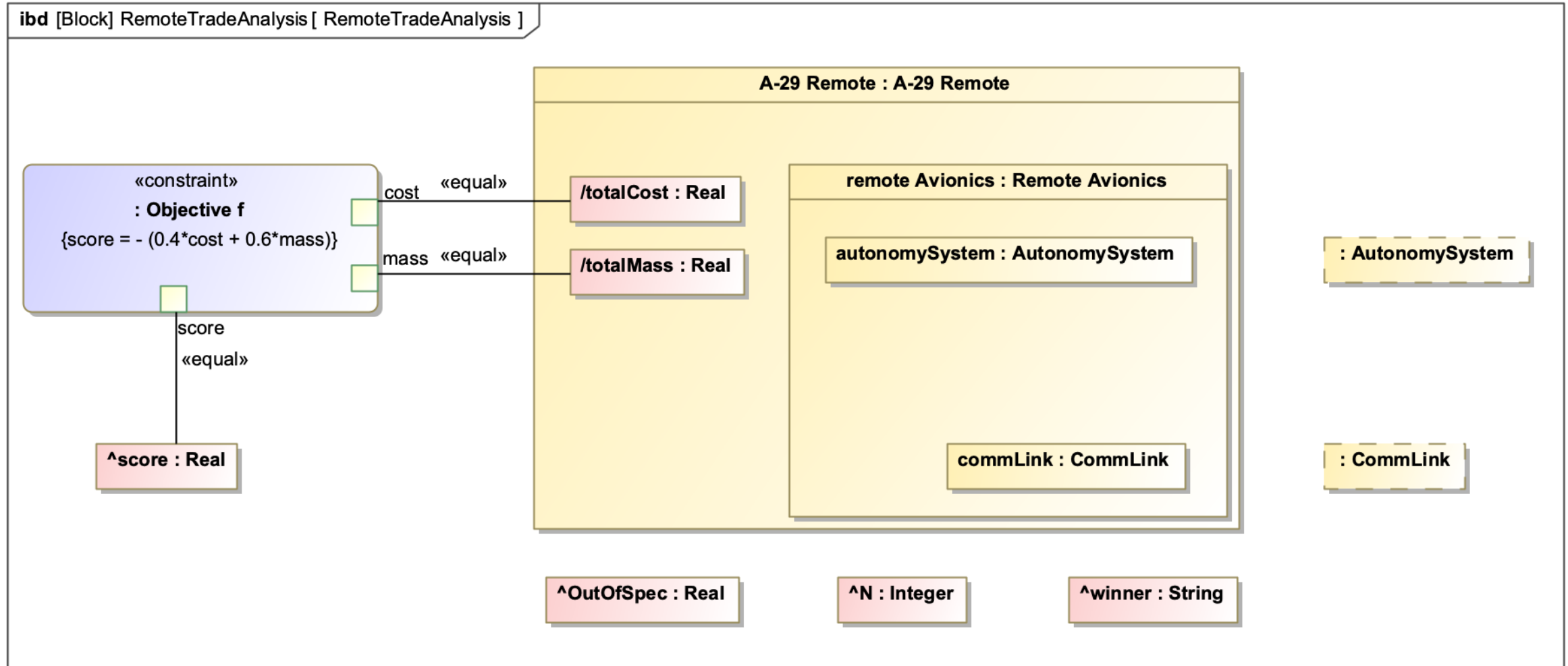


# Interconnect the block w/ the reference



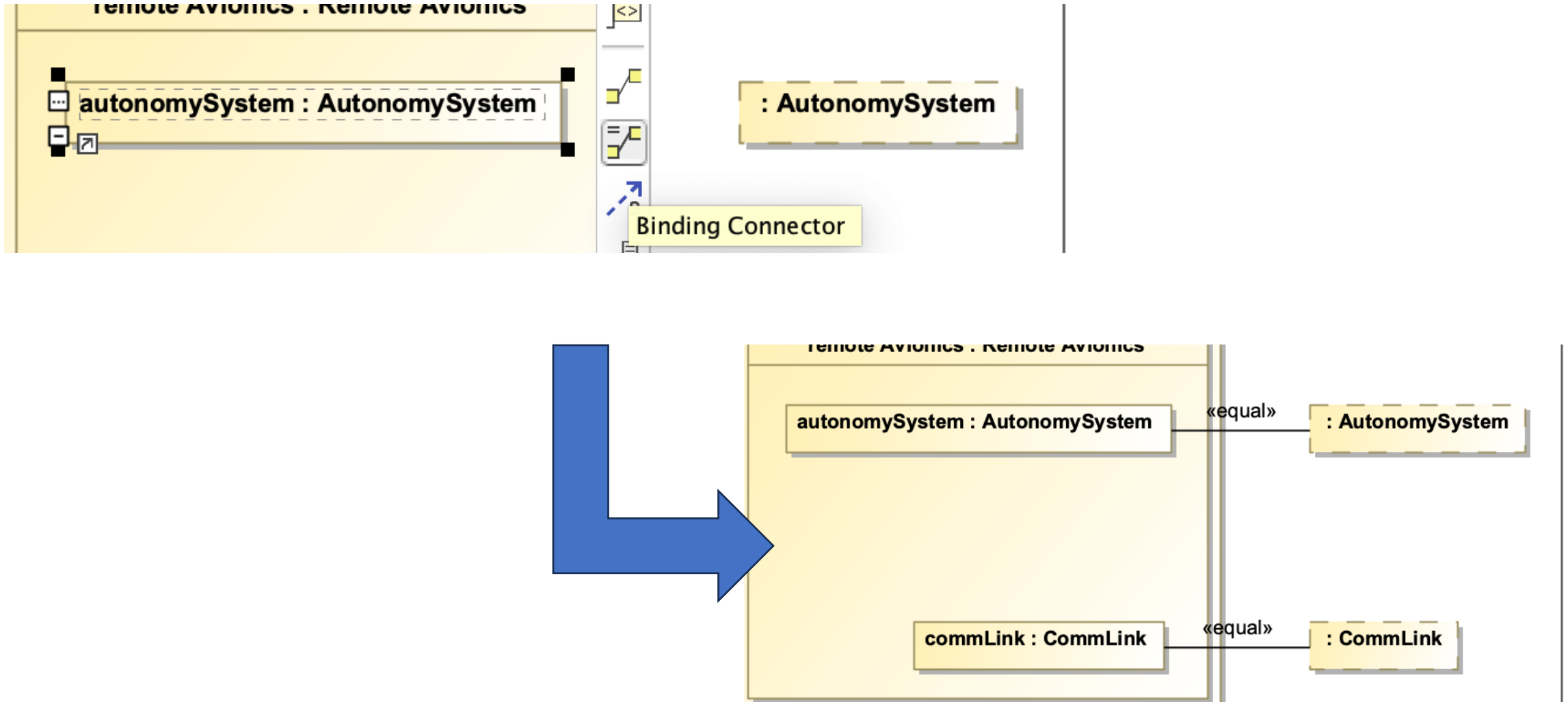


# How it shows



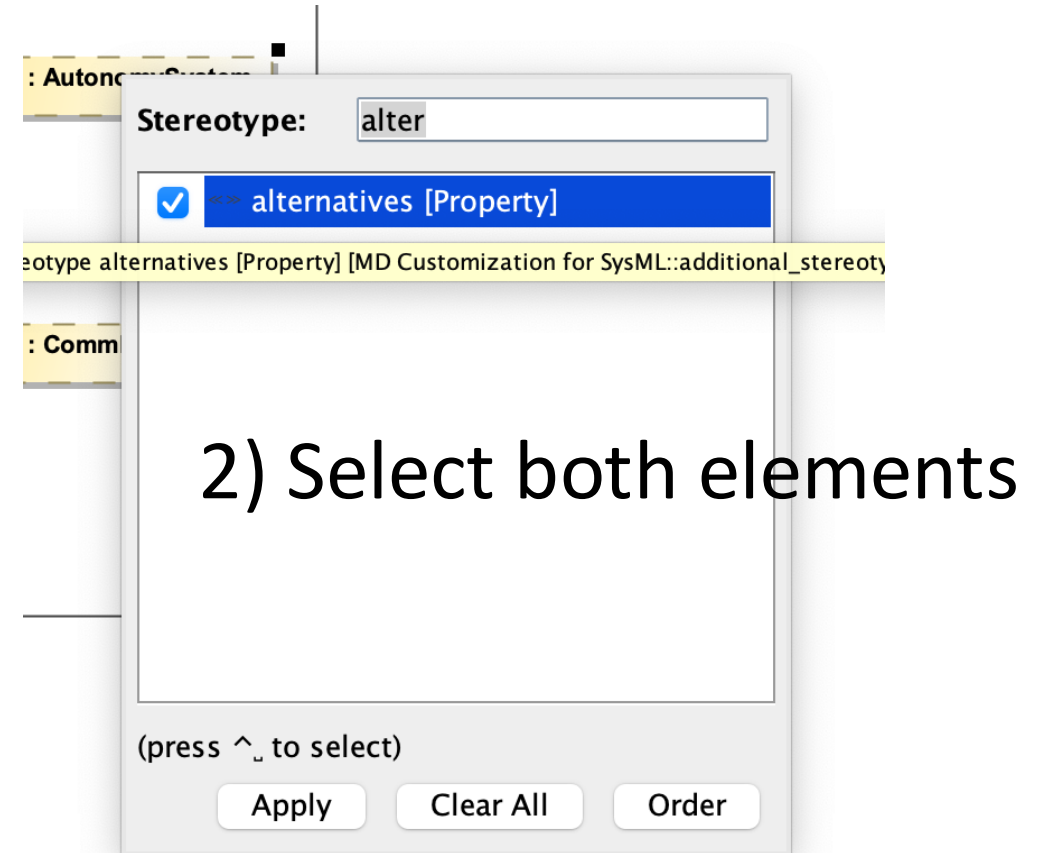
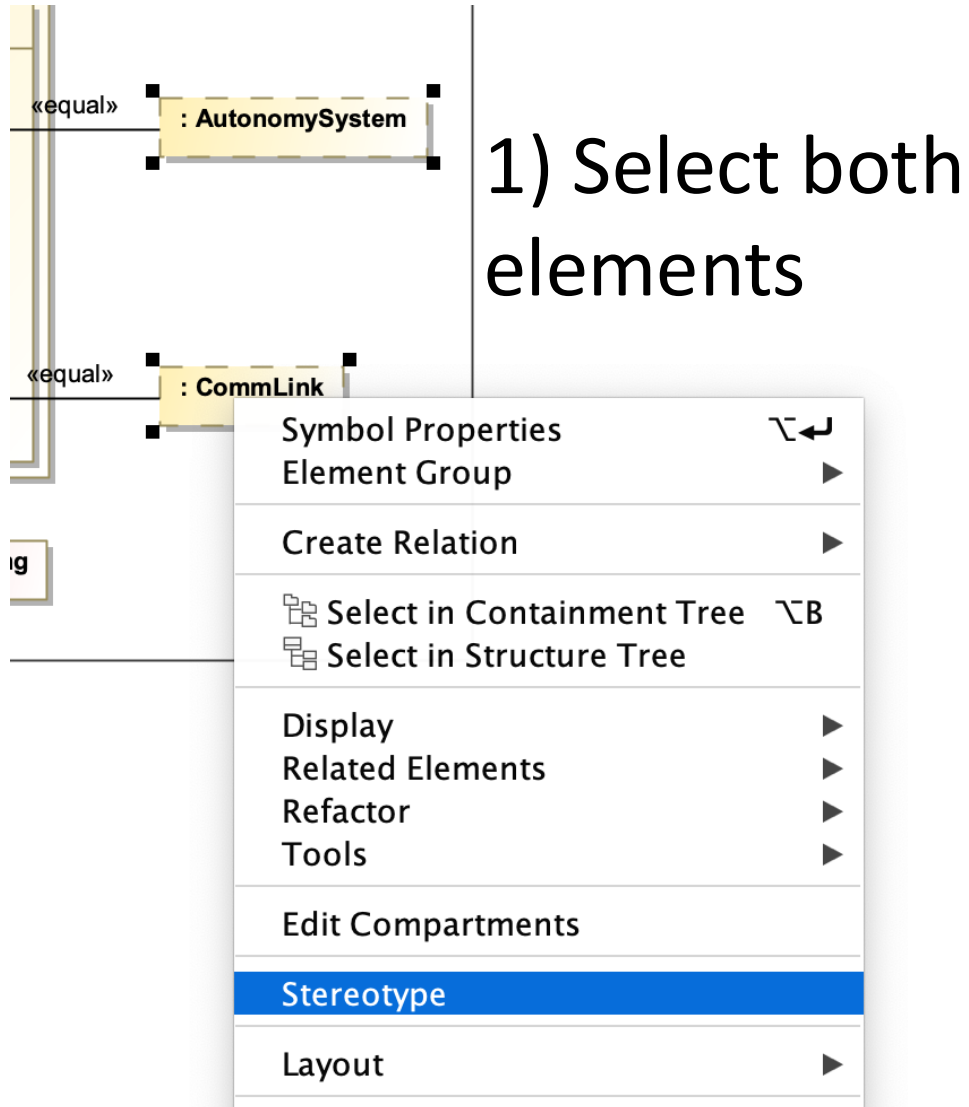


# Do a binding connection



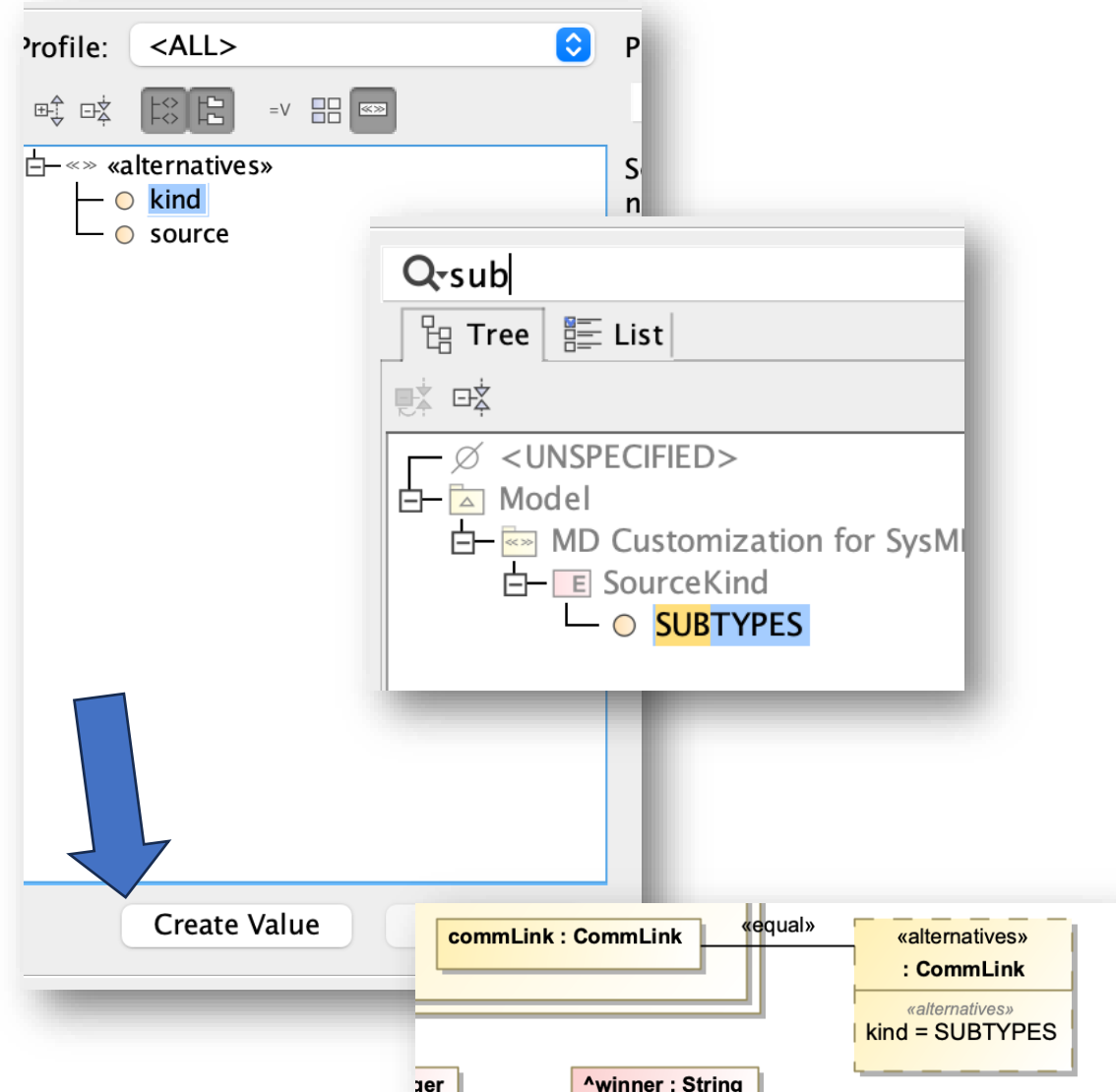
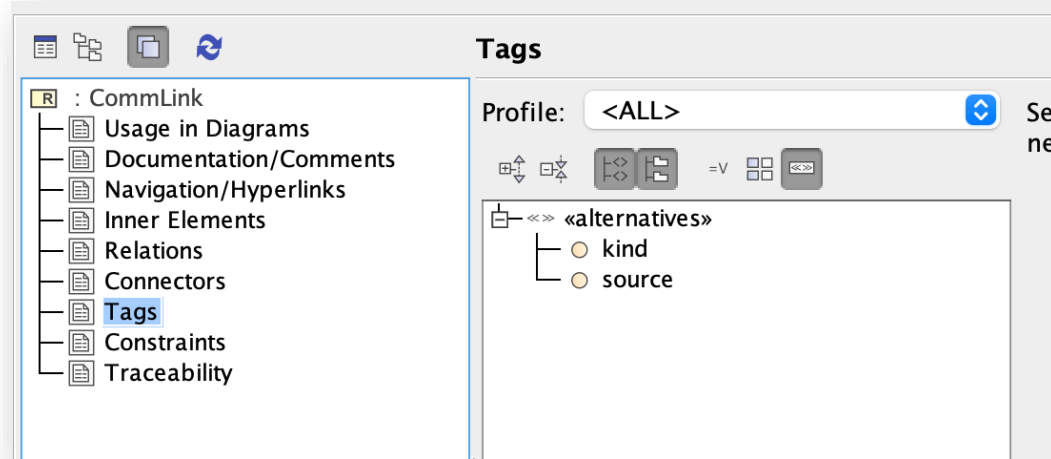
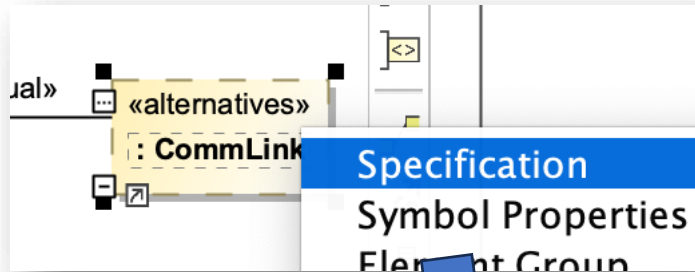


# Change the stereotypes to alternatives



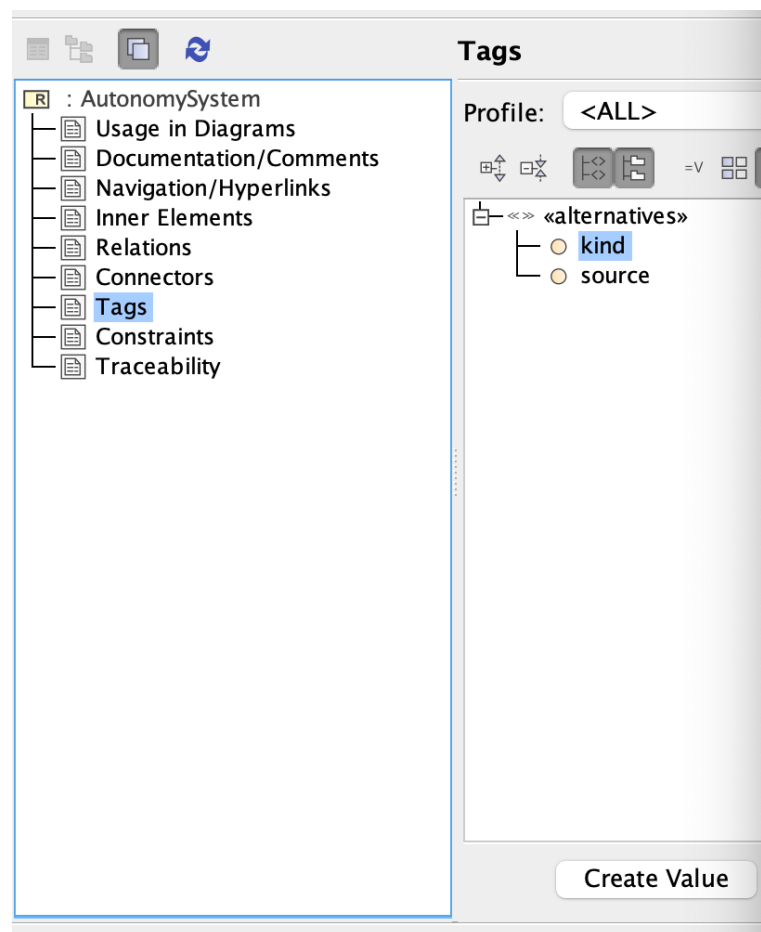
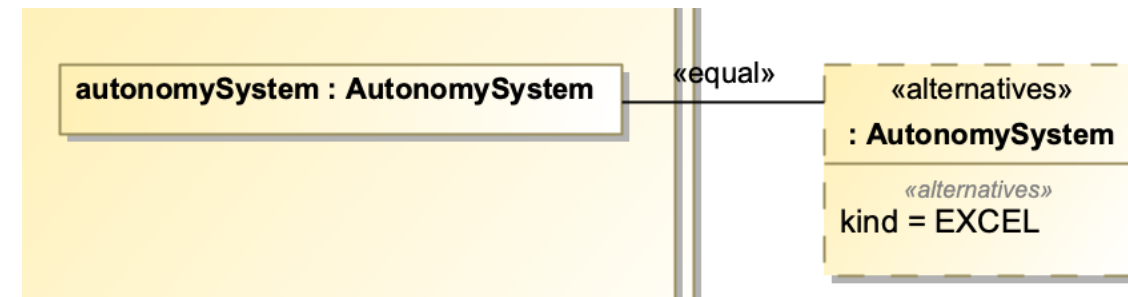
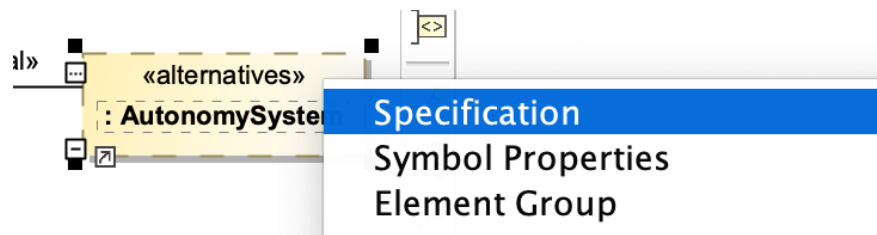


# Link the specialization w/ the references

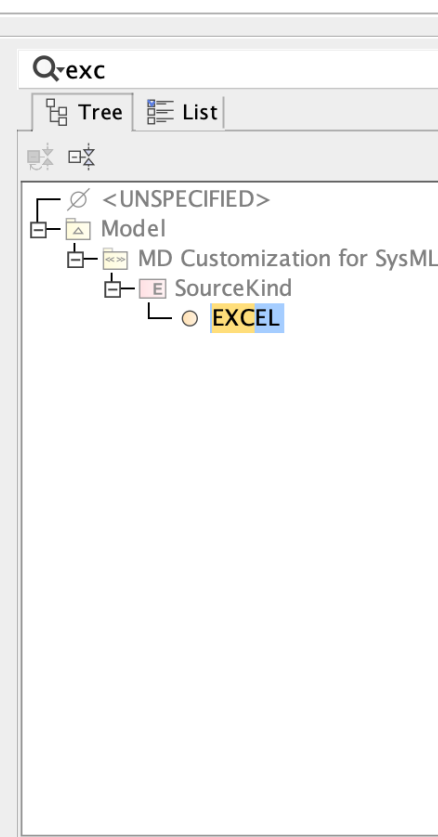




# Now link to the excel.



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



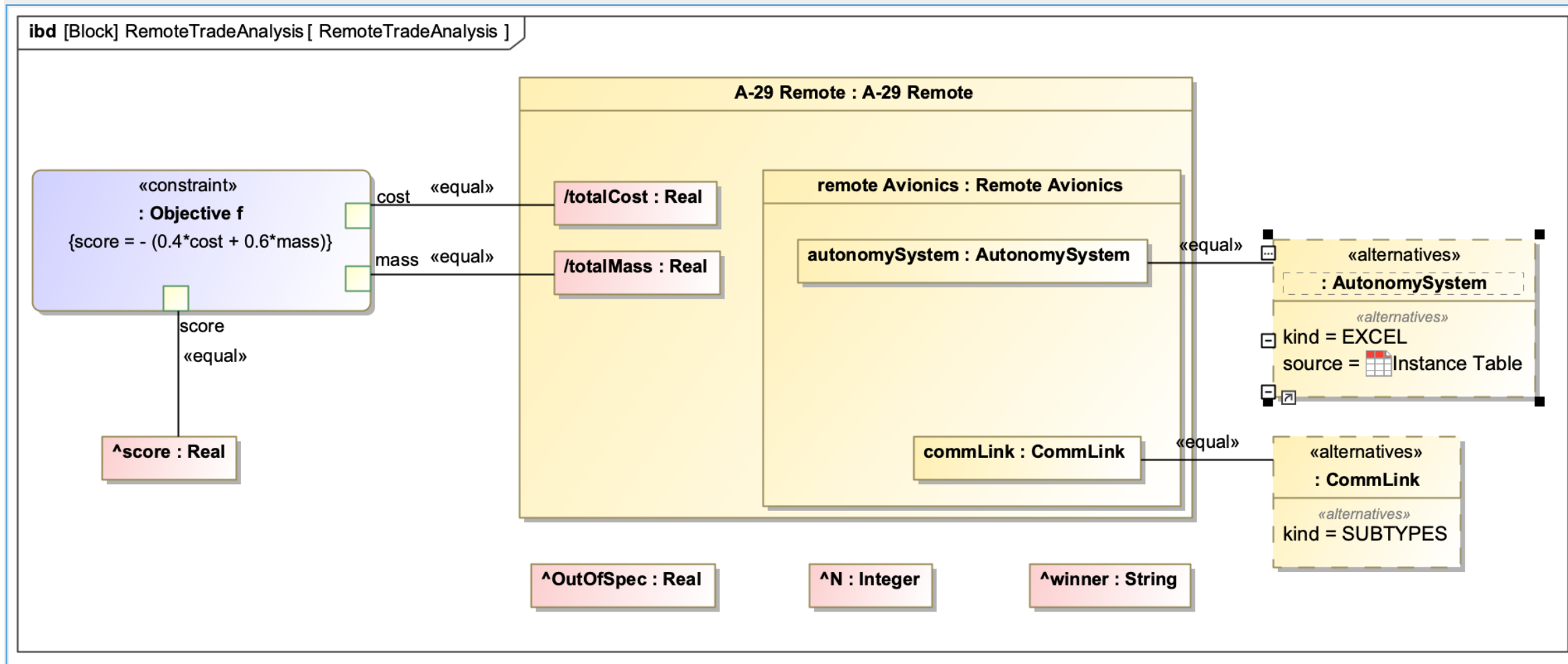


# Adjust the source link w/ the excel - instance table of the autonomy

The image shows a software interface with two overlapping windows. The background window is titled "Specification of Reference Property <>" and contains a section "Element tagged value specification" with the instruction "Select a tag and click the Create Value button to create new value for it." Below this is a "Tags" panel with a tree view for "AutonomySystem" containing items like "Usage in Diagrams", "Documentation/Comments", "Navigation/Hyperlinks", "Inner Elements", "Relations", "Connectors", "Tags" (highlighted), "Constraints", and "Traceability". To the right of the tree is a "Profile:" dropdown set to "<ALL>" and a table with columns for "kind" (set to "EXCEL") and "source". The foreground window is titled "Select Elements" and contains the instruction "Select, search for, or create elements". It has a search bar with the text "Q|" and radio buttons for "Search for:" with options "Type and Package", "Property", and "Any Element" (selected). Below the search bar are tabs for "Tree" and "List". The "Tree" view shows a hierarchy starting with "<UNSPECIFIED>" and "Model", followed by "Relations", "MD Customization for Requirements", "MD Customization for SysML", "MD Customization for ViewsViewpoints", "Requirements", "SuD", "Systems", and "Instance Table" (highlighted). Other items in the tree include "A-29 Platform", "AutonomySystem", "CommLink", and "FlightControl".



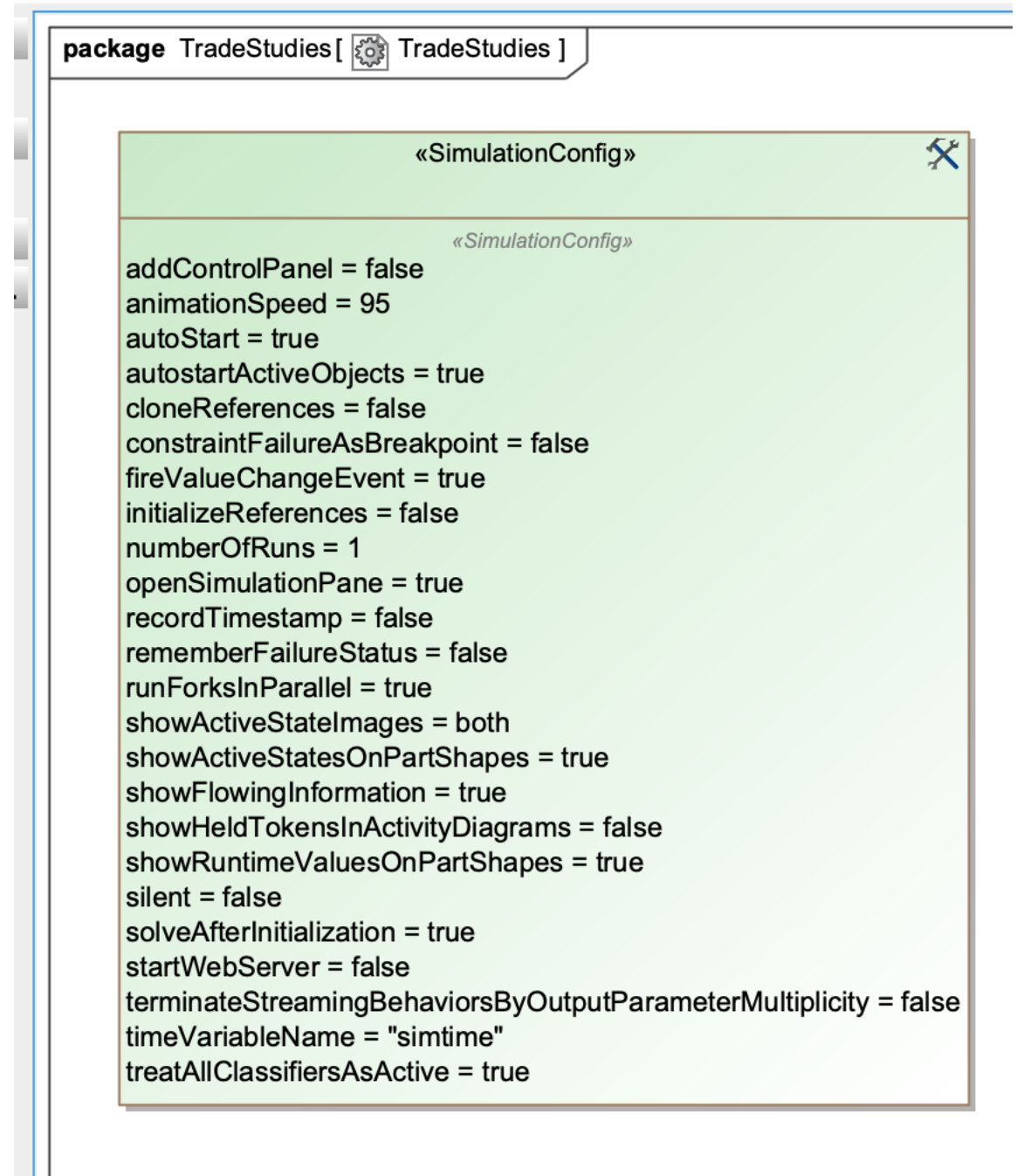
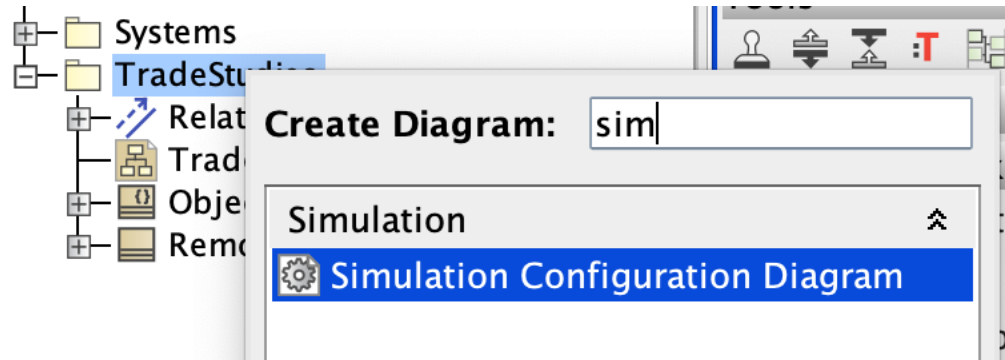
# All settle







# Create the Execution





# Link the trade studies

The screenshot displays a software interface with several panels. On the left, a 'Containment' tree shows a hierarchy: Model > Results > TradeStudies. The 'Results' folder is selected. Below this, a 'Properties' panel shows the 'Package' section with the following details:

Name	Results
Qualified Na...	Results
Owner	Model

In the center, a 'Tools' panel lists various tools, including 'Simulation Configuration', 'User Interface Configuration', 'Image Switcher', 'Active Image', 'Time Series Chart', 'Timeline Chart', 'Histogram', 'Select Properties Config', 'Sequence Diagram Generator', 'CSV Export', and 'Simulation Log'. A red arrow points from the 'Simulation Configuration' tool to the '«SimulationConfig»' element in the right panel.

On the right, the 'TradeStudies' package is shown, containing a '«SimulationConfig»' element. The configuration for this element is as follows:

```
package TradeStudies [ TradeStudies ]

«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
```



# Setup result location

The screenshot displays a software interface with several panels. A red arrow points from the 'Results' folder in the 'Model' tree to the 'Result Location' property in the 'Specification of Simulation Configuration' dialog.

**Model Tree:**

- Model
  - Requirements
  - Results
  - SuD
  - Systems
  - TradeStudies
    - Relations
    - TradeStudies
    - TradeStudies
    - Objective f
    - random name :)
    - RemoteTradeAnalysis

**Properties Panel:**

Properties

Element Symbol Tags

All

Simulation Configuration

Property	Value
Name	random name :)
Element ID	_2021x_2_58b20
Terminate St...	<input type="checkbox"/> false
Silent	<input type="checkbox"/> false
Animation Sp...	95
Show Active ...	<input checked="" type="checkbox"/> true

**Specification of Simulation Configuration random name :)**

Specify properties of the selected Simulation Configuration in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

**random name :)**

Properties: All

Property	Value
Name	random name :)
Element ID	_2021x_2_58b2086d_1756671461685_182
Terminate Streaming Behaviors By Output...	<input type="checkbox"/> false
Execution Target	RemoteTradeAnalysis [TradeStudies]
Auto Start	<input checked="" type="checkbox"/> true
Number Of Runs	1
Result Location	
Record Timestamp	<input type="checkbox"/> false
UI	
Execution Listeners	
Solve After Initialization	<input checked="" type="checkbox"/> true
Clone References	<input type="checkbox"/> false
Constraint Failure As Breakpoint	<input type="checkbox"/> false
Remember Failure Status	<input type="checkbox"/> false

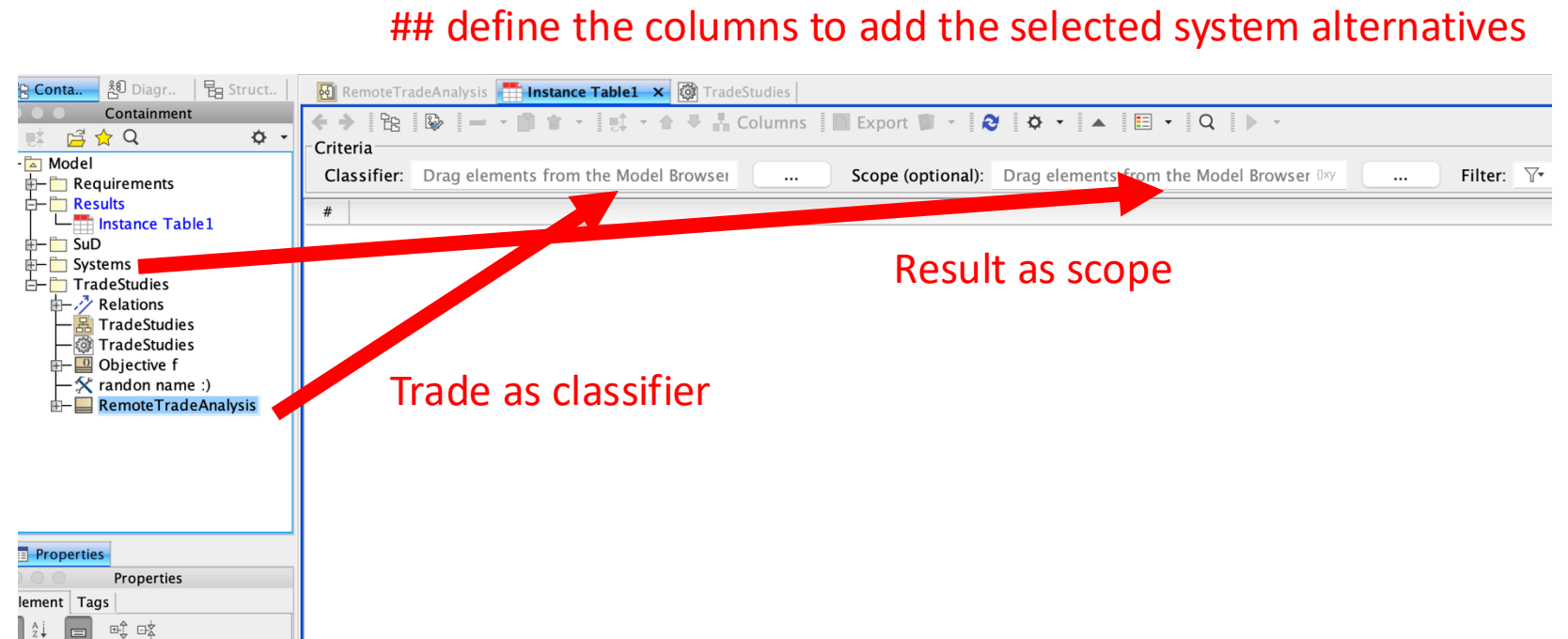
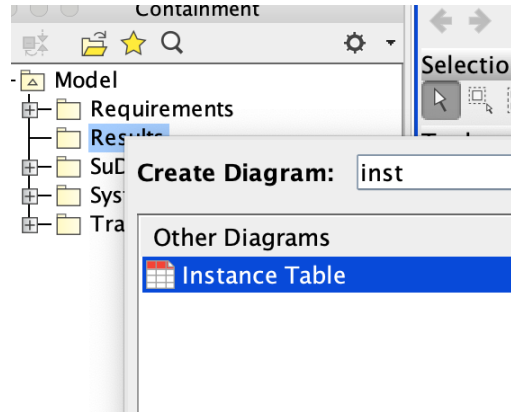
**Name**  
The name of the NamedElement.

Type here to filter properties

Close Back Forward Help



# Create an instance table to save the results





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

The screenshot shows the TradeStudies software interface. The top menu bar includes RemoteTradeAnalysis, Systems, Requirements, Instance Table1 (active), SuD, and TradeStudies. Below the menu is a toolbar with various icons. The main area displays a table with the following data:

#	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?

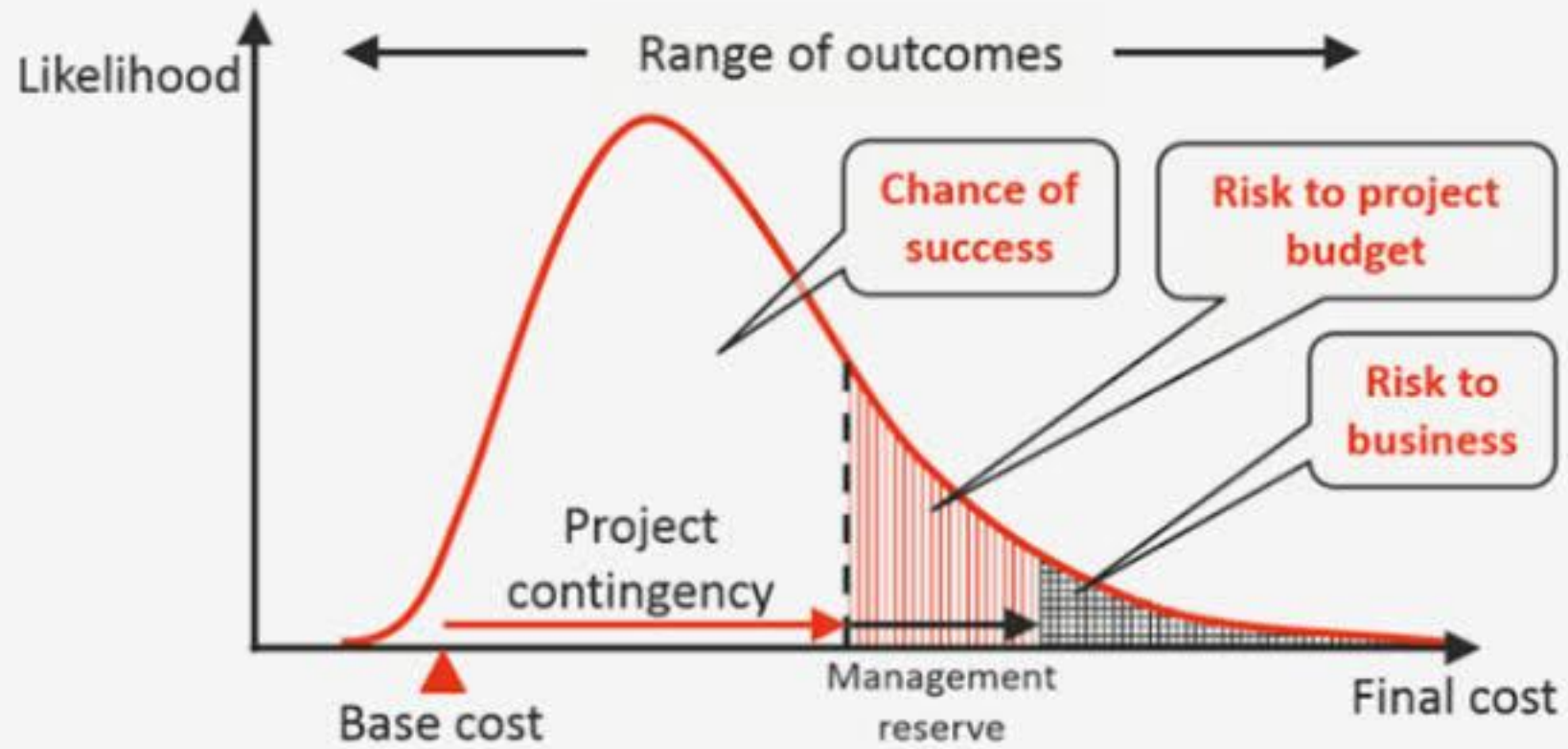
- Definition:
  - A **Monte Carlo Analysis** is a method that uses **random sampling and probability** to explore the behavior of a system.
  - Instead of calculating a single deterministic answer, we simulate the system **thousands of times** with different possible input values.



# 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.







# 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  $\geq$  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).

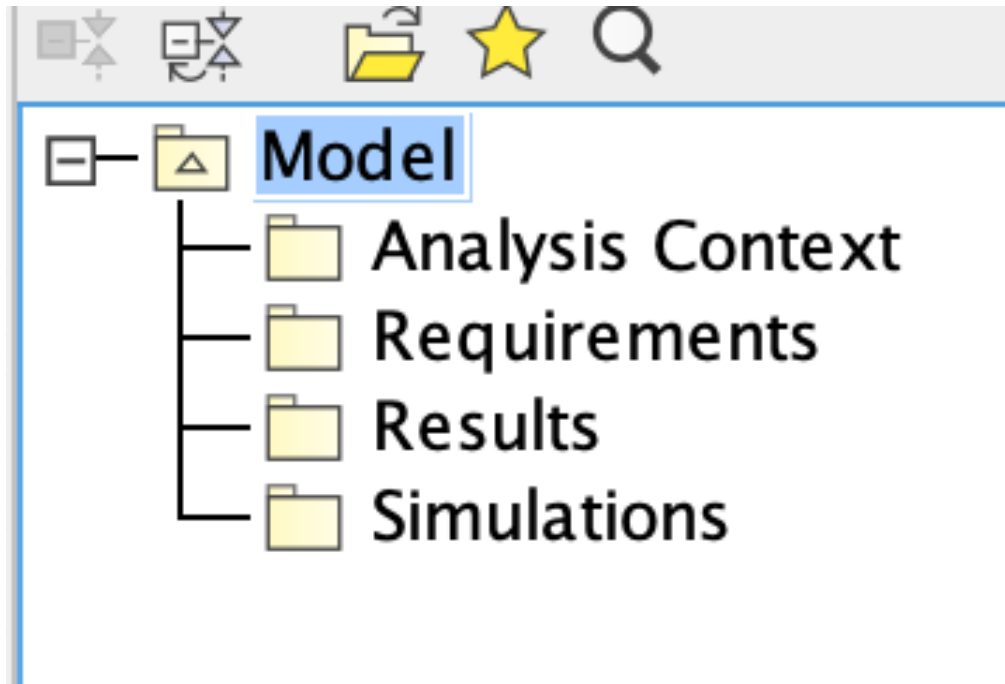


# Steps:

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

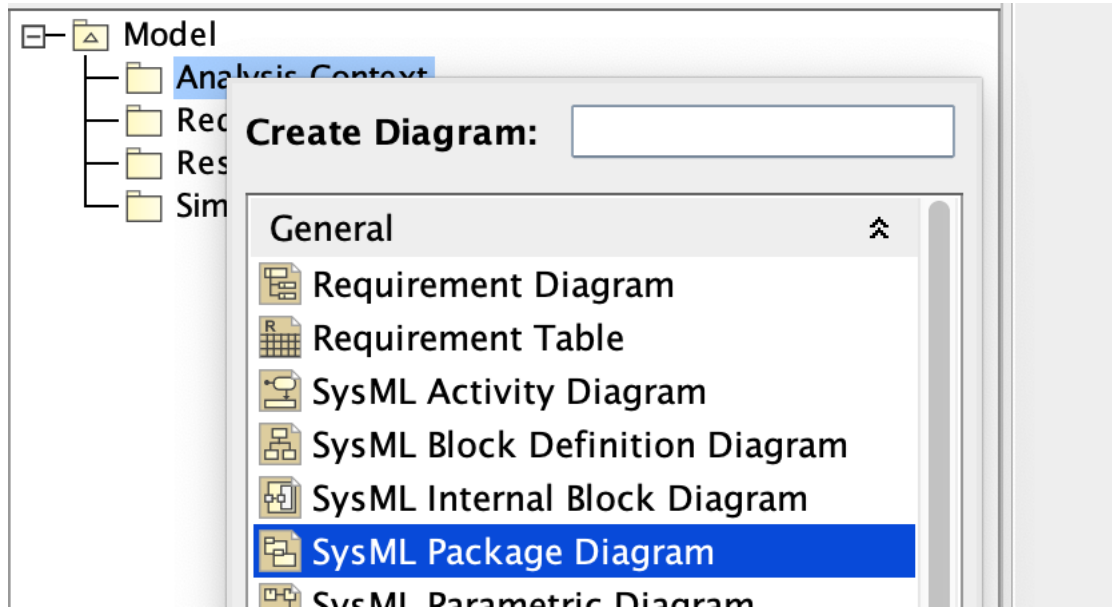


# 1 – Start creating the Package Structure

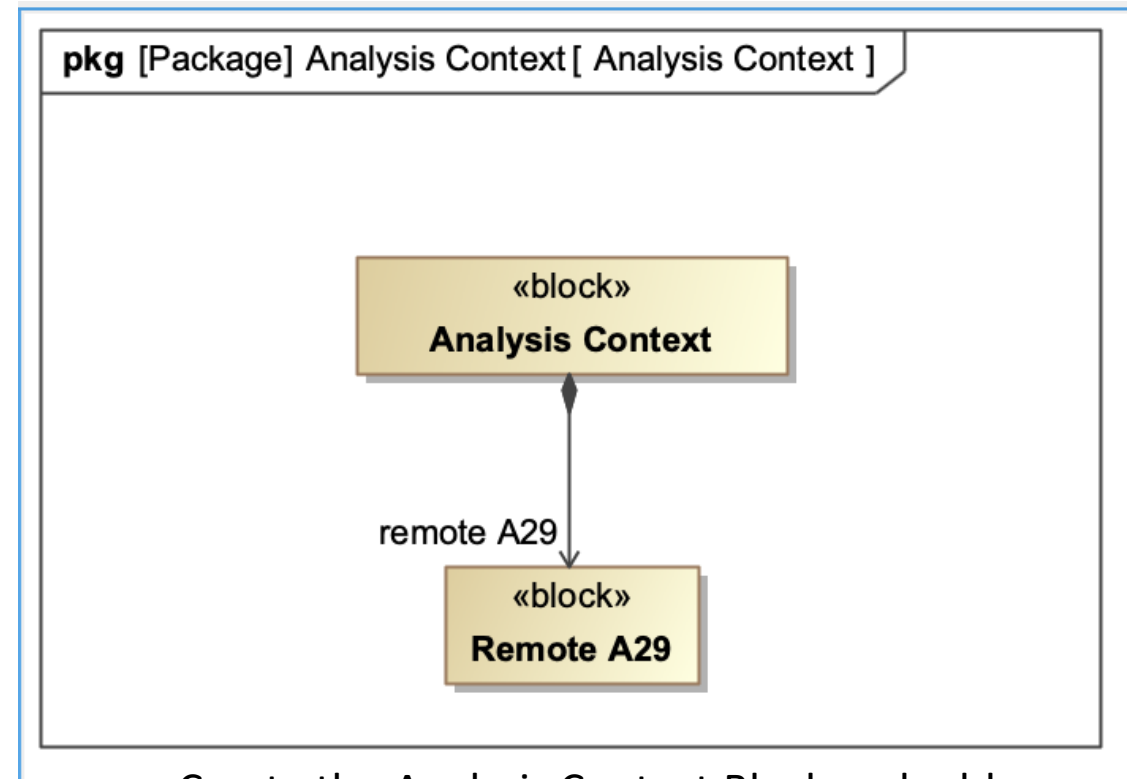




# 1 – Create a single element that we will apply Monte Carlo



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

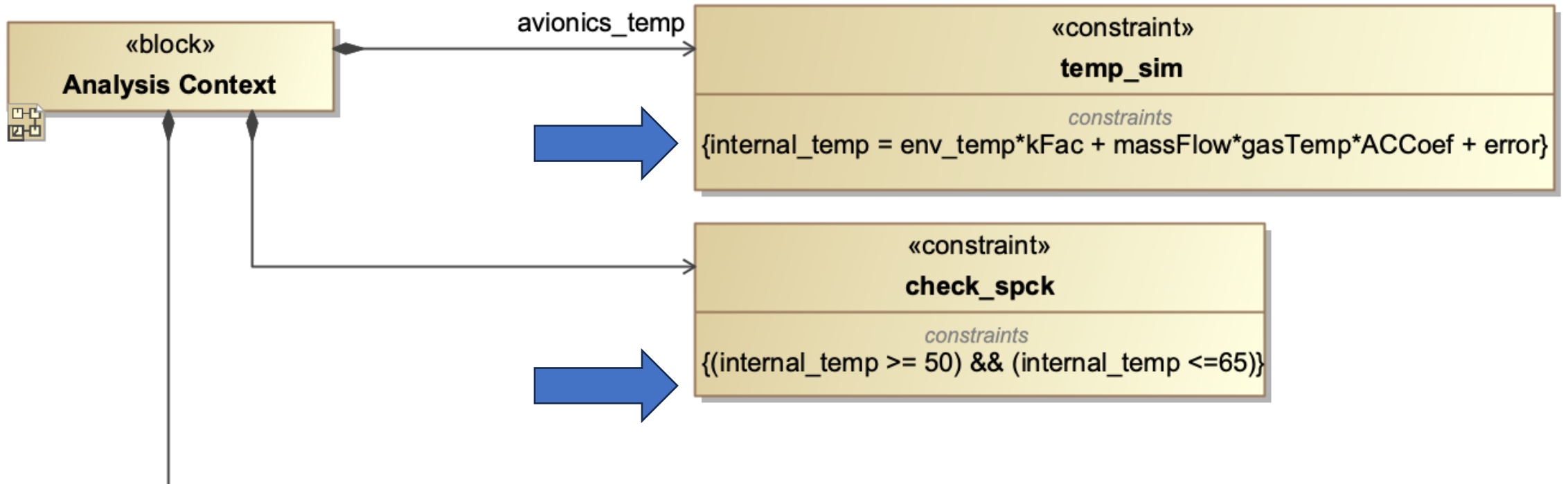


Create the Analysis Context Block and add the Remote A29





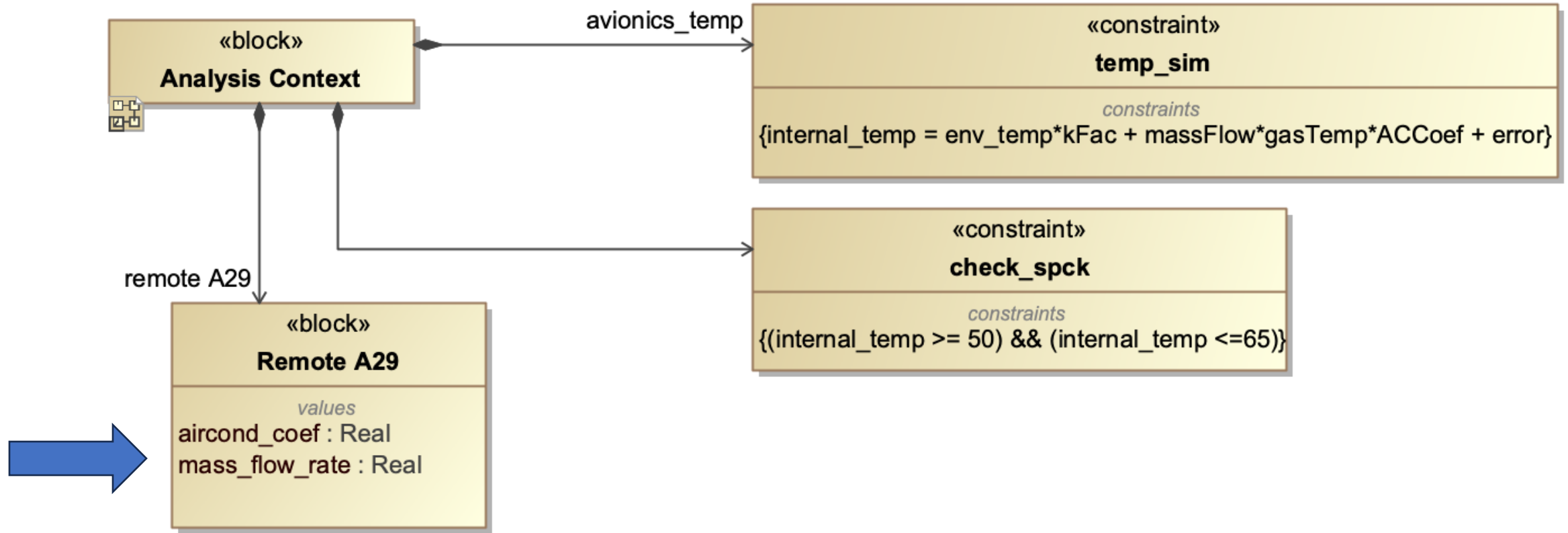
1. Define two constraints to create a simple “math” (remember the parametric diagram!?)



\*\*random equation

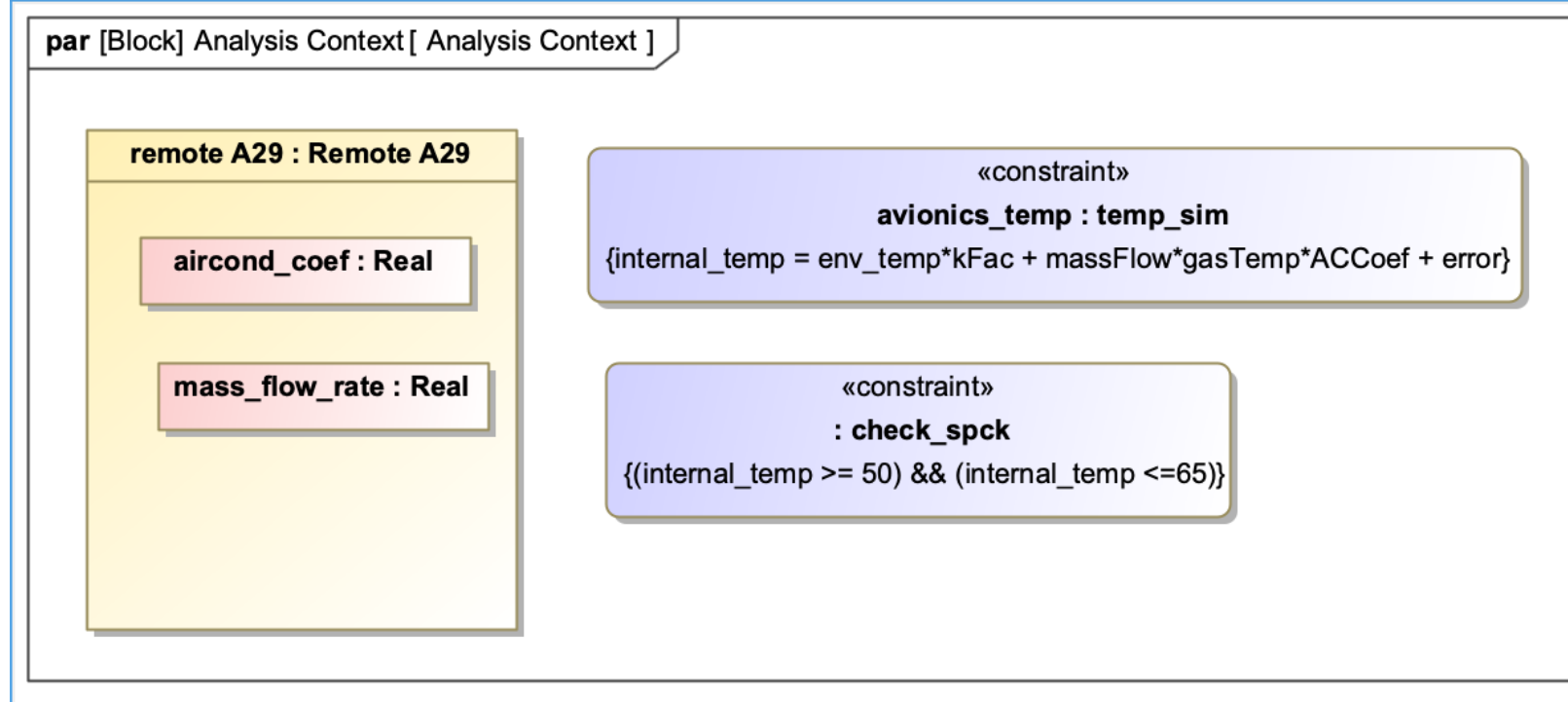
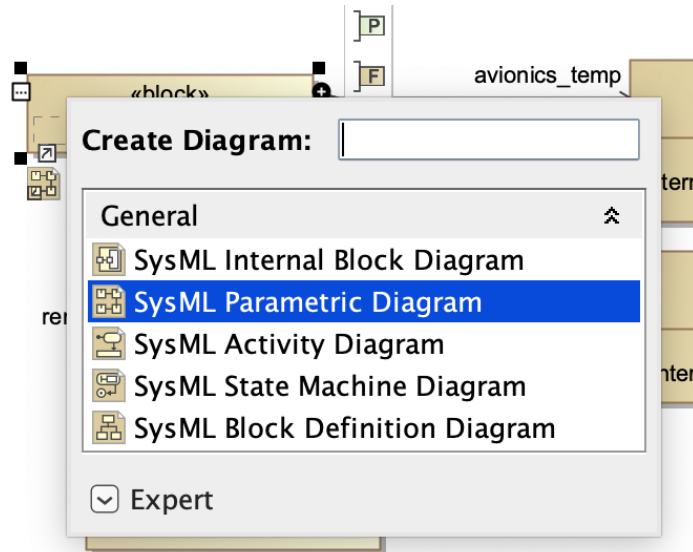


# 1. Create values to the SuD



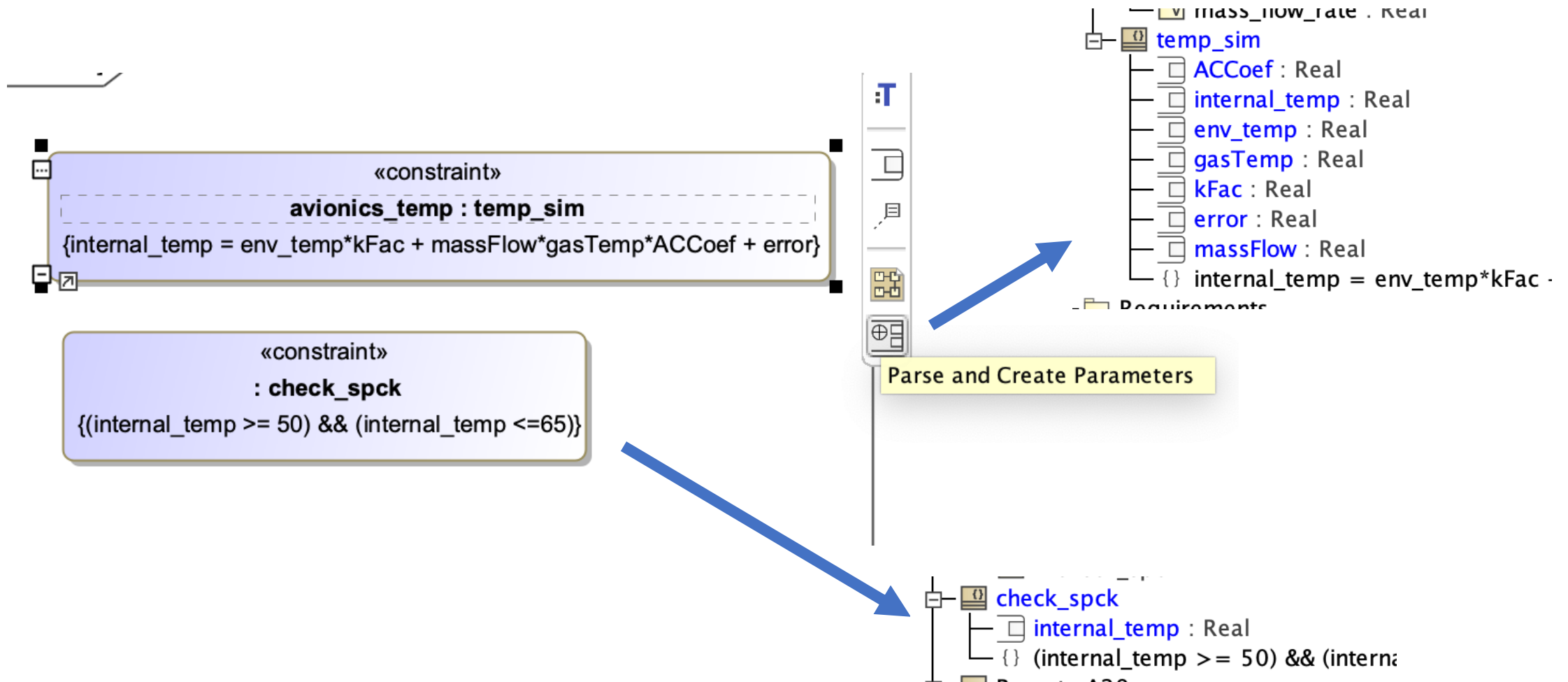


# 1. Create the Parametric Diagram of the Analysis Context





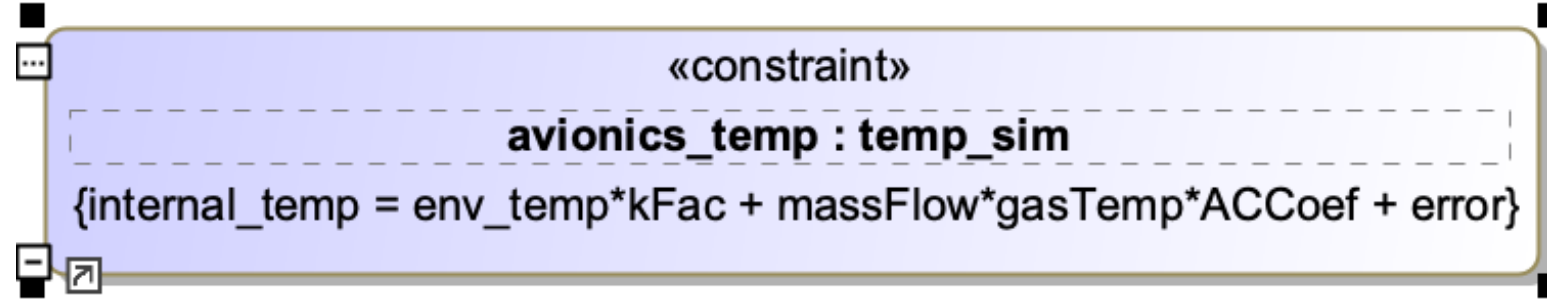
# 1. Parse the constraints





# 1. Expose the ports

context ]



Display All Parameters



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

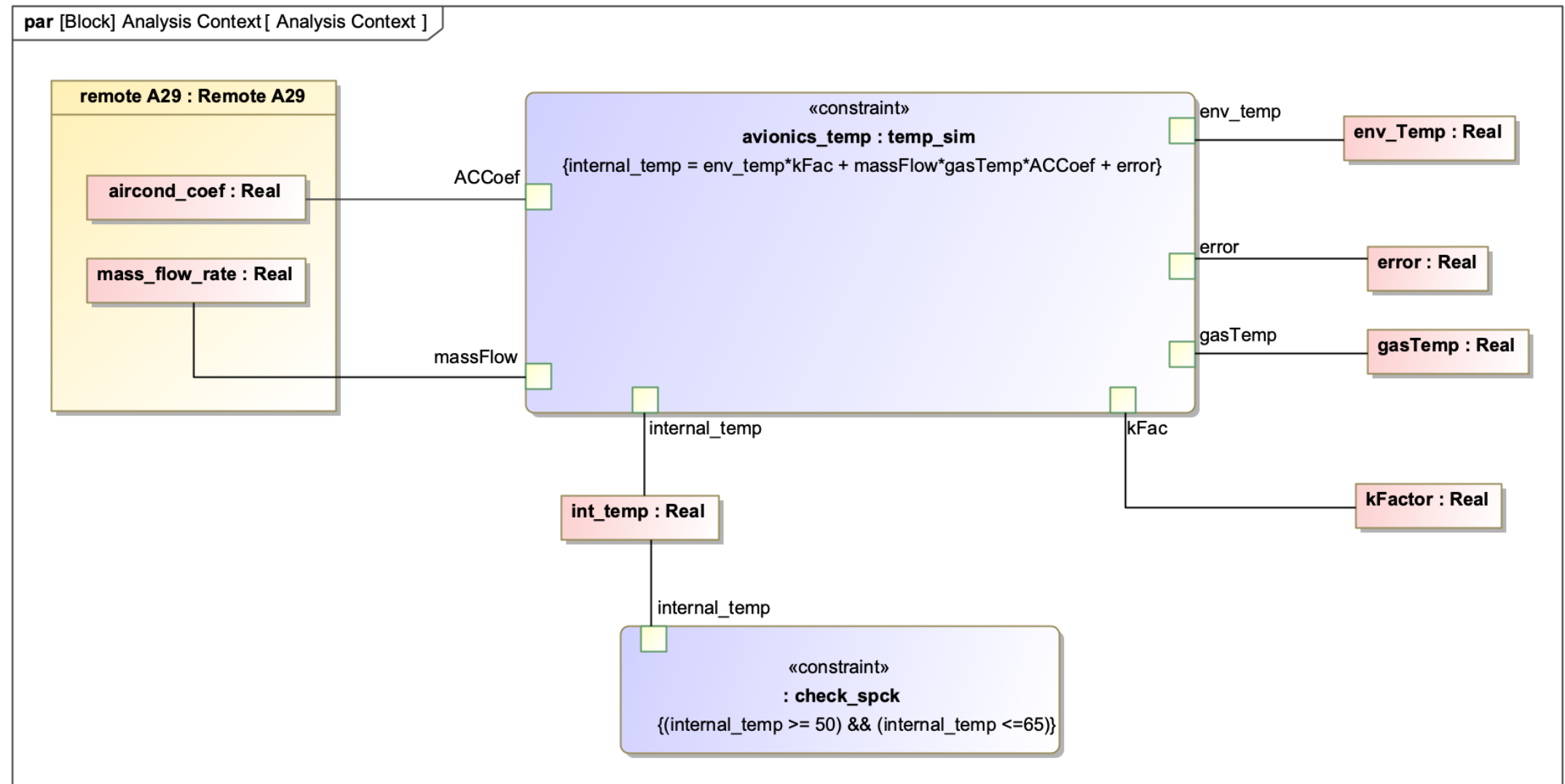
«constraint»

: check\_spck

{{(internal\_temp >= 50) && (internal\_temp <=65)}}



# Create analysis variables and link to the aircraft:





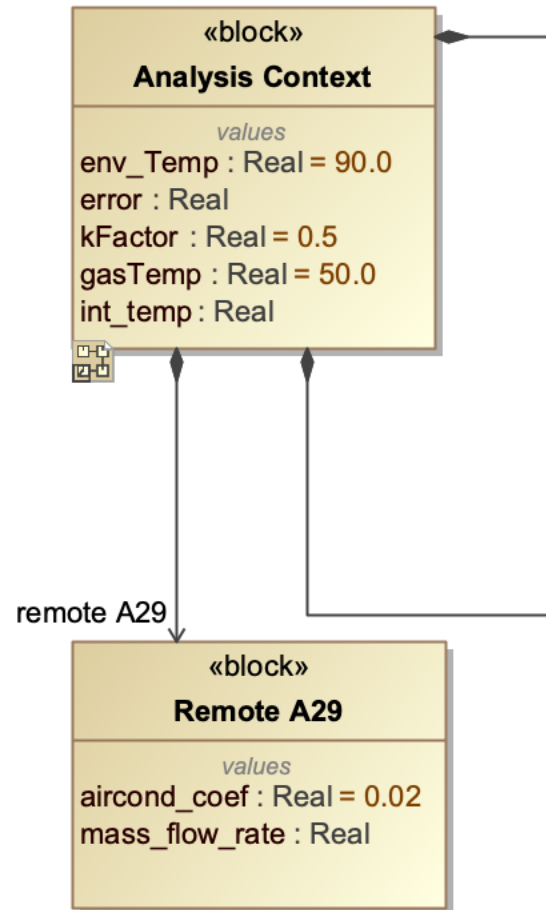
pkg [Package] Analysis Context [ Analysis Context ]







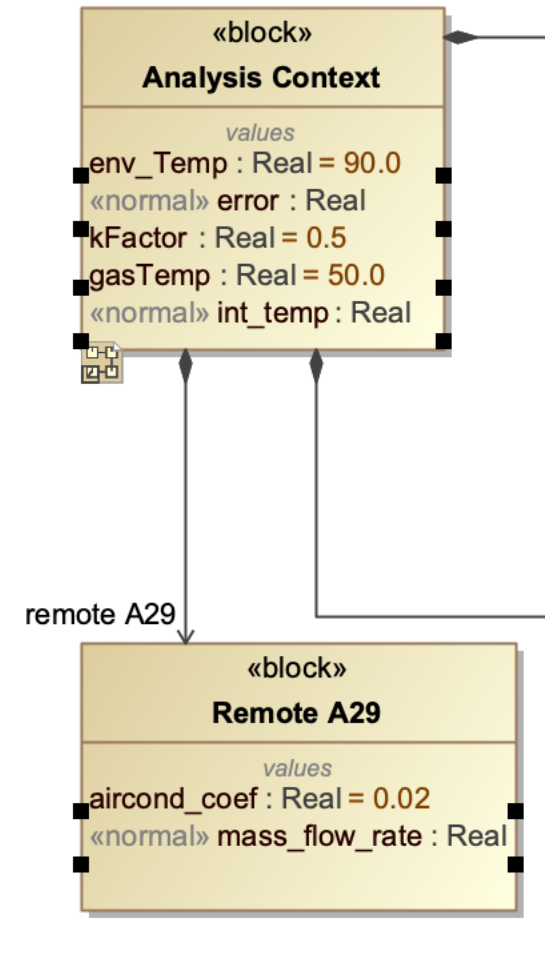
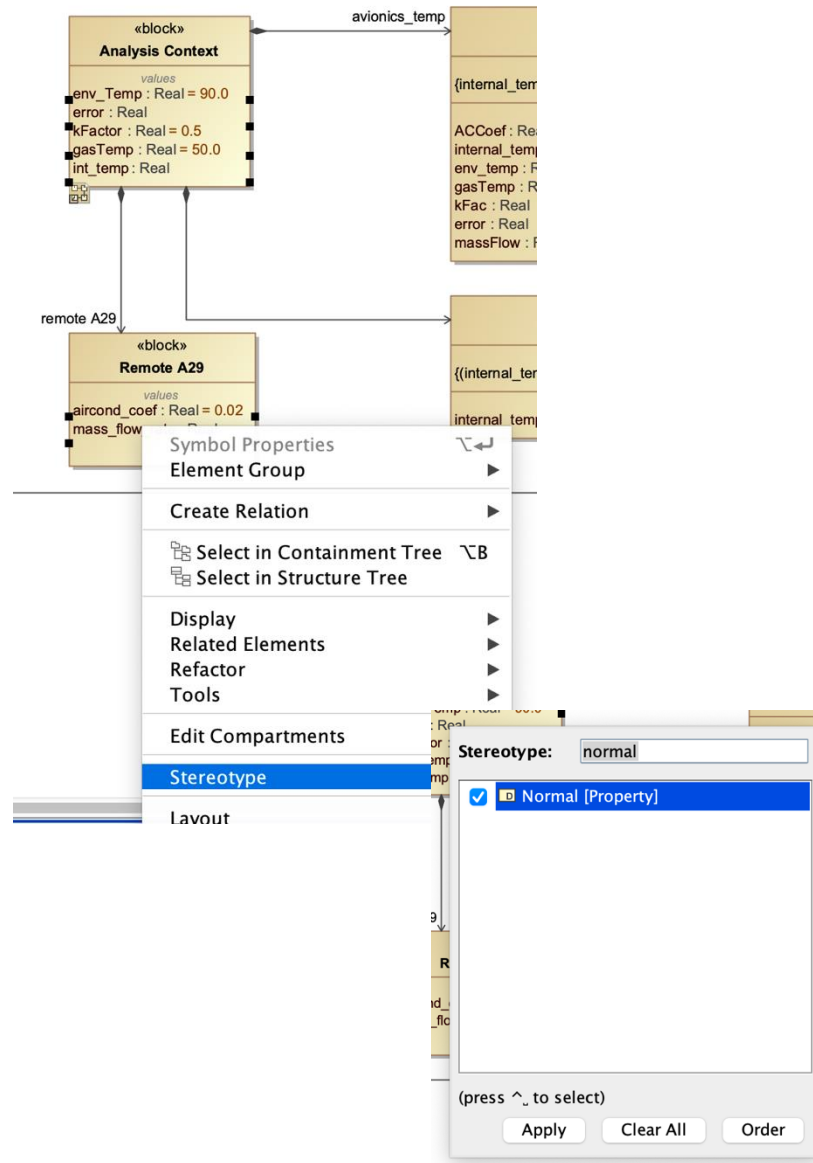
# 1. Initialize some values





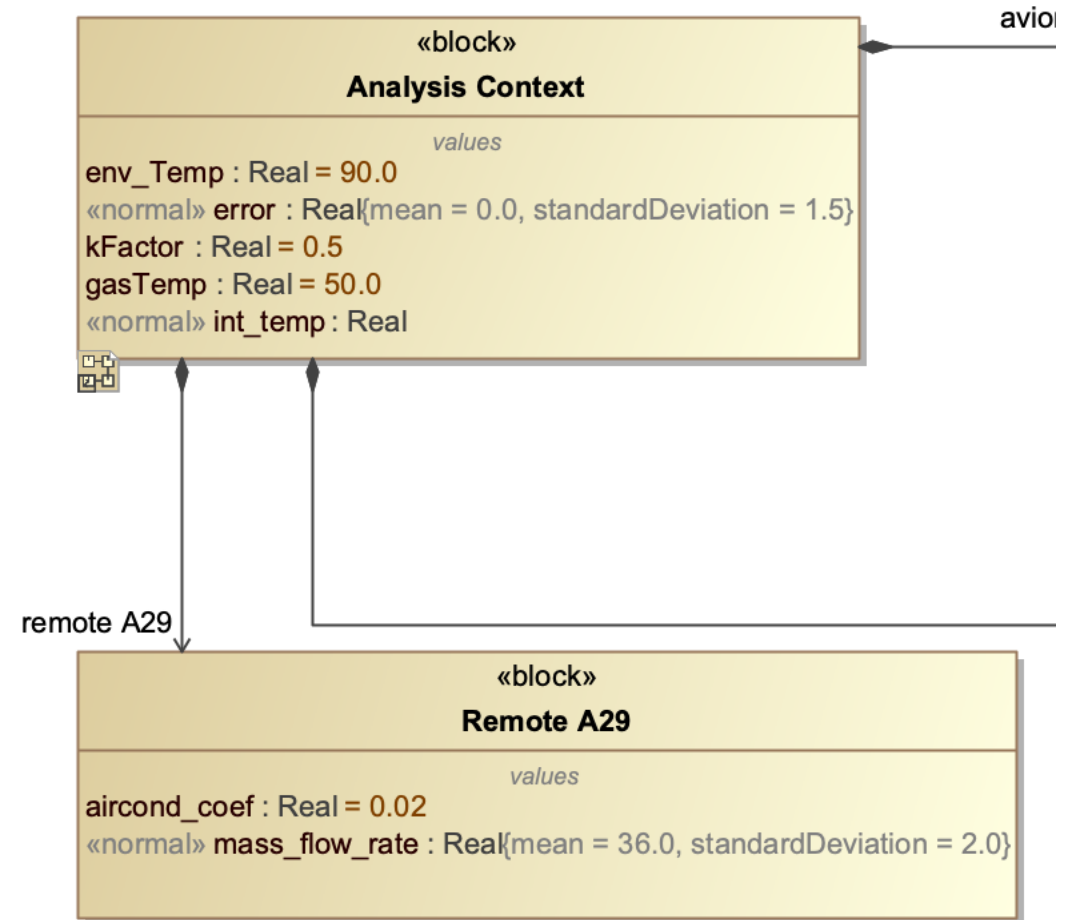
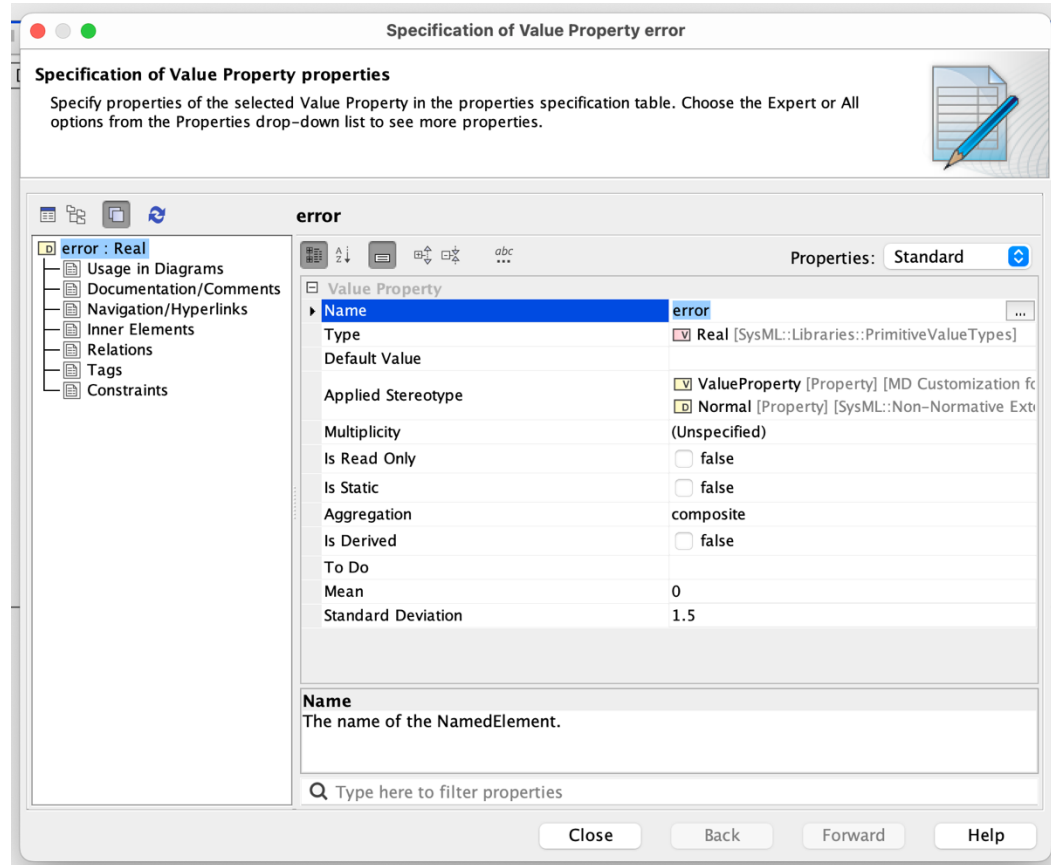
## 2. Add the normal distribution types

- Select the values to vary:
  - Error
  - int\_temp
  - mass\_flow
- Add Normal stereotype



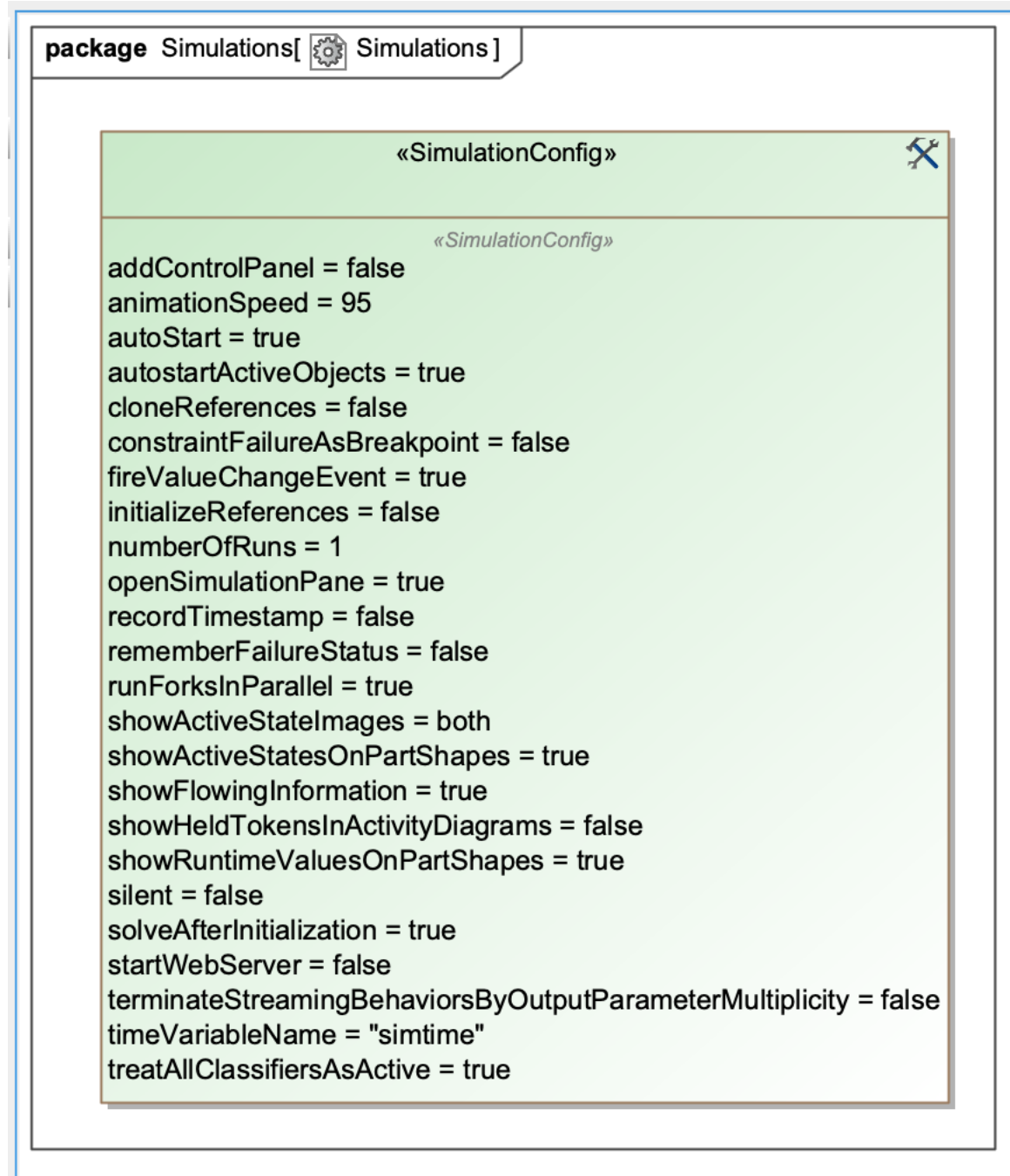
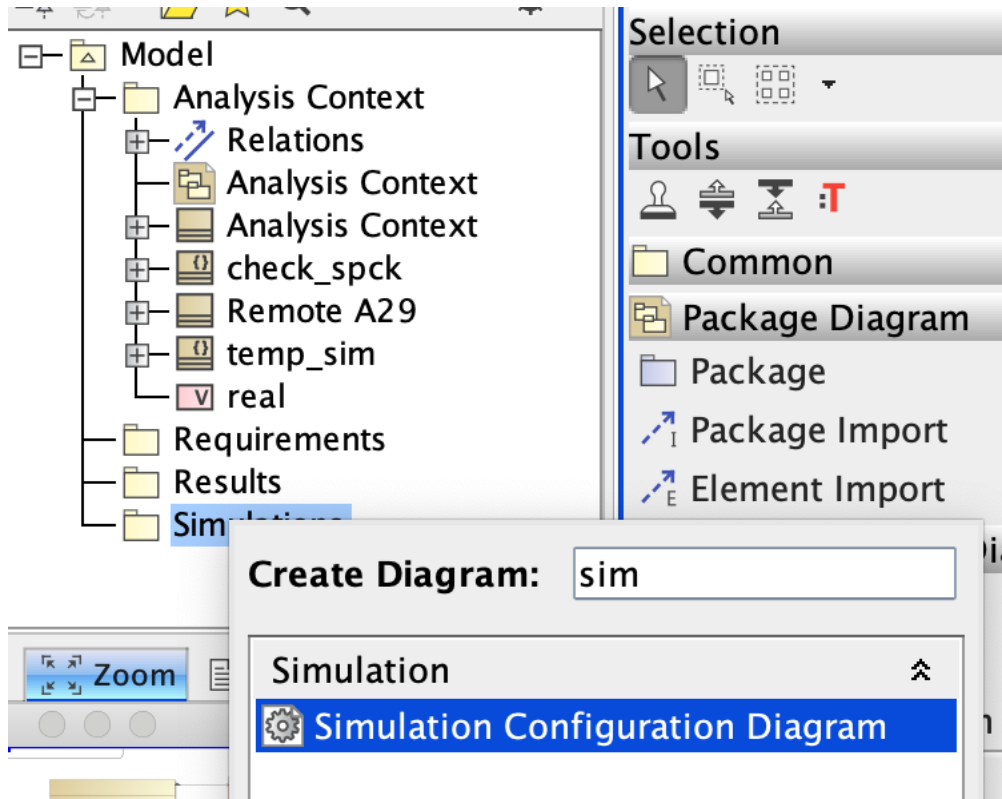


## 2. Fill mean and std deviation in error / mass\_flow





# 3. Create Simulation





### 3. Setup Simulation – bring the execution target

The screenshot displays a software interface with three main panels. The left panel shows a hierarchical tree structure under 'Model', with 'Simulations' and 'mc\_sim' highlighted. The middle panel, titled 'Selection' and 'Tools', lists various simulation-related tools, with 'Simulation Configuration Di...' selected. The right panel, titled 'package Simulations[ Simulations ]', shows a configuration window for '«SimulationConfig»'. A blue arrow points from the 'Simulation Configuration Di...' tool in the middle panel to the configuration window in the right panel.

**Model Tree Structure:**

- Model
  - Analysis Context
    - Relations
    - Analysis Context
    - Analysis Context
    - check\_spck
    - Remote A29
    - temp\_sim
    - real
  - Requirements
  - Results
  - Simulations
    - Simulations
    - mc\_sim

**Tools List:**


- Common
  - Simulation Configuration Di...
  - Simulation Configuration
  - User Interface Configuration
  - Image Switcher
  - Active Image
  - Time Series Chart
  - Timeline Chart
  - Histogram
  - Select Properties Config
  - Sequence Diagram Generator
  - CSV Export
  - Simulation Log


**Simulation Configuration Di... Settings:**

```
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. Set the number of runs



package Simulations[  Simulations ]

«SimulationConfig»  
**mc\_sim**

«SimulationConfig»

```
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
```



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

The screenshot displays a software interface with three main panels:

- Model Tree (Left):** A hierarchical view of the model structure. The 'Simulations' folder is expanded, showing 'mc\_sim' as a selected element.
- Selection Pane (Middle):** A list of tools and components. The 'Histogram' tool is highlighted, and a blue arrow points from it to the configuration editor.
- Configuration Editor (Right):** A panel for configuring the 'mc\_sim' package. It contains two sections:
  - «SimulationConfig» mc\_sim:** A list of configuration parameters for the simulation, such as '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"', and 'treatAllClassifiersAsActive = true'.
  - «Histogram»:** A section for configuring the histogram, including 'dynamic = false', 'annotateFailures = true', 'gridX = true', 'gridY = true', 'keepOpenAfterTermination = false', 'linearInterpolation = true', 'plotColor = "#BC334E"', and 'recordPlotDataAs = CSV'.





### 3. Configure Histogram value = int\_temp

The image shows two overlapping dialog boxes from a software application. The background dialog is titled "Specification of Histogram internal\_temp" and contains a "Specification of Histogram properties" section with instructions to specify properties in a table. It features a table with columns for "Name" and "Represents", and a "Value" section with fields for Title, X Label, Y Label, Dynamic, and Record Plot Data As. The foreground dialog is titled "Select Nested Properties" and contains a tree view of the "Analysis Context". The tree view shows a hierarchy of properties, with "int\_temp : Real" selected. The "Select Nested Properties" dialog also includes a "Clear All" button, a "Select All" button, and "OK" and "Cancel" buttons at the bottom.

**Specification of Histogram internal\_temp**

**Specification of Histogram properties**

Specify properties of the selected Histogram in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

**internal\_temp**

**Histogram**

Name	Represents

**Value**

Title

X Label

Y Label

Dynamic

Record Plot Data As

**Value**

Structural feature which value should be represented

Type here to filter properties

**Select Nested Properties**

Select nested properties in a root classifier

**Analysis Context**

- ☐ **check\_spck**
- ☐ **avionics\_temp : temp\_sim**
- ☐ **env\_Temp : Real = 90.0**
- ☐ **error : Real**
- ☐ **gasTemp : Real = 50.0**
- ☒ **int\_temp : Real**
- ☐ **kFactor : Real = 0.5**
- ☐ **remote A29 : Remote A29**

Clear All Select All

OK Cancel





### 3. Set “keep open” and “dynamic”

Specification of Histogram internal\_temp

**Specification of Histogram properties**

Specify properties of the selected Histogram in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

internal\_temp

Navigation/Hyperlinks

Properties: Standard

Histogram	
Name	internal_temp
Represents	Analysis Context [Analysis Context]
Value	int_temp : Real [Analysis Context::Analysis Cor
Title	
X Label	
Y Label	
Dynamic	<input checked="" type="checkbox"/> true
Record Plot Data As	CSV
Result File	CSV
Grid X	<input checked="" type="checkbox"/> true
Grid Y	<input checked="" type="checkbox"/> true
Plot Color	#BC334E
Keep Open After Termination	<input checked="" type="checkbox"/> true

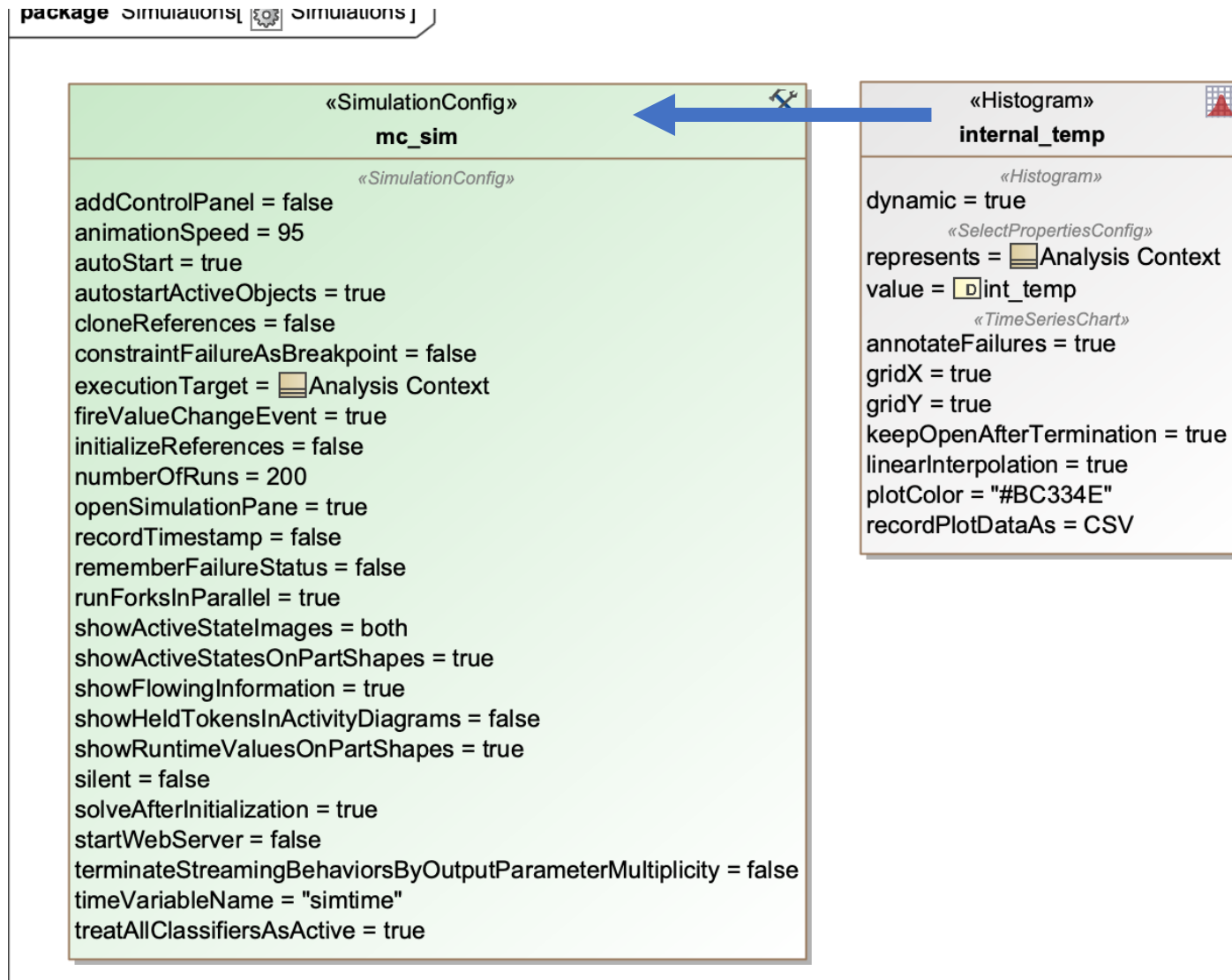
**Dynamic**  
If true, open the histogram during multiple runs which shows dynamic results, otherwise open the histogram at the end of all runs completion.

Type here to filter properties

Close Back Forward Help



### 3. Add histogram as UI





## 4. Run

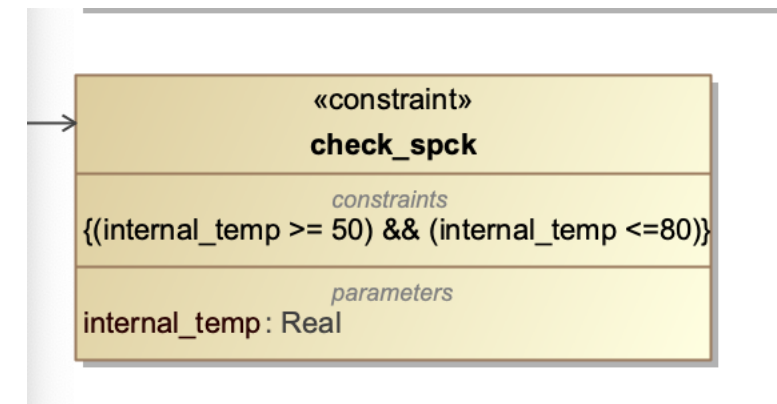
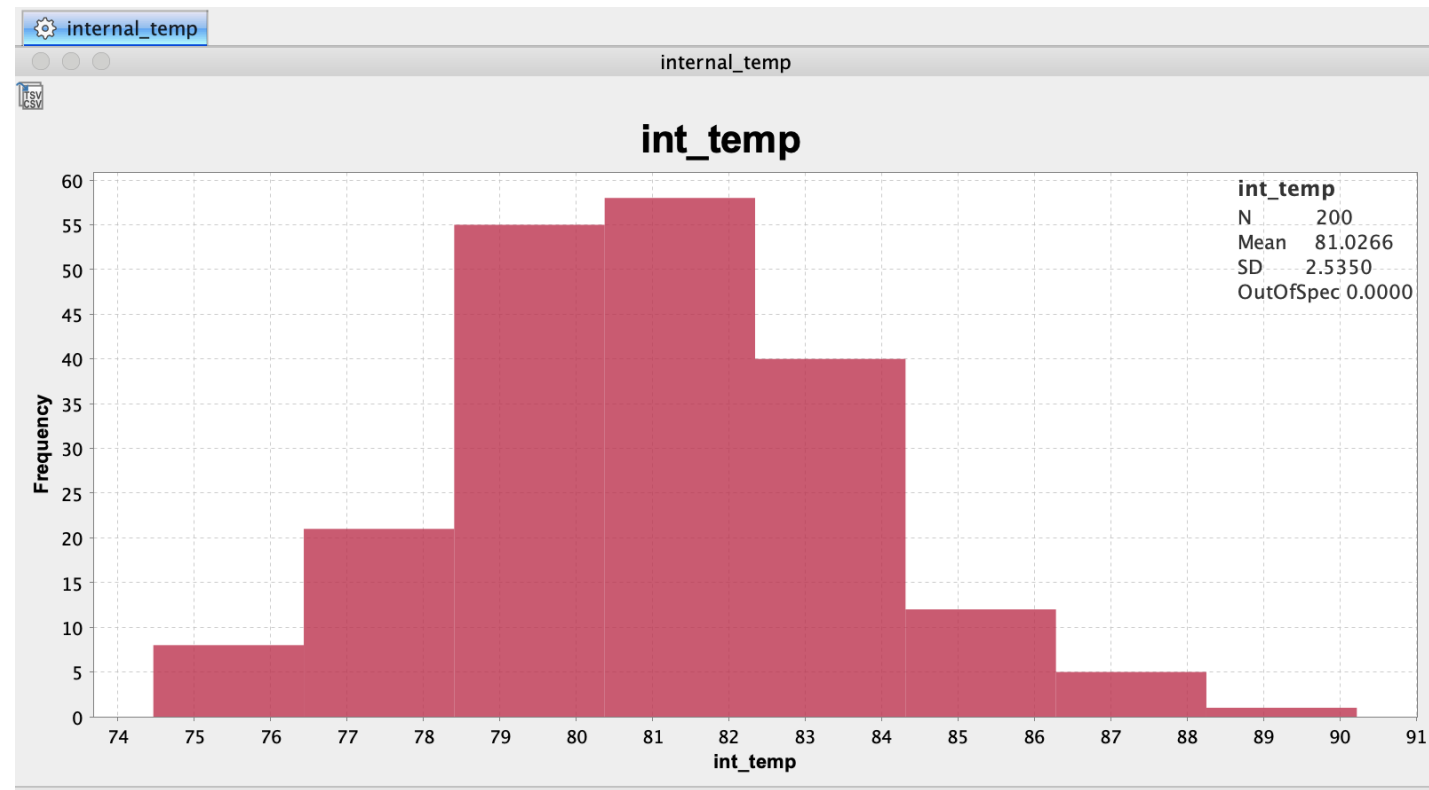
The screenshot displays the Simulink software interface. A blue arrow points to the 'mc\_sim' tab in the top toolbar. The main workspace shows the 'package Simulations[ Simulations ]' containing two components:

- «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



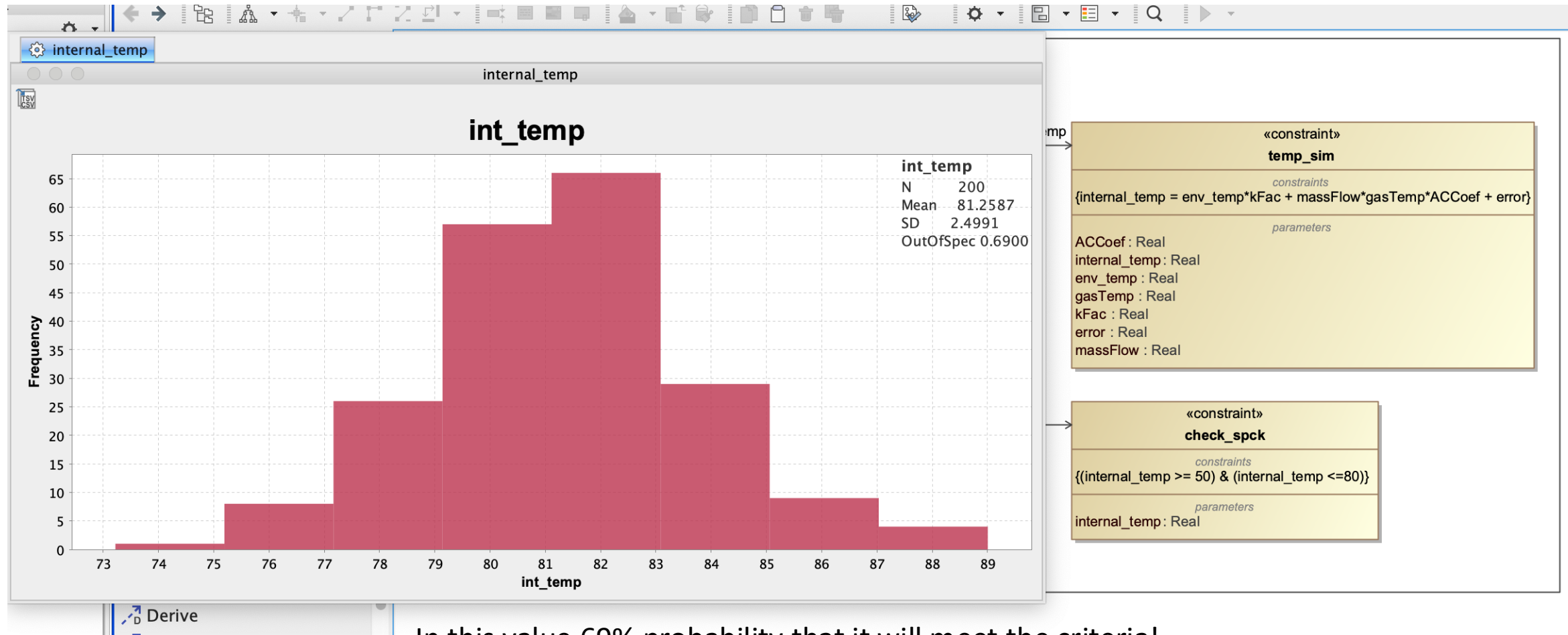
## 4. First Analysis

- 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..