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.

Cameo Simulation Toolkit 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::ad ditional_stereotypes::alternatives) to those properties, e.g., C : Caliper, R : Rotor, and P : Pad, as shown in the figure below.



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 **\cdot 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 **\cdot 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.

0							-					_
	🗋 Caliper Variants 🗙	<			∢ ▷ 🗉		Pad Variants 🗙				4 ⊳	Ξ
- ×	* * Be	D - 0 -	🛛 🕞 🔺 📄 🛶 🏹	殿 - 合 、	» 🗉 - 🛛 С	🔶 ×	Export 📳 🔹	188 D	- Q - 1	D) 🔺 🛶 •	m + - 11	
Crite	ria					Criter	ia					
Clas	sifier: Caliper	Scope (op	tional): Caliper 🛞	Filter:	7.	Clas	sifier: Pad	Scope (o	ptional): Pad	()og F	ilter: 🕎	
				1		-					1	
#	Name	Spring⊢orce	V caliper⊢riction⊢orce	I pressure	<u></u> diameter	#	△ Name	width	I thickness	CenterLength	U brakeMU	
1	Alphine K2	210	125	6.8	0.035	1	Mk 84S	0.046	0.0088	0.076	0.45	-
2	Alphine K3	220	130	7	0.036	2	Mk 86S	0.055	0.0095	0.076	0.45	-11
3	Alphine K5	240	135	7.5	0.038	3	Mk 825	0.04	0.0084	0.076	0.48	-11
4	Alphine K7	250	140	7.8	0.038	4	Mk 835	0.042	0.0084	0.076	0.48	-11
5	Boss 810	220	130	6.9	0.035	5	Mk 855	0.05	0.0088	0.076	0.45	
6	Boss B12	220	140	7.1	0.037	6	Proto C10F	0.04	0.0115	0.076	0.6	
7	Boss B15	240	150	7.3	0.039	7	Proto C5	0.042	0.0084	0.076	0.5	
8	Boss B18	240	150	7.7	0.04	8	Proto C5F	0.038	0.01	0.076	0.52	
9	Boss C10	220	130	6.9	0.035	9	Proto C7	0.042	0.0084	0.08	0.5	
10	Boss C12	220	140	7	0.037	10	Proto C7F	0.04	0.01	0.076	0.55	
11	Boss C15	240	150	7.2	0.039	11	Proto C9	0.042	0.0084	0.09	0.5	1
12	Boss C18	240	150	7.5	0.04	12	Proto C9F	0.038	0.0112	0.076	0.6	
13	Cobra C2	200	130	6.2	0.036	13	Proto C10	0.042	0.0084	0.1	0.5	
14	Cobra C2A	200	130	6.2	0.038	14	Saphire 62	0.038	0.008	0.076	0.6	1
15	Cobra C3	230	135	6.4	0.036	15	Saphire 63	0.038	0.009	0.08	0.6	
16	Cobra C3A	230	135	6.4	0.038	16	Saphire 64	0.038	0.01	0.08	0.6	
17	Cobra C5	280	135	6.5	0.037	17	Saphire 66	0.038	0.012	0.1	0.6	1
18	Cobra C5B	280	135	6.5	0.039	18	Saphire 85	0.038	0.012	0.09	0.6	1
19	Cobra C7	300	140	6.8	0.037	19	Titan P2OS	0.038	0.008	0.076	0.6	1
20	Cobra C78	300	140	6.8	0.039	20	Titan P3OS	0.038	0.0084	0.076	0.6	
						21	Titan P40S	0.05	0.009	0.076	0.6	
						22	Titan P50S	0.06	0.01	0.076	0.6	
						23	Titan P20	0.038	0.008	0.076	0.586	1
						24	Titan P30	0.038	0.0084	0.076	0.586	1
						25	Titan P40	0.05	0.009	0.076	0.586	1
						26	Titan P50	0.06	0.01	0.076	0.586	1

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



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.



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.

Be Containment	^{≿0} Diagrams		Caliper_E	xcel X					
Containment	.≊ ∓ ×		Evo	vrt 🕅 🖬 🚽 🗄 🗛 🕴	N - EA - EB				
¤⊈ ⊯ 🛱 👉 Q	Ċ.	Critor					i : ∨ n'n		
Excel TradeStu Caliper_Excel Trade	dy , cel «DiagramTableMapToDataSource» eStudy	Clas	a sifier: Calipe	springForce :	Scop	pe (optional): Drag eleme	nts from the Model Browser		
Pad_Excel Rotor_Exce Br BrakeTrade	«DiagramTableMapToDataSource» al «DiagramTableMapToDataSource» 2Study_Excel «Block» 2Study «SimulationConfig»	#	Name	force[newton]	force[newton]	pressure[megapa	scal] diameter [metre]		
TradeStudy Co GradeStudyCo Catalogs.xlsx «Atta Catalogs.xlsx «Atta	study_Excel : TradeStudy Config::Excel Ti nfig achedFile> >		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.						
Properties		_							
Properties	2 # X		Select E	xcel/CSV file					
Element Tags	Traceability Allocations		OFrom	n file system					
	All 🗸	1	From	n model					
	(*) InstanceTable [Disgram] [] [M] Stan	i II	Excel/C	SV File:					
Applied Charachura	W Discarce rable [Diagram] [UML Stands		mdel:/	19 0 1 e9b034a	1538739051377 85483	6 49478	2		
Applied Steleotype	WagramTableManToDataSource [Di:				-	-			
Applied Stereotype	IMI Standard Profile: MagicDr		Sync Op	tions					
Classifiers	Caliner [Structure]		If rows	in the file are deleter	d: Mark as obsolete	\sim			
Client Dependency			Mapping	Options					
clience bependency	35		Sheet:	Caliper	\sim				
	69		First ce	l: A1					
-1			CSV de	imiter:	~				
Column Width	101		Fire	row contains heading					
Column mour	131				iys	c			
	135		Table C	olumns	Drag colu	umns from right or left to m	ap Excel/CSV Columns		
	-1		springF	orce : force[newton]			springEorce (B1)		
Contoxt			caliperi	rictionForce : force	newton1		frictionForce (C1)		
Outern Columna			pressu	e : pressure[megapa	ascal]		pressure (D1)		
Default Pow Fleme			diamet	er : diameter[metre]			diameter (E1)		
Classifiers		i II							
			 Details 			ОК	Cancel Help		

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



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.

A 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.



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), *Cali per* (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.

⚠	ote	
	 The Starting Math Engine progress bar will be shown in this sample because MATLAB is used as the external esternal esternation with MATLAB. 	evaluator.
	 Any alternatives violating any of the attached constraints/Requirements will not be considered as the winner. Th used in the calculation of OutOfSpec. 	ey will be
	 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)							
	Cancel						
	Concer						
Productivity Tip							
Assigning a property value from the Model Browser	Assigning a property value from the Model Browser						
Select an element and drag it onto the desired property value cell of the Specification window.							
Keep tips window open after this process is completed Previous Tip	ext Tip						

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 A 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.

E Containment	Brake Results X 4 D E								
Containment C V X									
咚 踩 🗎 🏷 Q 🛛 🗘 🗘 🗘 🔹		Criteria							
Model	Classifier: BrakeTradeStud								
- 🥢 Relations									
- 🔂 Analysis [CarBrakingAnalysis.mdzip]	#	Name	OutOfSpec	N:	score :	W winner - String			
- 🔁 Requirements [Requirements.mdzip]		- Horne	:Real	Integer	Real	La Winch : Soling			
- 🔁 Structure [VehicleStructure.mdzip]	1	brakeTradeStudy	0.287	2080	-29.3952	P : Saphire 66, C : Alphine K7, R : Rotus 30			
- 🛄 TradeStudy Config	2	brakeTradeStudv1	0	6	-49,9599	C : Alphine K2, R : Rotus 25, P : Proto C1OF			
Brake TradeStudy	3	brakeTradeStudy2	0.287	2080	-29 3952	P : Saphire 66, C : Alphine K7, R : Rotus 30			
Brake Results			01207	2000	2510502				
	<					>			
BrakeTradeStudy «Block»						Show Description			
		Class is not posited. 2 years are displayed in the table							
Im- Image: BrakeTradeStudy : TradeStudy Config::Brake TradeStudy::Brak		not applied. 3 rows are displ	ayed in the table	•					
brakeTradeStudy1 : TradeStudy Config::Brake TradeStudy::Bra		Simulation							
E brakeTradeStudy2 : TradeStudy Config::Brake TradeStudy::Bra									
C = Alphine K7		ion				2 4 X			
H- G Main = brake tradestudy2.main	(🕼 🗐 📜 😕 📀 😫 💊° 🖱 🖾 🚍 Trigger: 🔽)								
B = Sachia 65	». Console x								
P = Saprile 60 P = Patrix 20		、							
E () corre = -30 205325125955533	BQ	2				Q •			
Score = -23.33223133033222 winner = "D : Sanhire 66. C : Alnhine K7. D : Potus 30"	00:00:	00,000 : Constraint(s) {brak	eLifeExpectancy	>= 57500.0}	owned by Ve	ehicleAnalysis failed.			
Engine TradeStudy	00:00:	00,000 : Requirement Brake	Pad Life is not sa	tisfied.	. I (0c	27)			
E- Excel TradeStudy		2080 of 2080 alternatives evaluated for <u>brake iradestudy</u> trade study (26 min 36 sec).							
E-California Code Study		Winning correge = -29,395235135855522							
TradeStudy Config						*			

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.