What's new in SysML plugin 18.1 FR

Released on: January 15, 2015

More Efficient Ports and Flows Modeling

Direction of Proxy Ports

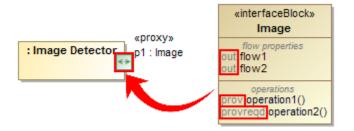
Review all owned and inherited flow properties and directed features of the Proxy Port using new **Direction** property! You can find it in the Specification window of the Proxy Port.

K Specification of Proxy Port ready	
	erties oxy Port in the properties specification ns from the Properties drop-down list to
🗉 te 🕫	ready
pout +ready : 03-Physical Model::3- p Documentation/Hyperlinks Documentation/Hyperlinks Documentation/Hyperlinks	2↓ ■ ₽‡ ₽ż abc Properties: All
Provided/Required Interfaces Definition Inner Elements Definitions	Proxy Port Direction Out [SysML::Ports&Flow]

The directed features are not included into combined direction of the Proxy Port by default. To include it set the **Include Directed Features into Combined Direction of Proxy Port** property value to *true* in the **Project Options** dialog.

If all features have direction "out" or "provided", the combined direction is "out". If all features have direction "in" or "required", the combined direction is "in". Otherwise the direction is "inout".

The Direction property is available in tables, matrices, criteria selection dialogs, reports, and open API.



Direction prefixes

Now directions of flow properties and Proxy Ports can be easily identified in many places with help of direction prefixes.

The flow property direction prefixes are now displayed:

- In compartment area of Part (1). To hide flow property direction prefix on the Part, set the Show Flow Property Direction in Compartment p roperty value to *true* in the Symbol Properties dialog of that Part.
- In the Model Browser (2).

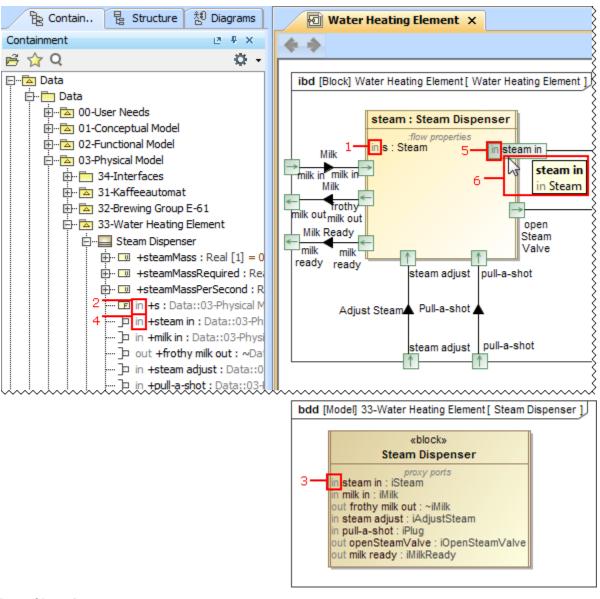
The Proxy Port direction prefixes are now displayed:

• In compartment area of Block (3). To hide the Proxy Port direction prefixes on the Block, set the **Show Proxy Port Direction in Compartment** property value to *false* in the **Symbol Properties** dialog of that Block.

The Proxy and Full Port direction prefixes are now displayed:

• In the Model Browser (4).

- On the Port shape when its name is displayed inside the shape (5). For this, open the Symbol Properties dialog of the Proxy or Full Port and select Name and Type Labels Inside or All Labels Inside as the Position of Labels property value. To hide the direction prefix on the Port shape, set the Show Direction Prefix Inside Port property value to *false* in the Symbol Properties dialog of that Port.
- On the ToolTip which opens when you move the pointer over the Proxy or Full Port or its name (6).



New Ports/ Interfaces property group

All owned and inherited Ports and their Interfaces are now listed in one place! Manage them on the left of the Block Specification window > Ports /Interfaces. For detailed information about selected Port interfaces open its Specification window > Interface Block Properties.

Block properties The Port/Interfaces no decontains a list of Block owned ports. Create or delete ports. Use the ports genicitation button to edit properties of a specific port. Image: Section of the port of the ports of a specific port. Image: Section of ports / Interfaces Image: Section of proxy Port Image: Section of proxy Port Image: Section of proxy Port Image: Section of proxy Port coffee beans out Image: Section of Proxy Port porties Image: Section of Proxy Port porties of a specific attribute. Image: Section of Proxy Port porties Image: Section of Proxy Port porties Image: Section of Proxy Port porti	K Specification of Block Bean Ho	opper			×	
Bean Hopper Documentation/Hyperinki Direction Proxy Port Proxy Port Proxy Port Proy Port Type Properties Direction Port Name Port Type Properties Properties Properties Operations Relations Allocations Instances Instances Create Redefine Delete Delete Delete Descriftation Default Value Owner Proverties Directions Interface Block Properties Name Type Default Value Owner Proverties Directions Direction button to edit properties Directions Directions Direction button to edit properties of a specific attribute. Directions Directions Directions Directions Directions Directions Directions Direction button to edit properties Directions Directions Directions Directions Directions Direction button to edit properties of a specific attribute. Directions Directions Directions Directions Directions Directions Direction button to edit properties of a specific attribute. Default Value Owner Directions	The Port/Interfaces node contain		elete ports. Use the por	ts		
Bean Hopper Bean Hopper Documentation/Hyperinki Direction Direction Direction Direction Proxy Port Direction Direction Proy Port Direction <pd< td=""><td>• • • •</td><td>Ports/Interfaces</td><td></td><td></td><td></td></pd<>	• • • •	Ports/Interfaces				
Outline flaudurfryge inso Outline flaudurfryge inso Operations Oper	Bean Hopper					
Constraints Constrain	Usage in Diagrams		Port Type	Type Features		
Specification of Proxy Port coffee beans out Create Redefine Delete Cost offee beans out := Coffee beans out := Coffee Data:: := out c : Coffee Detations Alocations Instance: Create Redefine Delete Cost of Proxy Port coffee beans out Specification of Proxy Port coffee beans out Interface Block Properties Inc :: Coffee Data:::0: :=	Constraints	Proxy Port				
Operations Pour coffee beans out Coffee (Data:: Content of Proxy Port coffee beans out Cose Back Eorward Help Cose Eorward Eorward Help Cose Eorward Eorw			■ iCoffee [Data::	⊡ in c : Coffee		
Allocations Instances Instances Instances Incerification of Proxy Port coffee beans out Close Back Eorward Help Interface Block Properties. Create or delete properties. Use the properties Delete Interface Block Properties Interface Bloc	Derations Behaviors	~ out coffee beans out	iCoffee [Data::	E out c : Coffee		
Create Redefine Delete Close Back Corward Help Specification of Proxy Port coffee beans out Coxy Port properties The Properties node contains a list of Proxy Port properties. Create or delete properties. Use the properties Specification button to edit properties of a specific attribute. Interface Block Properties Out coffee beans out : ~4Coffee Usage in Diagrams Connectors Provided/Required Interfaces Provided/Required Interfaces Provided/Required Interfaces Provided/Required Interfaces Provided/Required Interfaces Inc Coffee Data::03 Flow Properties In c Coffee Data::03 Inc Coffee Data::03 Inc Coffee Data::03 Inc Coffee Data::03 Inc Coffee Data::03	🗈 Allocations					
Specification of Proxy Port coffee beans out			Create Redef	ine Delete		
Proxy Port properties The Properties node contains a list of Proxy Port properties. Create or delete properties. Use the properties specification button to edit properties of a specific attribute. Image: Specific attribute Image: Spe		Close	Back	ward Help		
Proxy Port properties The Properties node contains a list of Proxy Port properties. Create or delete properties. Use the properties specification button to edit properties of a specific attribute. Image: Specific attribute Image: Spe						
Proxy Port properties The Properties node contains a list of Proxy Port properties. Create or delete properties. Use the properties specification button to edit properties of a specific attribute. Image: Specific attribute Image: Specific attribute Image: Specific attribute Image: Spec	Specification of Proxy Port cof	ffee beans out			x	
 □ out coffee beans out : ~iCoffee □ Usage in Diagrams □ Connectors □ Provided/Required Interfaces □ Provided/Required Interfaces □ Inner Elements □ Relations □ Tags □ Constraints □ Interface Block Properties □ Up Down Create Redefine Delete 	Proxy Port properties The Properties node contains a list of Proxy Port properties. Create or delete properties. Use the properties					
Image: in Diagrams Image: Connectors Provided/Required Interfaces Image: Default Value Inner Elements Image: Default Value Owner Flow Properties Image: Default Value Owner Image: Default Value Image: Default Va	🗉 🖻 🔁	Interface Block Properties	5			
Provided/Required Interfaces Inner Elements Inner Elements Interface Block Properties Interface Block Properties Interface Block Properties Up Down Create Redefine Delete		e ∰ 2∔ ⊟∔ ⊡‡ ⊡≵ abc				
Imer Denents Imer Denents Imer Denents In c Constraints Imer face Block Properties Imer face Block Properties Up Down Create Redefine Delete	Provided/Required Interface	es	Default Value	Owner		
Interface Block Properties Up Down Create Redefine Delete	Relations		:03	📑 iCoffee [Data:	::0 🗊	
Image:	Constraints					
Close Back Eorward Help		Up Down	Create	Redefine De	elete	
		Close	<u>B</u> ack	Eorward	Help	

Validation of Ports compatibility

Validation of Ports is more efficient now! It is easier to create the system with compatible connections between Ports because of additional validation of Ports compatibility rules. They check if:

- Flow is possible
 Flow Properties and directed features match
 Item Flow or Item Property type is compatible with the Interface Flow Property type

Invalid connectors are highlighted in red. Click 🥝 on the smart manipulator toolbar and choose one of the suggested solutions.

ibd [Block] Water Heating Element [Water Heating Element] Water : Rotary Pump Water Water : Heating Element] Cold cold in cold cold water in cold cold	
Milk steam : Steam Dispenser Milk milk in Milk in milk frothy out milk out	Change "steam in" type to conjugated "iAdjustWater" Change "steam out" type to conjugated "iSteam" Ignore Select in Validation Results



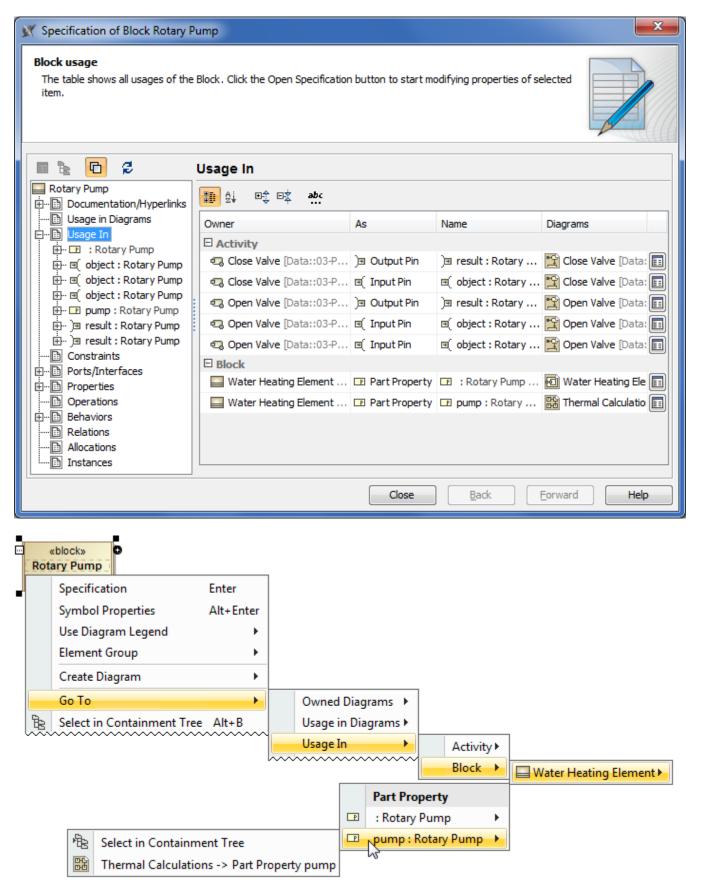
Element usage traceability

The traceability of the Block, Signal, and Instance elements is much easier now! There are two ways to inspect those element usages in your model:

- For the Block and Signal elements:
- On the left of the Block or Signal Specification window, click Usage In.
- From shortcut menu of the Block or Signal, choose Go To > Usage In.
 - For the Instance element:
- On the left of the Block Specification window, click Instances.
- From shortcut menu of the Block, choose Go To > Instances.

Block usage traceability

- Part
- Instance
- Partition
- Lifeline
- Reference



Signal usage traceability

You can trace the Signal when it is used:

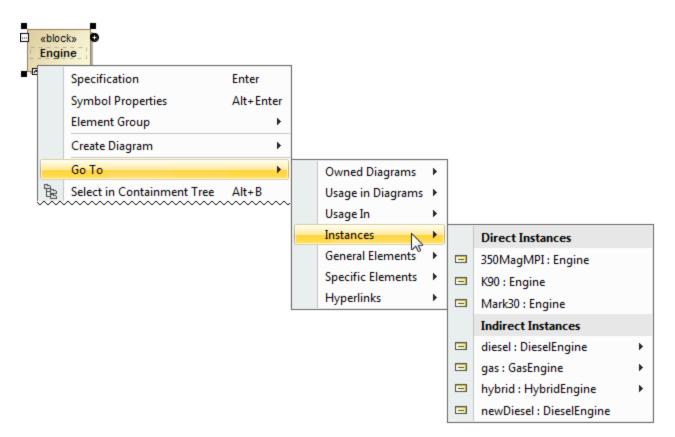
- In Trigger on Transitions or Accept Event Actions
 As Item Property type
 As Flow Property type
 In Send Signal Action or Broadcast Action
 In Sequence message

Specification of Signal Pump Or Signal usages as type. This table shows all usages of the s				
🗉 te To 💋	Usage In			
Pump On	₩ 2↓ № № 40	ć		
Usage in Diagrams	Owner	As	Name	Diagrams
Template Parameters Inner Elements Relations	Activity Start Pumping [0 Interface Block	Send Signal Action	Pump On [03	😭 Start Pumping [0 🗐
Usage In 	Pump [03-Physic	E Flow Property	E out +pump: 0	
⊕····E temFlow:flow for Pump ⊕·· D Pump On ⊕·· / ³ Transition:Pump On[03-F	A 34-Water Heatin	- ItemFlow	ItemFlow:flow	된 Water Heating E 📰
Tags	Rotary Pump [03	7 Transition	7 Transition:Pu	🛱 Rotary Pump [03 📰
Constraints Traceability Allocations				Open Diagram Open Specification Select in Containment Tre
				Go To
]			
Create Relation				
	-			
Element Group				
Go To	Usage in Diagrams	•		
Open in New Tree	Hyperlinks	•		
	Usage In	Activity	•	
		Interface Block	•	
		ItemFlow		
		State Machine	Rotary Pu	
				ransition ransition:Pump On[Off - Pumpi

Instance usage traceability

Trace all direct and indirect Instances of your Block.

Specification of Block Engine			×		
Block Instances The Instance node contains a list of all direct and indirect Instances of the Block. Click the Open Specification button to start modifying properties of selected item.					
	Instances				
Engine 	🚹 ĝį 🖙 🕄	∋ż abc			
Usage in Diagrams	Name	Classifier	Diagrams		
Usage In	🗆 Direct Insta	nces			
Ports/Interfaces	K90	Engine [Engine Tr			
Properties	Mark30	Engine [Engine Tr			
Operations Behaviors	350MagMPI	Engine [Engine Tr			
Relations	Indirect Instances				
Allocations	diesel	DieselEngine [Eng	📓 engine trade study 📰		
⊡… 🛅 Instances ⊕ 🚍 350MagMPI : Engine	hybrid	HybridEngine [Eng	🔚 engine trade study 📰		
⊕	gas	GasEngine [Engin	📓 engine trade study 📰		
🖻 ·· 📼 Mark30 : Engine	newDiesel	DieselEngine [Eng			
i∰… ा diesel : DieselEngine					
🖽 🖃 hybrid : HybridEngine					
			Close Help		



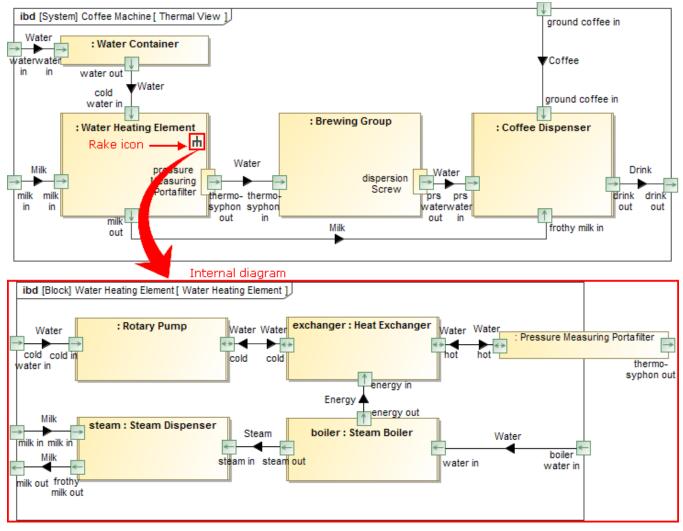
Informative Rake icon

Follow and double-click on the symbol with rake icon for faster navigation in the model! According to the SysML specification, the decomposition of model elements can be represented by the rake icon. It means a reference to a more elaborated diagram.

The rake icon automatically appears on:

- Call Behavior Actions that can refer to another Activity diagram.
- Parts that can refer to another Internal Block diagram.
- Packages that can refer to another Package diagram.
- Constraint Properties that can refer to another Parametric diagram.
- Requirements that can refer to another Requirement diagram.
- Interaction Uses that can refer to another Sequence diagram.
- States that can refer to another State Machine diagram.
- Use cases can that can be realized by other behavior diagrams such as Activity, State Machine or Sequence.

The rake icon is shown by default on the symbol in the new SysML projects. To hide the rake icon set the Show Rake Icon value to false in the Symb ol Properties dialog.



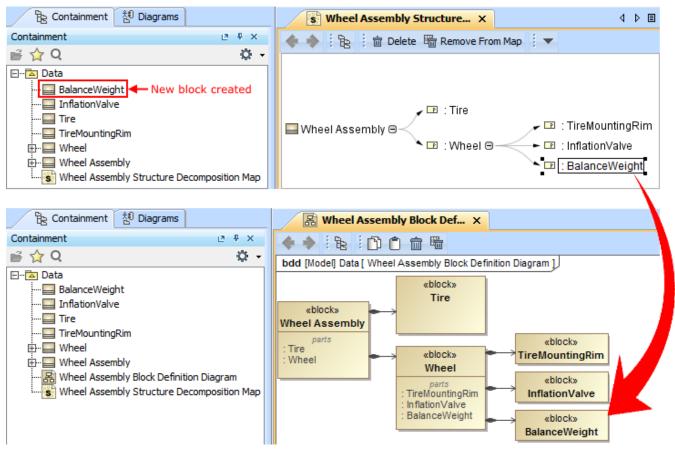
Simplified Entering of Information

Editable Relation Maps

Now the Relation Map works like a Mind Map! You can use it not only to browse through a model, but also to create new elements with single click. The Structure Decomposition and Activity Decomposition maps became the fastest model creation tools.

TIP! To create elements faster and more easily:

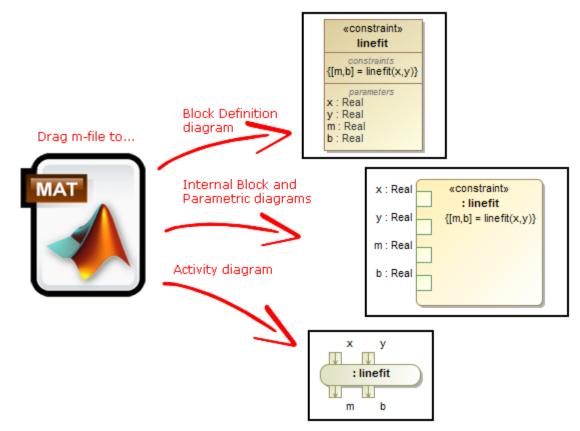
- Select the element and press Insert (Cmd+I for Mac users) to decompose it.
- Type Part or Action name and press Ctrl+Enter to create the element of the same type.



Wrapping MATLAB functions

Integrating custom MATLAB functions into the model is much easier now! Just drag the m-file to the diagram to create a Constraint Block, Constraint Property, or Call Behavior Action with corresponding function name and parameters. In earlier versions you could only drag the m-file on existing Constraint Block in Block Definition diagram.

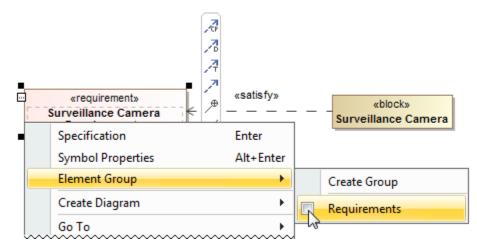
NOTE. M-file must contain function declaration.



Other News

Easy management of element groups

You can create element groups and add or remove group elements quickly and easily.



Verdict for Test Cases

When the Test Case is created, its return parameter for the verdict is created automatically.

🛒 Specification of Test Case Max Acceleration Test					
Specification of Parameters The Parameters contains a list of specific Parameters properties.					
Max Acceleration Test() : VerdictKind	Paramet		abc		
Documentation/Hyperlinks	to ĉi				
Parameters	Name	Туре	Default Value	Direction	
Template Parameters Relations	🗉 Genera	al			
	verdict1	💷 VerdictKind		return 🗉	
Constraints					
Usage in Diagrams					
Attributes	Lunn				

New property groups in Block Specification window

In addition to newly created Usage In, Instances, and Ports/Interfaces property groups, the UML-oriented Attributes property group in the Block Specification window is now replaced with the SysML-oriented Properties property group. All owned and inherited Block properties are grouped as in the Block compartments.

Specification of Block FuelTankAssembly					×	
Block properties The Properties node contains a list of Block properties. Create or delete properties. Use the properties specification button to edit properties of a specific attribute.						
🔳 🗽 🖸 💋	🔳 🗽 🖆 🥩 Properties					
FuelTankAssembly	∰ ≜į Bį ⊡‡ Bž abc					
🗈 Usage in Diagrams	Name	Туре	Default Value	Owner		
⊡⊡ Usage In ⊡ Constraints	Flow Properties					
	in fuelSupply	Fuel [HS		🔲 FuelTankAssembly [
Properties	out fuelReturn	Fuel [HS		🔲 FuelTankAssembly [
⊕ ⊡ : Fuel ⊕ ⊡ in fuelSupply : Fuel	Part Properties					
. In idesupply : I del 	fp	FuelPump		🔲 FuelTankAssembly [
🖽 📼 fp : FuelPump	Reference Properties					
🕀 🗔 fuelFlowRate : Real		Fuel [HS		FuelTankAssembly		
⊕ fuelWeight : Real ∰ Operations	Value Properties					
Behaviors	fuelFlowRate	💷 Real [Sys		FuelTankAssembly	200	
Relations Allocations Instances	fuelWeight	💷 Real [Sys		FuelTankAssembly		
	Up	Down	Create	Redefine Delete		
				Close Help		

More Classes in Open API

The com.nomagic.magicdraw.sysml.util.SysMLUtilities class was added to the open APIs. It provides utility methods for easier work with SysML projects. For more information, see Magicbraw installation directory openapi/docs.

SysML 1.4 Support

Now adjunct properties can represent State entry/do/exit behaviors and Transition effect behavior. From the State or Transition shortcut menu, select T ools > Adjunct Property > Create Adjunct Property for <...> to create adjunctProperty representing entry/do/exit behaviors or Transition effect behavior.

Fixed Issues

You can check the list of publicly available or your own reported issues fixed in version 18.1 FR.