

# What's new in Cameo Systems Modeler 18.0 LTR

Release date: 2nd June 2014

## Cameo Systems Modeler Enterprise Edition

The new Cameo Systems Modeler Enterprise edition, with Cameo Simulation Toolkit bundled in for engineering analysis automation. For more generic Cameo Simulation Toolkit new features and enhancements, visit [What's new in Cameo Simulation Toolkit 18.0 LTR?](#)

**NOTE.** The Cameo Simulation Toolkit evaluation version is also included in the Cameo Systems Modeler Architect edition.

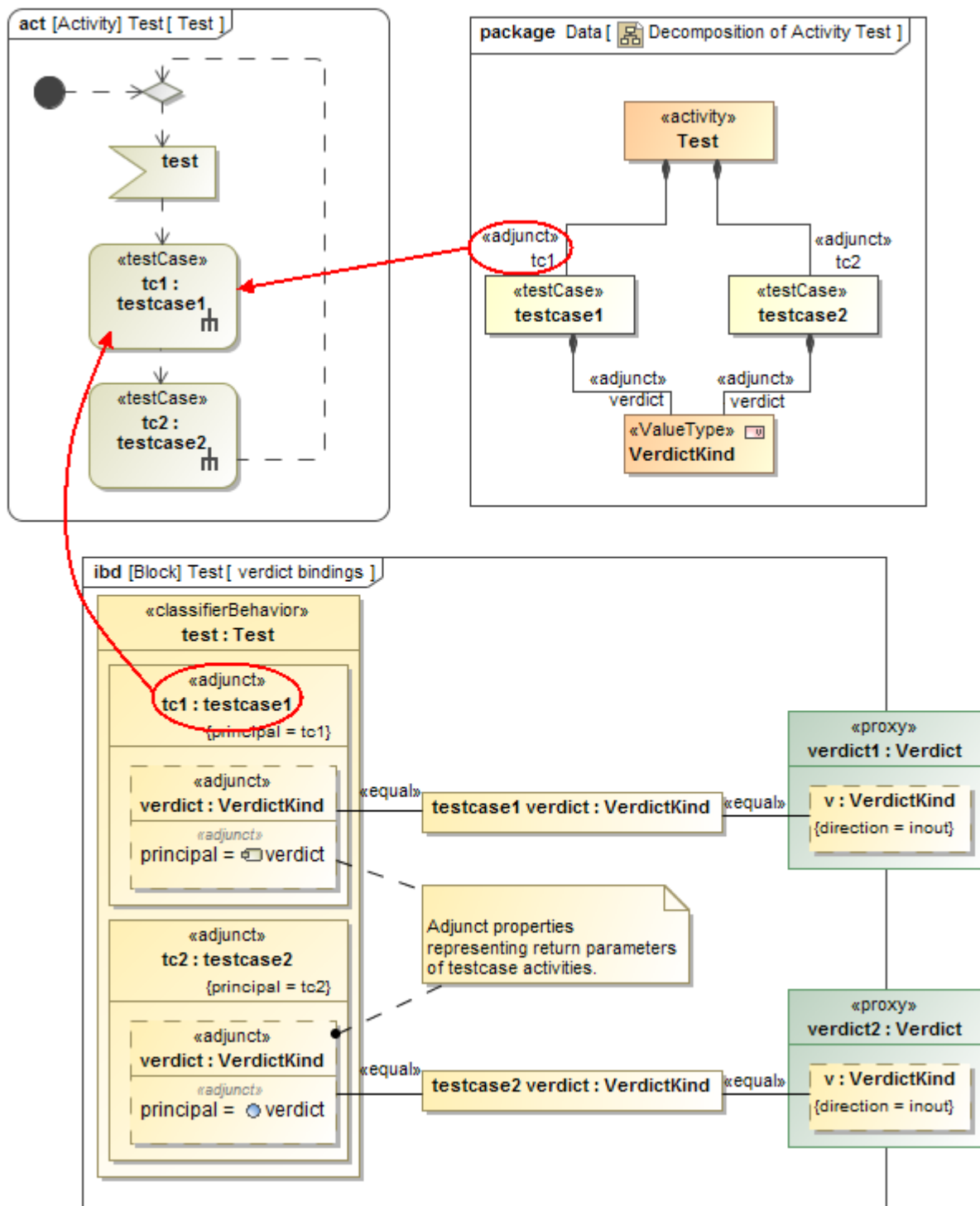
## SysML 1.4 Support

We are the first to introduce the new [SysML 1.4 specification](#) support, including new concepts such as AdjunctProperty, BoundReference, and ElementGroup, and the new QUDV and ISO 80000 libraries.



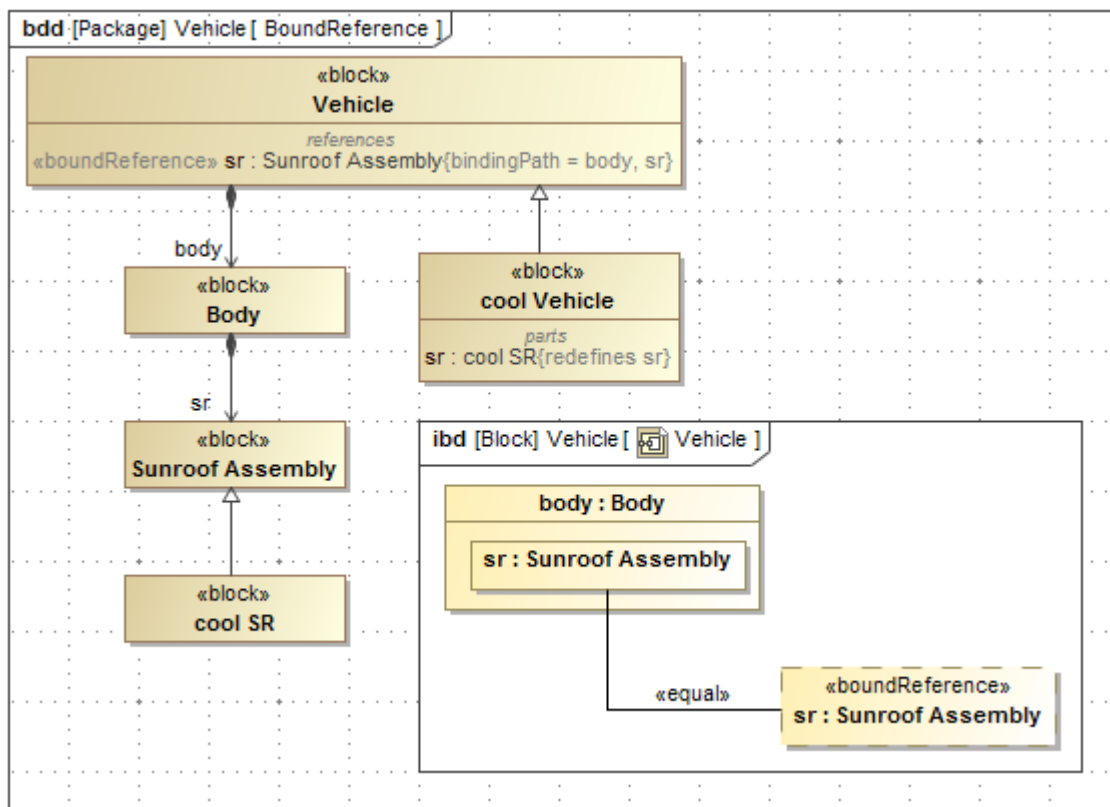
## Navigation Through Non-properties

SysML 1.4 adds an extension of properties, so they are now able to represent behavior diagram elements, semantically equivalent to properties, such as behavior parameters, call actions, or object nodes. This enables navigation through non-properties using deep nested connectors, and structural and behavioral elements linking, such as ports association with behavior parameters.



## Bound Reference

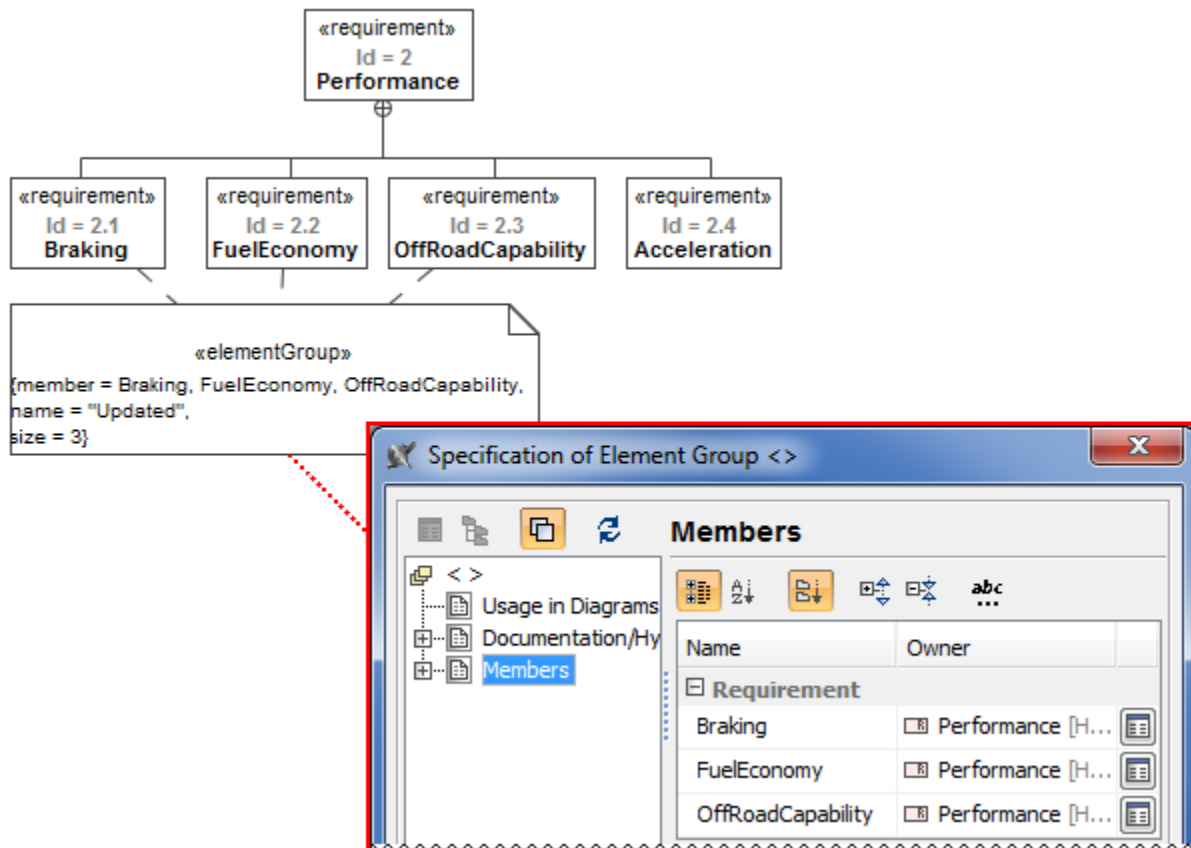
Bound Reference provides the modeler with a new way to quickly redefine and constrain deep nested system parts, without the considerable modeling efforts required previously. Bound Reference creates a virtual "shortcut" to a deep nested system part where multiplicity and type can be restricted by direct redefinition in subtypes.



## Element Group

The SysML 1.4 ElementGroup concept provides a convenient and lightweight mechanism to group model elements based on an arbitrary defined criterion without imposing ownership or containment constraints.

An element group is persistent, but does not own its elements and thus an element can participate in an unlimited number of named groups. For example, it can group elements associated with a particular baseline, have a certain risk level, responsible person, review status or any other custom criteria. Each Element Group is based on a UML Comment and extends the capability of comments to refer to multiple annotated elements, as shown below:



## New QUDV and ISO 80 000


The SysML 1.4 QUDV library was improved to:


- Comply with International vocabulary of metrology (VIM 3rd edition)
- Encode ISO/IEC 80000 definitions of base quantities and units to provide semantics for computer-based dimensional analysis











The ISO/IEC 80000 library, which is a collection of 14 standards, is available to use in new projects on demand (from the shortcut menu, select **Modules > Load Module**).

## OMG SysML 1.4 ISO 80000 : ISO/IEC 80000 Collection

### Coverage of Normative Definitions

 Complete Coverage

 Partial Coverage

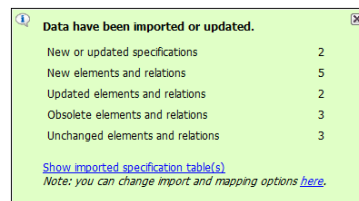
part 1	<b>ISO 80000-1:2009 : General</b>	SysML's QUDV 
part 2	<b>ISO 80000-2:2009 : Mathematical signs &amp; symbols</b>	
part 3	<b>ISO 80000-3:2006 : Space and time</b>	
part 4	<b>ISO 80000-4:2006 : Mechanics</b>	
part 5	<b>ISO 80000-5:2007 : Thermodynamics</b>	
part 6	<b>IEC 80000-6:2008 : Electromagnetism</b>	
part 7	<b>ISO 80000-7:2008 : Light</b>	
part 9	<b>ISO 80000-9:2009 : Physical Chemistry and Molecular Physics</b>	
part 10	<b>ISO 80000-10:2009 : Atomic and nuclear physics</b>	
part 13	<b>IEC 80000-13:2008 : Information science and technology</b>	



### ReqIF Support

The **Requirements Interchange Format** (ReqIF) support enables requirements import from the requirements management tools, such as IBM DOORS 9.4 and 9.5, IBM DOORS Next Generation, PTC Integrity, Polarion, and Siemens Teamcenter.

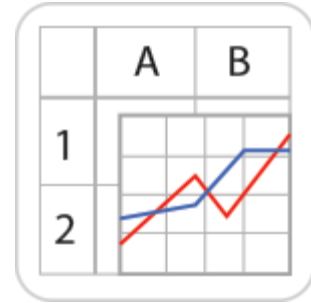
Requirements can also be updated by importing a new version of the file again. The user is notified about changed or removed requirements and can manage the changes.



← Notification window

#	Hierarchy Id	Name	Status
1	1	<input type="checkbox"/> Introduction	Updated
2	1.1	<input type="checkbox"/> Scope	Unchanged
3	1.2	<input type="checkbox"/> Assumptions(This Section Was Updated)	Obsolete
4	1.3	<input type="checkbox"/> Exclusions	Obsolete
5	1.3.1	<input type="checkbox"/> System Instances	Obsolete
6	2	<input type="checkbox"/> Changes Made	Unchanged
7	2.1	<input type="checkbox"/> Changes	Updated
8	3	<input type="checkbox"/> Contents	Unchanged

The new **Instance Table** allows you to manage block instances, perform engineering analysis, as well as present imported spreadsheet data or trade study results.



The instance table is a spreadsheet-like data entry and review interface. Actually, the feature is based on a generic table.

With the help of an instance table you can easily:

- Review instances of one or more classifiers, in the single place.
- Create instances for one or more classifiers.
- Edit slot values of the instances displayed in the table.
- Customize the representation of the table.
- Export the data into an HTML, XLSX, or CSV file.

Cases for the practical application:

- Instance specifications management
- Testing data and results representation
- Parametric analysis results
- Excel spreadsheet import and representation
- Bill of Materials (BOM)
- Trade studies

All that needs to be done is create an instance table, add the instances, and then choose the slots to review and edit. This form of reviewing and editing data will definitely save a lot of time, as opening the Specification window of each instance is no longer required.

Instances of different classifiers

Non-editable cell represents intersection between instance and inappropriate column

Columns determined by different classifiers, for editing slot values

#	Name	Classifier	Boat.Year : Integer	Boat.Make : String	Boat.Beam : String	Engine.Serial Num : String	Engine.Year : Integer
1	WN234CD	Boat	1999	Calbaria	8.7"		
2	WN123AB	Boat	1977	Hanter	8"		
3	Mark30	Engine				M3060	1962
4	K90	Engine				C1075	1975
5	350MagMPI	Engine				M30099	1999

## New Approach for Calculating Metrics

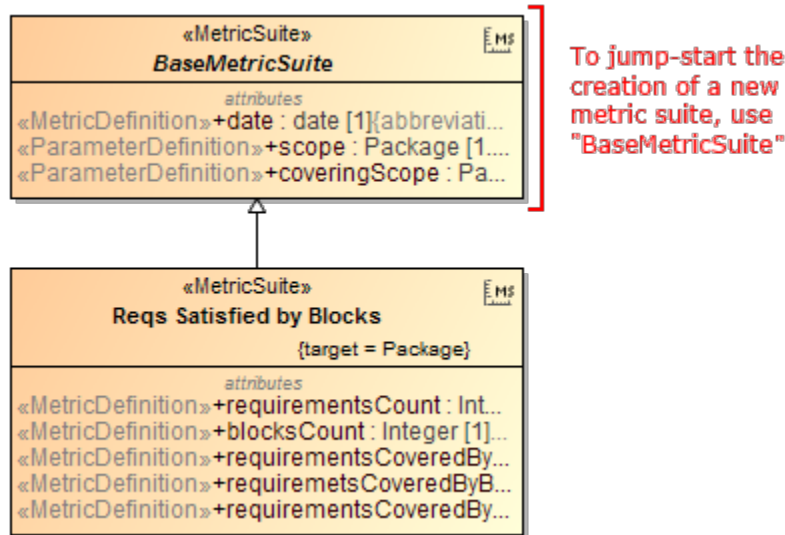
Need to perform a comprehensive analysis of current state of your project or get only the numerical evaluation of some particular aspect of it? Must track the evolution of your model over the course of time? Expect to have all the information represented in a user-friendly environment? **Try the new approach for calculating metrics!**

With this new approach you can

- Create metric suites for calculating metrics.
- Calculate metrics according to given parameters.
- Easily customize the representation of the metrics in the new type table – the metric table.

Creating a new elementary metric suite is as simple as creating a new class with a couple of stereotyped attributes: one for parameter definition and one for metric definition.

The parameter definition can be any model element or a primitive type, such as real or integer. The metric definition, that is, a formula for calculating metrics, can be easily specified either as a **structured expression** (by using the operations of the expression evaluation engine) or as a **script operation** written in any script language supported by MagicDraw (for example, JavaScript, Jython, Groovy). Moreover, one metric definition can use the results of other metric definitions.



Once the metric suite is created, you can calculate the metrics for a selected element. Analyze the results of the calculations, that is, metrics, in a metric table, which is automatically created for these metrics representation. You can easily reorganize the columns of the table, calculate new metrics, and recalculate the existing metrics with other parameters.

The screenshot shows the MagicDraw interface with the 'Metrics' menu open. The 'New Metric Table' dialog is displayed, showing the 'Criteria' section with the following data:

#	Date	Scope	Requirements Count	Requirements Covered By Block Count	Requirements Covered By Block Percentage
1	2014.02.05 17.11	HSUVModel	18	1	5.56
2	2014.02.06 14.50	HSUVModel	18	1	5.56
3	2014.02.06 14.52	HSUVModel	18	2	11.11

New Reports



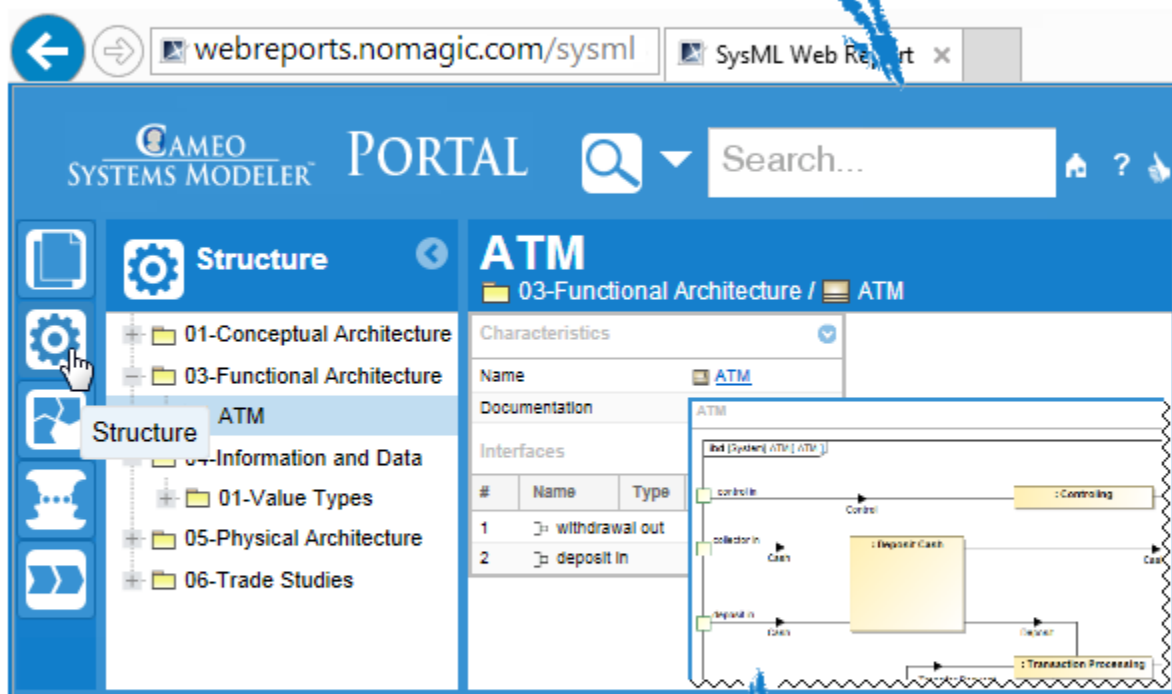
We present two new technologies for better model publishing:

- Web Portal report template
- Model based document generator

## Web Portal report template

Web Portal report provides a capability of sharing model information among all stakeholders in a browsable, clean and less-technical way. As opposed to the existing Web Publisher report, it does not clone original SysML model view, but re-organizes the most essential and relevant information by the Four SysML Pillars - Requirements, Structure, Behaviors and Parametric Constraints. Reviewers can provide [feedback](#) on any element or diagram.

Easily access information  
by Quick Search!



Browse in clearly grouped content!

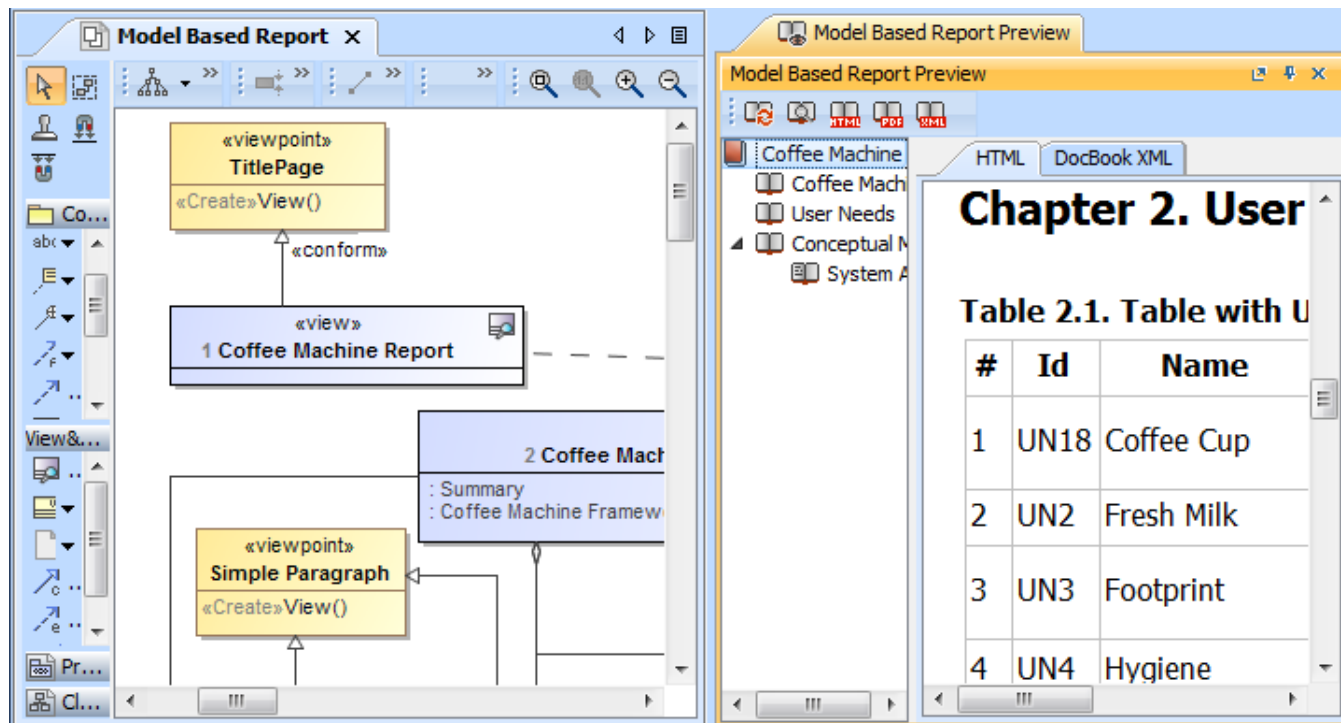
Discover the content which you  
or your colleagues have prepared!

**NOTE.** This is a technology preview of the systems engineering functionality. Please explore this new technology, experiment with it, and get back to us at [support@nomagic.com](mailto:support@nomagic.com) with your feedback and suggestions about possible improvements or features you are missing. We seek to create as serviceable tool as it could be, so we are thankful and very appreciative of your contributions!

## Model based document generator



SysML 1.4 improves the concept of View and Viewpoint to reflect perspectives of different stakeholders. The views are constructed from a subset of the model that addresses their concerns. Cameo Systems Modeler introduces a new technology which interprets the SysML 1.4 Views and Viewpoints models to construct XML document conforming with DocBook standard. A combination of diagrams, tables, model queries and simple text fragments can be presented in a built-in preview window or exported to PDF or HTML documents.



**NOTE.** This is a technology preview of the document modeling functionality. Please explore this new technology, experiment with it, and get back to us at [support@nomagic.com](mailto:support@nomagic.com) with your feedback and suggestions about possible improvements or features you are missing. We seek to create as serviceable tool as it could be, so we are thankful and very appreciative of your contributions!

## Extended Script Languages Support

Custom validation rules, derived properties, queries for smart packages, metric definitions, criteria for dependency matrices, and so forth can now be defined (as executable operations) in any of the following script languages:

- JavaScript
- Jython
- JRuby
- Groovy
- BeanShell

This speeds writing of these operations, since an already known language can be used rather than learning additional languages.

## Glossary

Avoid confusion – **use Glossary!** Define your terms in one place and later use them everywhere in your project. Reuse glossaries in other projects.

Glossary saves time by ensuring consistent usage of terminology in the organization. It also improves the communication between team members since terms are understood in the same way and definitions become visible everywhere the terms are used.

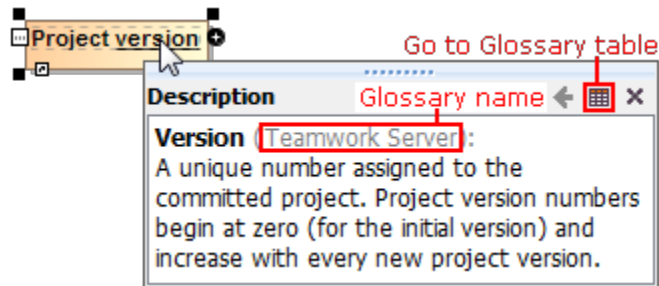
#	Term	Description
1	Author	A user who has committed a new <u>project version</u> .
2	Comment	Optional description of changes in the committed <u>version</u> .
3	Project	A physical working unit that consists of model, model visualizations (diagrams, tables, matrices, etc.), configuration data, and references to other elements residing in modules.
4	Project category	<u>Project category</u> concept which enables visual grouping of projects in Teamwork Server <u>repository</u> .
5	Repository	A storage place for projects and their versions managed by the Teamwork Server.
6	Version	A unique number assigned to the committed <u>project</u> . <u>Project version</u> numbers begin at zero (for the initial <u>version</u> ) and increase with every new <u>project version</u> .

Creating a term is quick and easy. Just do one of the following:

- Add any word or phrase to the glossary while typing or editing.
- Drag any element you wish to become a term to the glossary table.
- Click the **Add New** or **Add Existing** button on the glossary table toolbar.

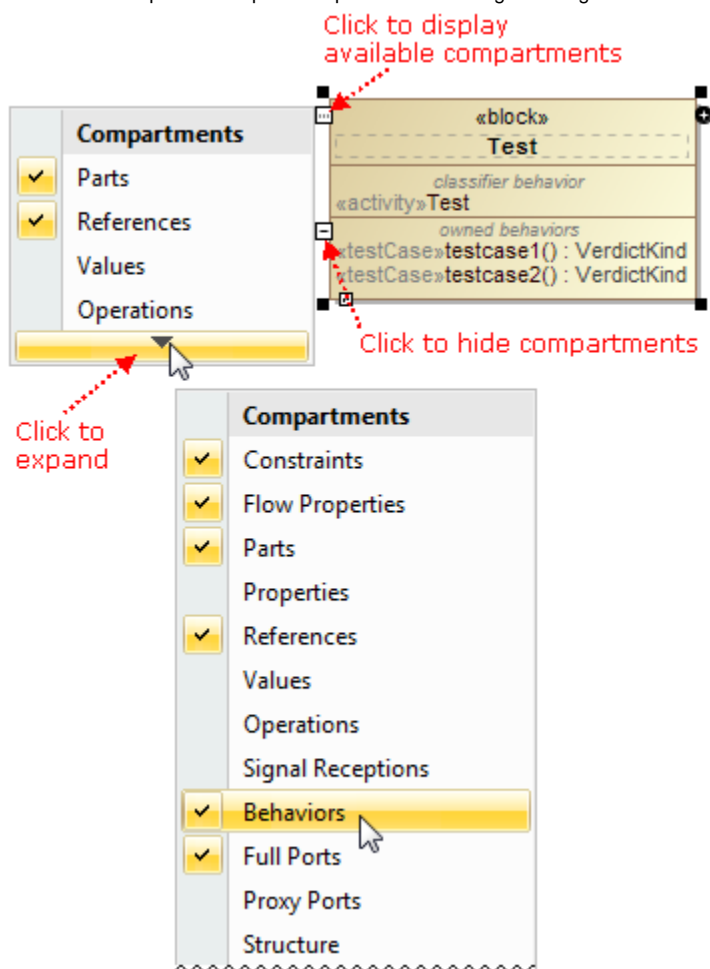
After terms are defined in the glossary, it is easy to use them in your project. Just press **Ctrl+spacebar** while typing to get a list of available terms.

For your convenience, a term description becomes visible when you move the pointer over this term.



### Enhanced Compartments and Their Management

- New smart manipulator for quick compartments enabling/disabling on the left shape border.



- Empty compartment lines are not shown anymore, making shapes cleaner.
- Labels added on all compartments, including operations, signal receptions and others.
- New Behaviors compartment on Blocks and parts listing their behaviors.
- Port, as a special kind of the Part, is capable showing all Part compartments now, including default values, structure, operations and even behaviors.

### SysML Profile API Changes

SysML Profile API changes were made in relation to the SysML 1.4 support.

The following constants were moved from the *com.nomagic.magicdraw.sysml.util.SysMLProfile* to *com.nomagic.magicdraw.sysml.util.MDCustomizationForSysMLProfile*:

```
public static final String CONSTRAINTPROPERTY_STEREOTYPE = "ConstraintProperty";
public static final String QUANTITYKIND_STEREOTYPE = "QuantityKind";
public static final String QUANTITYKIND_DEFINITIONURI_PROPERTY = "definitionURI";
public static final String QUANTITYKIND_DESCRIPTION_PROPERTY = "description";
public static final String QUANTITYKIND_SYMBOL_PROPERTY = "symbol";
public static final String UNIT_STEREOTYPE = "Unit";
```

The following methods were moved from the *com.nomagic.magicdraw.sysml.util.SysMLProfile* to the *com.nomagic.magicdraw.sysml.util.MDCustomizationForSysMLProfile*:

```
getConstraintProperty()
getQuantityKind()
getUnit()
isQuantityKind()
isUnit()
isConstraintProperty()
```

The constant *NESTEDCONNECTOREND\_PROPERTYPATH\_PROPERTY* changed to *ELEMENTPROPERTYPATH\_PROPERTYPATH\_PROPERTY*.

## Fixed Issues

- To open the list of publicly available or your own **system modeling** specific issues those have been included into version 18.0 LTR, click [here](#).
- To open the list of publicly available or your own **simulation** specific issues those have been included into version 18.0 LTR, click [here](#).
- To open the list of publicly available or your own **other** issues those have been included into version 18.0 LTR, click [here](#).