Trade study analysis

A trade study or trade-off study is the activity of a multidisciplinary team to identify the most balanced technical solutions among a set of proposed viable solutions (System Engineering Manual, Federal Aviation Administration, 2006).

A trade study is used to compare with 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 “moe’s”) that corresponds to evaluation criteria and has a calculated value or value distribution. The moe’s for a given solution is 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.

Magic Model Analyst has built-in support for trade study analysis. The **TradeStudyExamples** sample model is used as a demonstration for trade study analysis through the following steps.

1. Use **TradeStudy «Block» «Analysis»** through the following step:
    - From the **MD Customization for SysML::analysis patterns::trade study** package, drag **TradeStudy «Block» «Analysis»** into the Block Definition diagram (BDD).

        Note

        The **TradeStudy** package is available in all SysML projects. If the **MD Customization for SysML** package is not visible, click ⚙ ▾ in the Containment tree pane and select **Show Auxiliary Resources**.

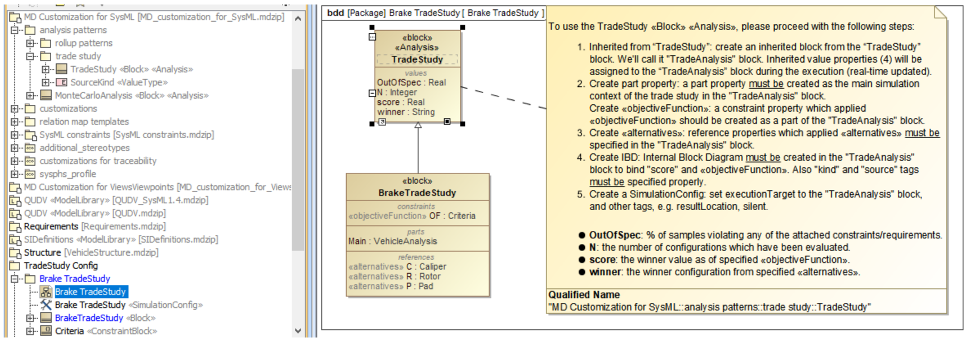
    The **TradeStudy** Block contains the following four value properties which will be inherited by the *TradeAnalysis* Block.

    - **OutOfSpec: Real**: counts how much percent of sample value that violates any attached constraints or Requirements.
    - **N: Integer**: the number of configurations evaluated.
    - **score: Real**: set as the **winner** value of the «objectiveFunction».
    - **winner: String**: used to keep the **winner** configuration as a string.
2. Create the *TradeAnalysis* Block by doing the following:
    a. Inherit the **TradeStudy** (Block).
    Create a new Block and Generalization from the **TradeStudy** Block so that the new *TradeAnalysis* Block is inherited, e.g., an instance of *BrakeTradeStudy*. Those four inherited value properties will be assigned to the *TradeAnalysis* Block and updated real-time during the execution.
    b. Add main simulation context as Part.
    Add the main Block of simulation context as a Part property of the *TradeAnalysis* Block, e.g., *Main : VehicleAnalysis*.
    c. Create **«objectiveFunction»**.
    An objective function is a special type of constraint Block used to determine the overall value of any weighted alternatives in terms of weighted criteria. You must create a constraint Block containing related constraint parameters and a constant expression for determining the best weighted value, e.g., *Criteria*.

        Note

        - The specification of the constraint must be an equation with LHS = RHS, where LHS contains only one parameter to bind with *TradeAnalysis::^score*, and RHS can contain multiple parameters (bound with corresponding value properties) to evaluate as **winner** criteria.
        - The output of **«objectiveFunction»** will be compared with previous weighted results. If an alternative is greater, it will be set as the **winner** (maximum value by default). However, if you want the minimum value, use a negative value, e.g., *value = -distance*.

    d. Add **«objectiveFunction»**.
    Create the constraint property in the *TradeAnalysis* Block typed by the created constraint Block, and apply **«objectiveFunction»**, e.g., *O F : Criteria*.
    e. Create **«alternatives»**.
    Create reference properties typed by a Block of alternatives and apply **«alternatives»** (*Qualified Name*: *MD Customization for SysML*::*additional_stereotypes*::*alternatives*) to those properties, e.g., *C : Caliper*, *R : Rotor*, and *P : Pad*, as shown in the figure below.


MD Customization for SysML [MD_customization_for_SysML.mdzip]
analysis patterns
rollup patterns
trade study
TradeStudy «Block» «Analysis»
SourceKind «ValueType»
MonteCarloAnalysis «Block» «Analysis»
customizations
relation map templates
SysML constraints [SysML constraints.mdzip]
additional_stereotypes
customizations for traceability
sysphs_profile
MD Customization for ViewsViewpoints [MD_customization_for_Views
QUDV «ModelLibrary» [QUDV_SysML1.4.mdzip]
QUDV «ModelLibrary» [QUDV.mdzip]
Requirements [Requirements.mdzip]
SIDefinitions «ModelLibrary» [SIDefinitions.mdzip]
Structure [VehicleStructure.mdzip]
TradeStudy Config
Brake TradeStudy
Brake TradeStudy
Brake TradeStudy «SimulationConfig»
BrakeTradeStudy «Block»
Criteria «ConstraintBlock»
bdd [Package] Brake TradeStudy [Brake TradeStudy]
«block» «Analysis» TradeStudy
values
OutOfSpec : Real
N : Integer
score : Real
winner : String
«block» BrakeTradeStudy
constraints
«objectiveFunction» OF : Criteria
parts
Main : VehicleAnalysis
references
«alternatives» C : Caliper
«alternatives» R : Rotor
«alternatives» P : Pad

To use the TradeStudy «Block» «Analysis», please proceed with the following steps:

1. Inherited from “TradeStudy”: create an inherited block from the “TradeStudy” block. We'll call it "TradeAnalysis" block. Inherited value properties (4) will be assigned to the “TradeAnalysis” block during the execution (real-time updated).
2. Create part property: a part property must be created as the main simulation context of the trade study in the "TradeAnalysis" block.
Create «objectiveFunction»: a constraint property which applied «objectiveFunction» should be created as a part of the "TradeAnalysis" block.
3. Create «alternatives»: reference properties which applied «alternatives» must be specified in the "TradeAnalysis" block.
4. Create IBD: Internal Block Diagram must be created in the "TradeAnalysis" block to bind "score" and «objectiveFunction». Also "kind" and "source" tags must be specified properly.
5. Create a SimulationConfig: set executionTarget to the "TradeAnalysis" block, and other tags, e.g. resultLocation, silent.

- **OutOfSpec**: % of samples violating any of the attached constraints/requirements.
- **N**: the number of configurations which have been evaluated.
- **score**: the winner value as of specified «objectiveFunction».
- **winner**: the winner configuration from specified «alternatives».

Qualified Name
"MD Customization for SysML::analysis patterns::trade study::TradeStudy"

Structure of the TradeAnalysis Block.

3. Create binding of the *TradeAnalysis* Block through the following steps:

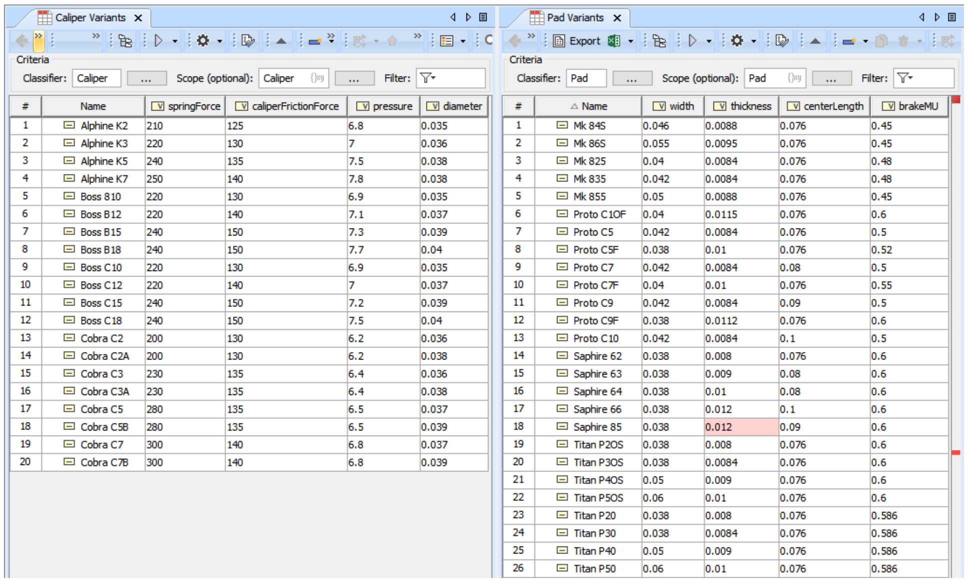
a. Create an Internal Block diagram (IBD).
Create an IBD of the *TradeAnalysis* Block. You must select Parts which have been set as **«objectiveFunction»** and **«alternatives»** to display. You must also display ^*score* (inherited property) in the diagram.

b. Bind **«objectiveFunction»**.
You must bind ^*score* with a Binding Connector to the LHS parameter of **«objectiveFunction»**, e.g., ^*score* – *value*. You must also bind a value property of the main simulation context with a Binding Connector to the RHS parameter of **«objectiveFunction»**, e.g., *stoppingDi stance* – *distance*.

c. Bind **«alternatives»**.
You must bind each **«alternatives»** with a Binding Connector to each Part property.

d. Set kind/source tags.
For each **«alternatives»**, **source** depends on the following **kinds** through the **Tags** settings.

- **kind = TABLE**: **source** must be an instance table which must be the same Classifier as the alternative property, as shown in the two figures below. However, sorting of rows in the table is not necessary.



Caliper Variants

Criteria
Classifier: Caliper   Scope (optional): Caliper   Filter:

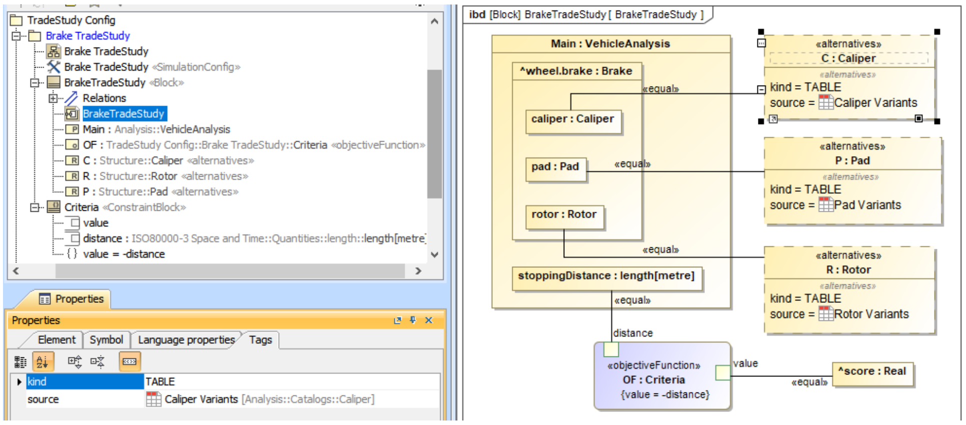
#	Name	springForce	caliperFrictionForce	pressure	diameter
1	Alphine K2	210	125	6.8	0.035
2	Alphine K3	220	130	7	0.036
3	Alphine K5	240	135	7.5	0.038
4	Alphine K7	250	140	7.8	0.038
5	Boss 810	220	130	6.9	0.035
6	Boss B12	220	140	7.1	0.037
7	Boss B15	240	150	7.3	0.039
8	Boss B18	240	150	7.7	0.04
9	Boss C10	220	130	6.9	0.035
10	Boss C12	220	140	7	0.037
11	Boss C15	240	150	7.2	0.039
12	Boss C18	240	150	7.5	0.04
13	Cobra C2	200	130	6.2	0.036
14	Cobra C2A	200	130	6.2	0.038
15	Cobra C3	230	135	6.4	0.036
16	Cobra C3A	230	135	6.4	0.038
17	Cobra C5	280	135	6.5	0.037
18	Cobra C5B	280	135	6.5	0.039
19	Cobra C7	300	140	6.8	0.037
20	Cobra C7B	300	140	6.8	0.039

Pad Variants

Criteria
Classifier: Pad   Scope (optional): Pad   Filter:

#	Name	width	thickness	centerLength	brakeMU
1	Mk 84S	0.046	0.0088	0.076	0.45
2	Mk 86S	0.055	0.0095	0.076	0.45
3	Mk 825	0.04	0.0084	0.076	0.48
4	Mk 835	0.042	0.0084	0.076	0.48
5	Mk 855	0.05	0.0088	0.076	0.45
6	Proto C1OF	0.04	0.0115	0.076	0.6
7	Proto C5	0.042	0.0084	0.076	0.5
8	Proto C5F	0.038	0.01	0.076	0.52
9	Proto C7	0.042	0.0084	0.08	0.5
10	Proto C7F	0.04	0.01	0.076	0.55
11	Proto C9	0.042	0.0084	0.09	0.5
12	Proto C9F	0.038	0.0112	0.076	0.6
13	Proto C10	0.042	0.0084	0.1	0.5
14	Saphire 62	0.038	0.008	0.076	0.6
15	Saphire 63	0.038	0.009	0.08	0.6
16	Saphire 64	0.038	0.01	0.08	0.6
17	Saphire 66	0.038	0.012	0.1	0.6
18	Saphire 85	0.038	0.012	0.09	0.6
19	Titan P2OS	0.038	0.008	0.076	0.6
20	Titan P3OS	0.038	0.0084	0.076	0.6
21	Titan P4OS	0.05	0.009	0.076	0.6
22	Titan P5OS	0.06	0.01	0.076	0.6
23	Titan P20	0.038	0.008	0.076	0.586
24	Titan P30	0.038	0.0084	0.076	0.586
25	Titan P40	0.05	0.009	0.076	0.586
26	Titan P50	0.06	0.01	0.076	0.586

TABLE, kind of «alternatives», must be the same Classifier as the alternative property.



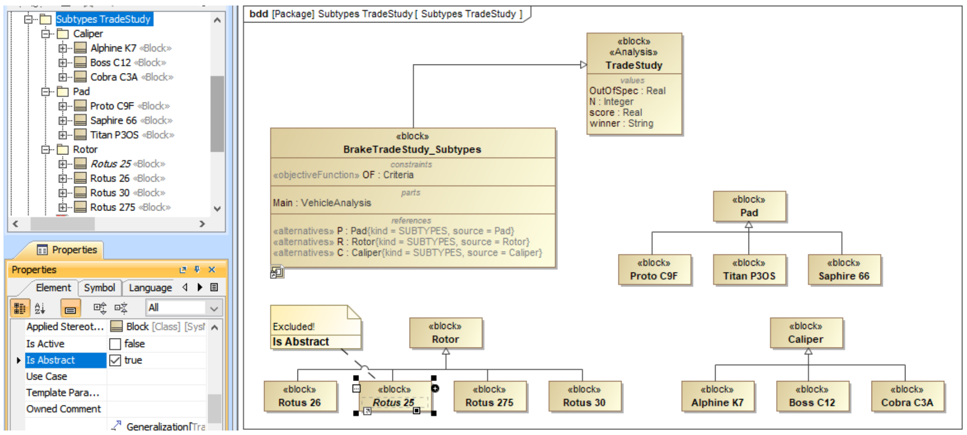
```
TradeStudy Config
  Brake TradeStudy
    Brake TradeStudy
    Brake TradeStudy «SimulationConfig»
    BrakeTradeStudy «Block»
      Relations
      BrakeTradeStudy
      Main : Analysis::VehicleAnalysis
      OF : TradeStudy Config::Brake TradeStudy::Criteria «objectiveFunction»
      C : Structure::Caliper «alternatives»
      R : Structure::Rotor «alternatives»
      P : Structure::Pad «alternatives»
    Criteria «ConstraintBlock»
      value
      distance : ISO80000-3 Space and Time::Quantities::length::length[metre]
      {} value = -distance
```

Properties
Element | Symbol | Language properties | Tags

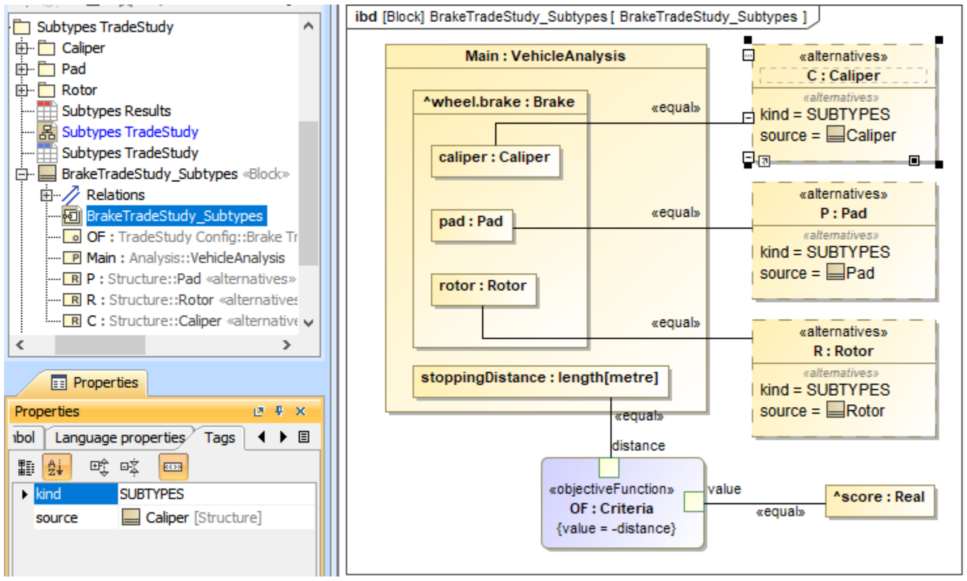
kind	TABLE
source	Caliper Variants [Analysis::Catalogs::Caliper]

Binding of the TradeAnalysis Block in the Internal Block diagram (kind = TABLE).

- **kind = SUBTYPES**: **source** must be a parent Block of subtypes which must be the same type as the alternative property, as shown in the two figures below. Also, the parent Block will not be evaluated as well as any Blocks which have *Is Abstract* = true.



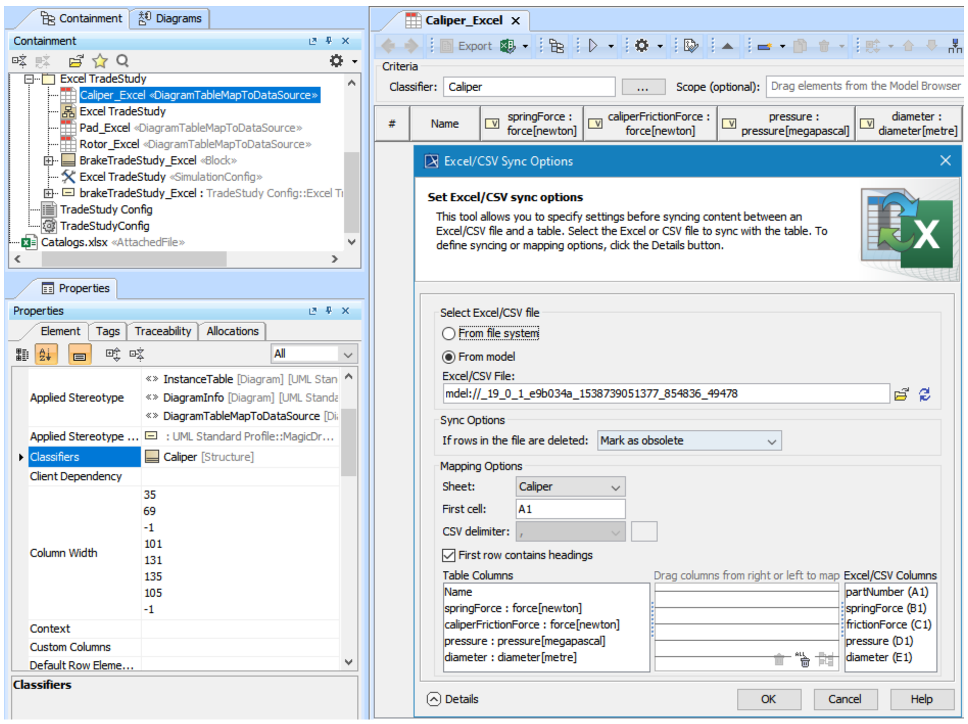
SUBTYPES, kind of «alternatives», must be the same type as the alternative property.



kind	SUBTYPES
source	Caliper [Structure]

Binding of the TradeAnalysis Block in the Internal Block diagram (kind = SUBTYPES).

- **kind = EXCEL**: **source** must be an instance table linked to an Excel file, and table columns must be mapped to Excel/CSV columns as shown in the two figures below (see also Sync with Excel or CSV files). However, you do not need to use the **Read from File** command from the **Publish** Excel toolbar to load data into the table.


Containment
Diagrams
Caliper_Excel
Export
Criteria
Classifier: Caliper
Scope (optional): Drag elements from the Model Browser
#
Name
springForce : force[newton]
caliperFrictionForce : force[newton]
pressure : pressure[megapascal]
diameter : diameter[metre]
Excel TradeStudy
Caliper_Excel «DiagramTableMapToDataSource»
Excel TradeStudy
Pad_Excel «DiagramTableMapToDataSource»
Rotor_Excel «DiagramTableMapToDataSource»
BrakeTradeStudy_Excel «Block»
Excel TradeStudy «SimulationConfig»
brakeTradeStudy_Excel : TradeStudy Config::Excel T
TradeStudy Config
TradeStudyConfig
Catalogs.xlsx «AttachedFile»
Properties
Element
Tags
Traceability
Allocations
All

Applied Stereotype	InstanceTable [Diagram] [UML Stan DiagramInfo [Diagram] [UML Standa DiagramTableMapToDataSource [Di
Applied Stereotype ...	: UML Standard Profile::MagicDr...
Classifiers	Caliper [Structure]
Client Dependency	
Column Width	35 69 -1 101 131 135 105 -1
Context	
Custom Columns	
Default Row Eleme...	

Classifiers

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]

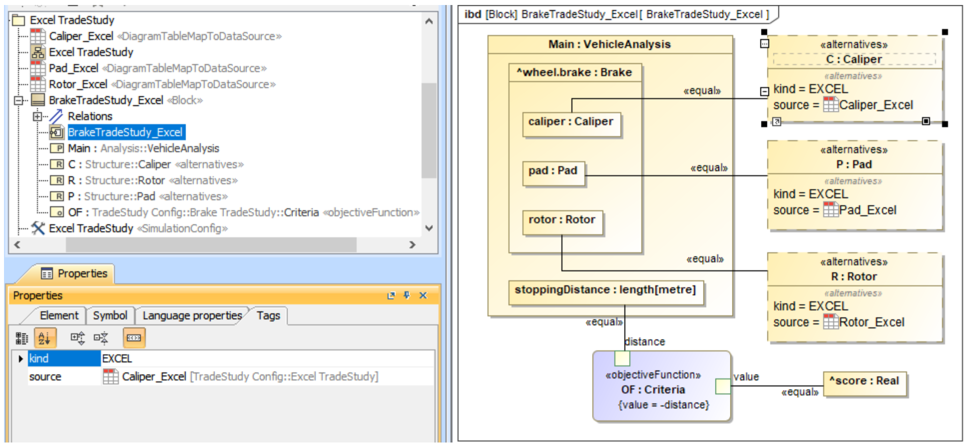
Excel/CSV File: mdl://_19_0_1_e9b034a_1538739051377_854836_49478

Sync Options
If rows in the file are deleted: Mark as obsolete

Mapping Options
Sheet: Caliper
First cell: A1
CSV delimiter: ,
[✓] First row contains headings

Table Columns	Excel/CSV Columns
Name	partNumber (A1)
springForce : force[newton]	springForce (B1)
caliperFrictionForce : force[newton]	frictionForce (C1)
pressure : pressure[megapascal]	pressure (D1)
diameter : diameter[metre]	diameter (E1)

EXCEL, kind of «alternatives», must be linked to an Excel file, and table columns must be mapped to Excel/CSV columns.



kind	EXCEL
source	Caliper_Excel [TradeStudy Config::Excel TradeStudy]

Binding of the TradeAnalysis Block in the Internal Block diagram (kind = EXCEL).

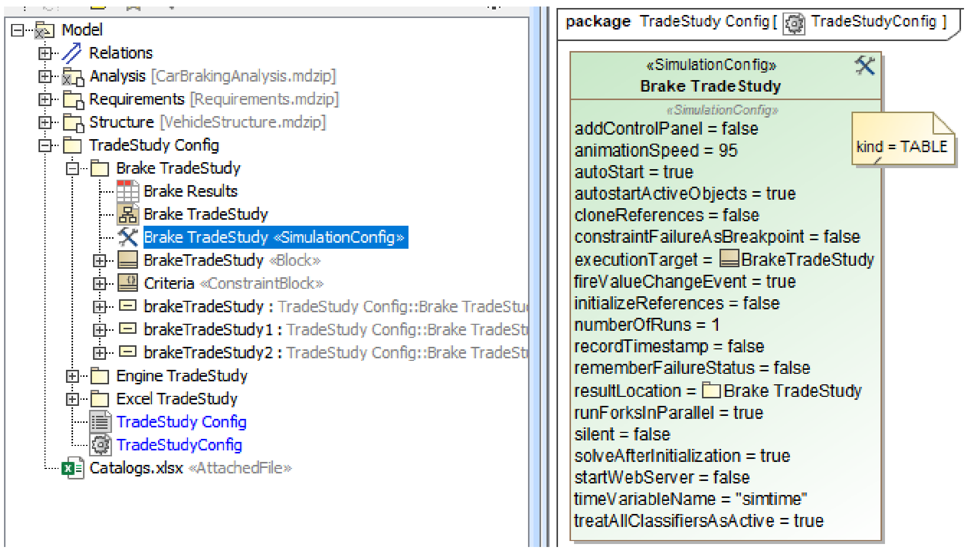
4. Create SimulationConfig and other settings.
   You can create a SimulationConfig, so that it can specify the **resultLocation** of this **TradeStudy** to a package, e.g., instances of the winning configuration will be saved. Also, the following tags can be set, as shown in the figure below.
   a. **executionTarget**
      executionTarget must be set to the *TradeAnalysis* Block, e.g., *BrakeTradeStudy* in the sample. *executionTarget* can be an instance of the *TradeAnalysis* Block so that the instance can be reconfigured.
   b. **resultLocation**
      A package/instance table must be specified so an instance with related information will be saved after running the simulation. The information includes **N**, **OutOfSpec**, **score**, **winner**, and other elements.

      Note

      - If you do not create a SimulationConfig and run the *TradeAnalysis* Block directly, **TradeStudy** will not be triggered, but the *TradeAnalysis* Block will be run as normal .
      - If the **executionTarget** of a SimulationConfig is the *TradeAnalysis* Block, **TradeStudy** execution will override other executions, so Monte Carlo simulation will not be triggered even if **numberOfRuns** is more than 1.

   c. **silent**
      You can set the **silent** tag accordingly to see the animation.
   d. **rememberFailureStatus**
      You can use *rememberFailureStatus* when evaluating those weighted alternatives. Any alternatives violating any of the attached constraints/Requirements will not be considered as the **winner**, but they will be used in the calculation of **OutOfSpec**.
   e. Tags neglected

You can neglect **animationSpeed**, **constraintFailureAsBreakpoint**, **UI**, and **autoStart**.



```
Model
  Relations
  Analysis [CarBrakingAnalysis.mdzip]
  Requirements [Requirements.mdzip]
  Structure [VehicleStructure.mdzip]
  TradeStudy Config
    Brake TradeStudy
      Brake Results
      Brake TradeStudy
      Brake TradeStudy «SimulationConfig»
      BrakeTradeStudy «Block»
      Criteria «ConstraintBlock»
      brakeTradeStudy : TradeStudy Config::Brake TradeStu
      brakeTradeStudy1 : TradeStudy Config::Brake TradeSt
      brakeTradeStudy2 : TradeStudy Config::Brake TradeSt
    Engine TradeStudy
    Excel TradeStudy
    TradeStudy Config
    TradeStudyConfig
  Catalogs.xlsx «AttachedFile»
```

package TradeStudy Config [TradeStudyConfig]

«SimulationConfig»
Brake TradeStudy

«SimulationConfig»
addControlPanel = false
animationSpeed = 95
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionTarget = BrakeTradeStudy
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1
recordTimestamp = false
rememberFailureStatus = false
resultLocation = Brake TradeStudy
runForksInParallel = true
silent = false
solveAfterInitialization = true
startWebServer = false
timeVariableName = "simtime"
treatAllClassifiersAsActive = true

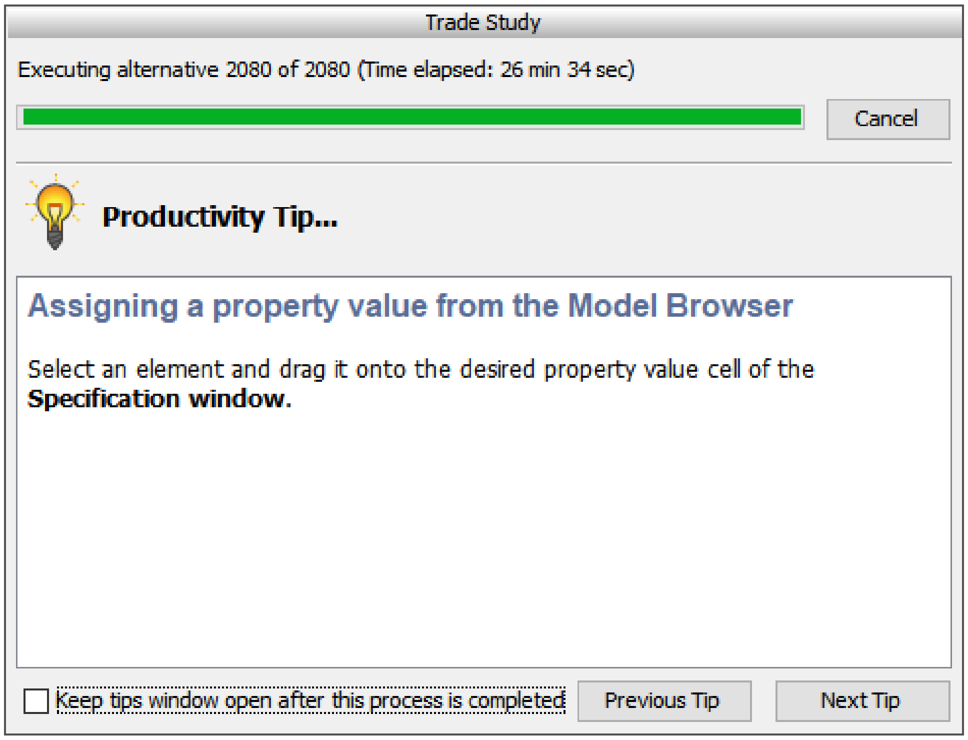
kind = TABLE

SimulationConfig created for other settings, e.g., executionTarget and resultLocation.

5. Run the SimulationConfig.
When running the SimulationConfig, the information of **TradeStudy** will be printed in the **Console** pane. The Simulation pane will be disabled, but all warning/errors will be printed. There is a progress bar shown with the description of **Executing alternative [n] of [total iterations] (Time elapsed: [time])**, and the **Cancel** button that allows canceling **Trade Study** as shown in the figure below. Simulation Toolkit automatically iterates all variants and instantiates all possible configurations in memory. For example, *Brake* which contains the combinations of *Pad* (26 instances), *Caliper* (20 instances), and *Rotor* (4 instances) will have 26 x 20 x 4 = 2,080 configurations. The **winner** value on each iteration will be compared directly with the **score** value property.

Note

- The **Starting Math Engine** progress bar will be shown in this sample because MATLAB is used as the external evaluator. See also Integration with MATLAB.
- Any alternatives violating any of the attached constraints/Requirements will not be considered as the **winner**. They will be used in the calculation of **OutOfSpec**.
- **rememberFailureStatus** of a SimulationConfig will be used when evaluating those alternatives.
- In some cases, you can click **Unlock** in the **Simulation** pane to see execution details during the execution.



Trade Study

Executing alternative 2080 of 2080 (Time elapsed: 26 min 34 sec)

Productivity Tip...

Assigning a property value from the Model Browser

Select an element and drag it onto the desired property value cell of the **Specification window.**

A progress bar shown with description and the Cancel button during the simulation.

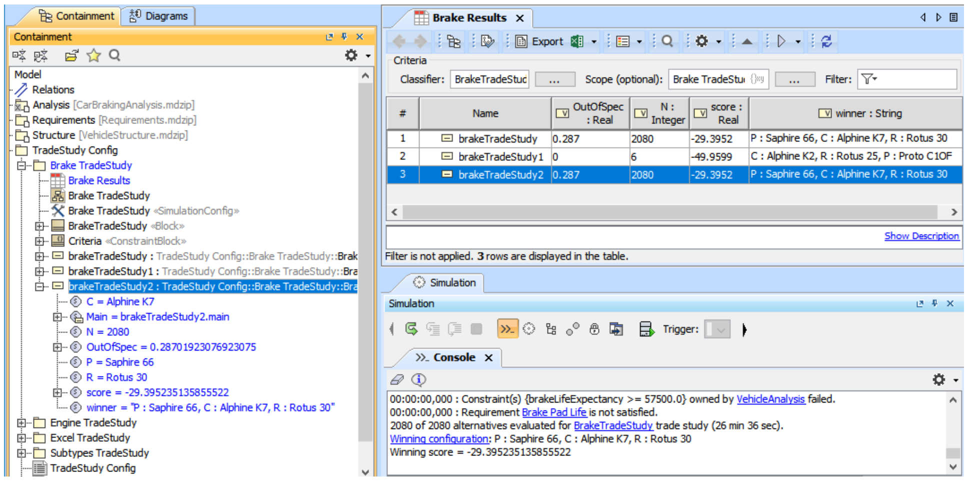
6. Get the **TradeAnalysis** result.

   When the simulation is either completed or canceled, **winning** information will be printed on the **Console** pane in the following three lines as shown in the figure below.

   - The first line shows the number of iterations (completed/canceled) of all alternatives for ***executionTarget*** with elapsed time.
   - The second line displays the *winning* configuration from the **winner** string.
   - The third line is the **winning score** from the ^*score*.

   > **Note**
   >
   > The **winner** string is printed with the formats as follows:
   > *AlternativeProperty.Name1 : StringKind [, AlternativeProperty.Name2 : StringKind, AlternativeProperty.Name3 : StringKind, …]*
   >
   > where *StringKind* will apply the following rules, depending on **kind** of alternative:
   >
   > - *kind=TABLE*: then *StringKind*=InstanceName, e.g., P : Saphire 66, C : Alphine K7, R : Rotus 30.
   > - *kind=SUBTYPES*: then *StringKind*=subtypesName, e.g., power : Diesel, support : Wheels, stopping : Brakes.
   > - *kind=EXCEL*: then *StringKind*=#Row, e.g., R : #5, P : #21, C : #21, where *Row* is the number of Excel rows.

   The result instance will be saved at the location as specified in **resultLocation** of SimulationConfig. You can create an instance table, set a Classifier to the *TradeAnalysis* Block, and set *Scope* to the package of the results.



```
Model
  Relations
  Analysis [CarBrakingAnalysis.mdzip]
  Requirements [Requirements.mdzip]
  Structure [VehicleStructure.mdzip]
  TradeStudy Config
    Brake TradeStudy
      Brake Results
      Brake TradeStudy
      Brake TradeStudy «SimulationConfig»
      BrakeTradeStudy «Block»
      Criteria «ConstraintBlock»
      brakeTradeStudy : TradeStudy Config::Brake TradeStudy::Brak
      brakeTradeStudy1 : TradeStudy Config::Brake TradeStudy::Bra
      brakeTradeStudy2 : TradeStudy Config::Brake TradeStudy::Bra
        C = Alphine K7
        Main = brakeTradeStudy2.main
        N = 2080
        OutOfSpec = 0.287019230769230 75
        P = Saphire 66
        R = Rotus 30
        score = -29.395235135855522
        winner = "P : Saphire 66, C : Alphine K7, R : Rotus 30"
    Engine TradeStudy
    Excel TradeStudy
    Subtypes TradeStudy
    TradeStudy Config
```

Brake Results

Criteria — Classifier: BrakeTradeStud | Scope (optional): Brake TradeStu | Filter:

#	Name	OutOfSpec : Real	N : Integer	score : Real	winner : String
1	brakeTradeStudy	0.287	2080	-29.3952	P : Saphire 66, C : Alphine K7, R : Rotus 30
2	brakeTradeStudy1	0	6	-49.9599	C : Alphine K2, R : Rotus 25, P : Proto C1OF
3	brakeTradeStudy2	0.287	2080	-29.3952	P : Saphire 66, C : Alphine K7, R : Rotus 30

Filter is not applied. **3** rows are displayed in the table.

```
00:00:00,000 : Constraint(s) {brakeLifeExpectancy >= 57500.0} owned by VehicleAnalysis failed.
00:00:00,000 : Requirement Brake Pad Life is not satisfied.
2080 of 2080 alternatives evaluated for BrakeTradeStudy trade study (26 min 36 sec).
Winning configuration: P : Saphire 66, C : Alphine K7, R : Rotus 30
Winning score = -29.395235135855522
```

The TradeAnalysis result (in the last 3 lines) is printed on the Console pane, and the result instance is saved into a package and presented in the instance table.