

# Defining system model using variation points



This step does not require having Configurations [defined](#) in the model. It can be done before/in parallel with defining Configurations.

## What is a System Model with Variation Points?

A System Model, also known as a 150% model, is a model representing a system that has variable parts. These are marked using Variation Points.

## Guidelines for defining System Model with Variation Points

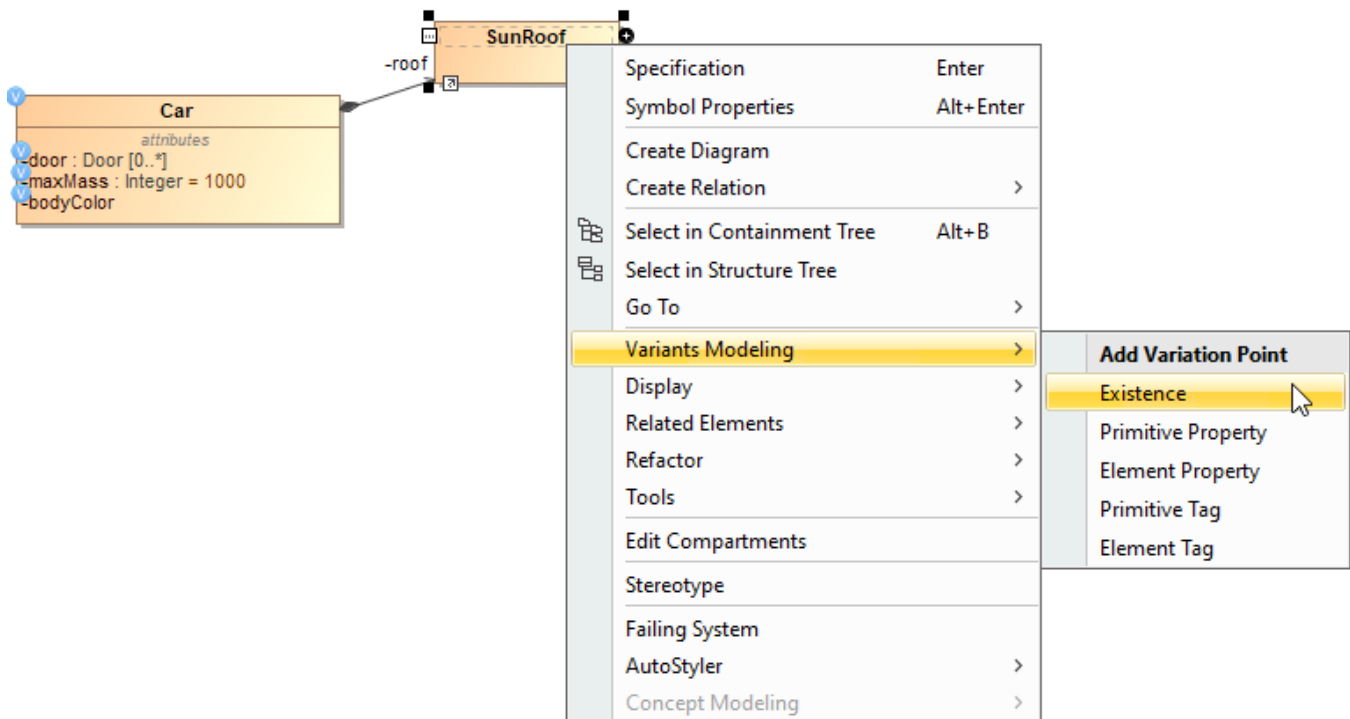
- Model your system using common SysML practices - Requirements, BDD, IBD, Activity diagrams, etc.
- Whenever some part of your System Model is variable, you need to mark that specific part using **Variation Points**.


## Kinds of Variation Points

If the model element (Requirement, Block, Part, Action, etc.) does not exist in some variants of your system, then mark it with the **Existence** Variation Point. It is the most commonly used Variation Point.

To mark an element with the Existence variation point

1. Right-click the element and then from the shortcut menu, select **Variants Modeling > Add Variation Point > Existence**.



This action marks the system model element with a Variation Point. A Variation Point icon  is displayed on it.



### Additional information

In the model, the Variation Point is stored as a special constraint with the «ExistenceVariationPoint» stereotype applied to it. Constraint's *constrainedElement* is your system model element.

If the model element (Requirement, Block, Part, Action, etc.) has any kind of field value that can be modified, then mark it with the **Modification/Value** Variation Point. It has the following flavors:

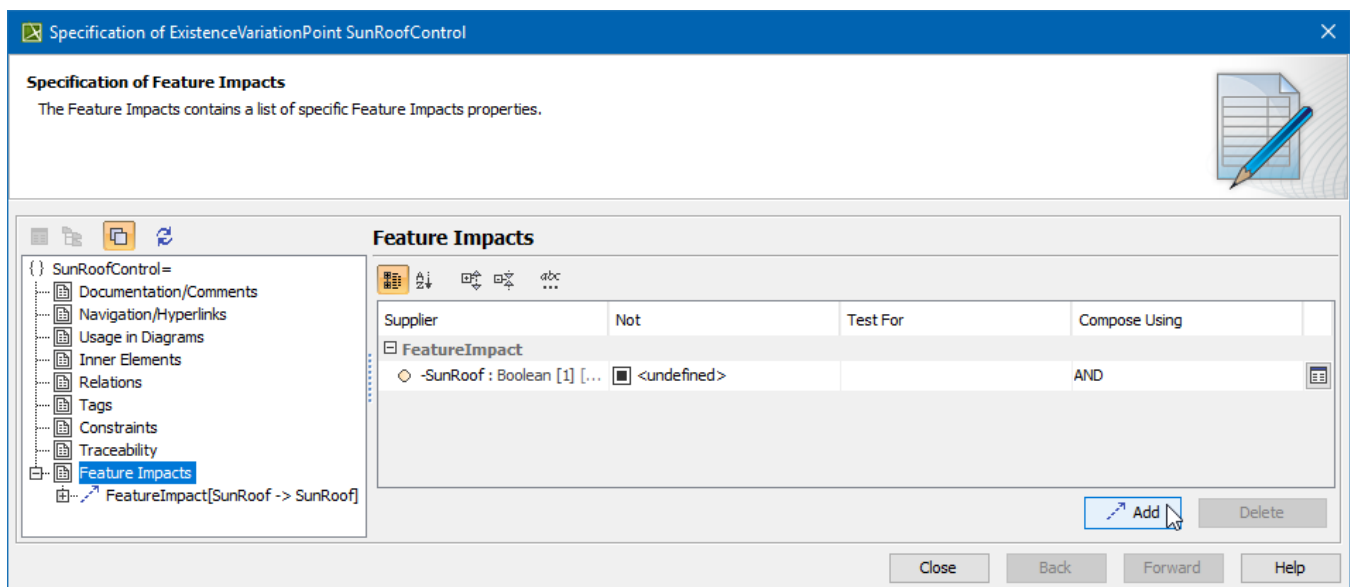
Variation point kind	Description	Values to specify
Primitive Property	Select it to set the values of UML model element properties that accept <i>strings</i> , <i>numbers</i> , <i>booleans</i> as values. Typical examples are default value ( <i>numeric</i> ), multiplicity ( <i>numeric</i> ), documentation ( <i>string</i> ), name ( <i>string</i> ).	Name of the property that is controlled by this variation point.  Expression returning the value for that property.
Primitive Tag	Select it to set the values of model element properties that accept <i>strings</i> , <i>numbers</i> , <i>booleans</i> as values, in cases when a property is not a standard UML property but defined as a tag in an extension profile. A typical example is a requirement text.	Name of the tag that is controlled by this variation point.  Expression returning the value for that tag.
Element Property	Select it to control the values of those UML model element properties that accept other UML elements as values. A typical example would be a property type, property default value for enumeration-type properties.	Name of the property that is controlled by this variation point.  Element Value Specification with a table of pairs: condition-value, where each condition is a <i>boolean</i> expression.
Element Tag	Select it to control the values of tags that accept other UML elements as values in cases when a property is not a standard UML property but defined as a tag in an extension profile.	Name of the tag that is controlled by this variation point.  Element Value Specification with a table of pairs: condition-value, where each condition is a <i>boolean</i> expression.

## Specifying features from the Feature Model

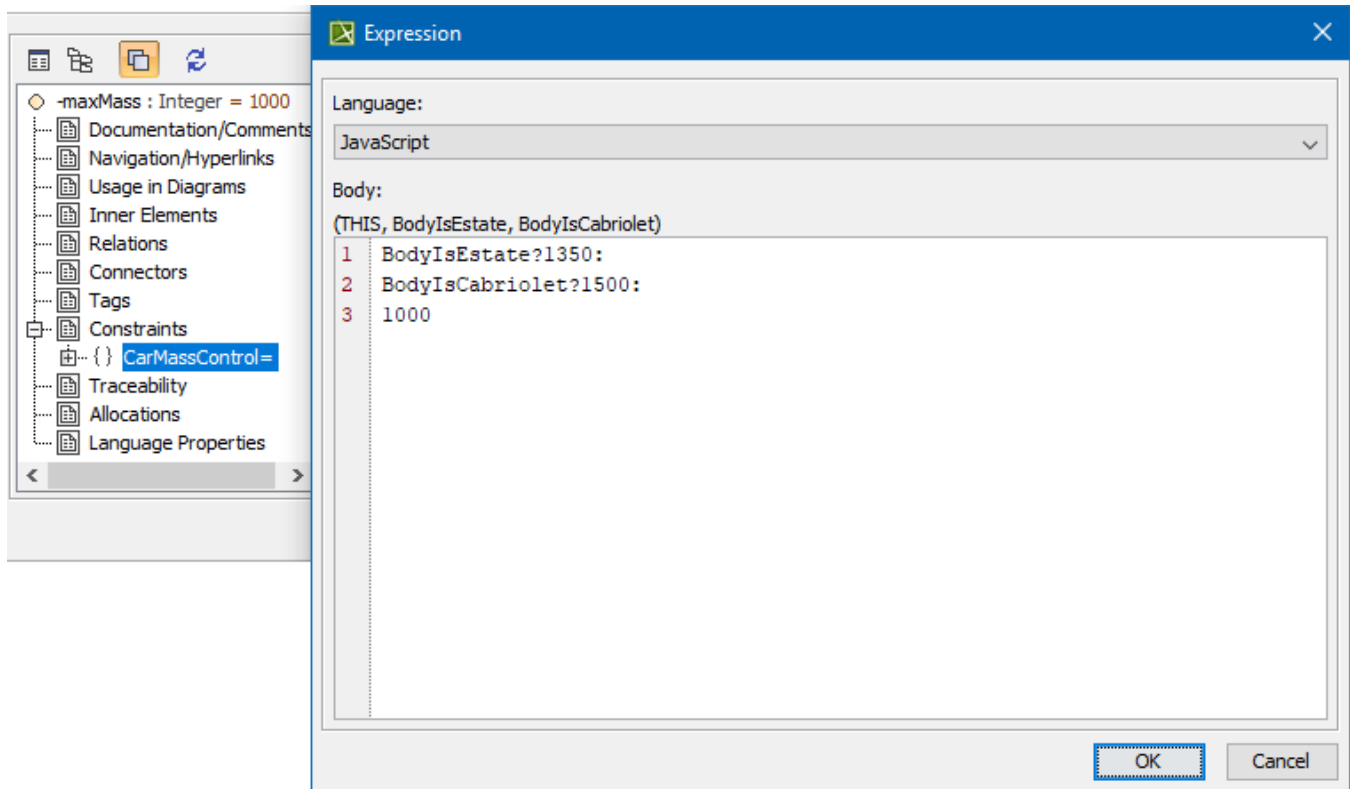
Once the Variation Point is specified, you need to specify feature impacts, i.e., specify the features from the Feature Model that influence this Variation Point.

To specify the features from the Feature Model

1. In the Variation Point Specification window, go to **Feature Impacts**.
2. Click the **Add** button to specify the influencing features.



Then, you need to construct a condition (*Boolean* expression) for the Variation Point. This is specified in the **Expression** field of the Variation Point Specification window. Use any of the scripting languages supported by the modeling tool (JavaScript, Python, Ruby, etc.). Oftentimes, if your expression is simple/trivial, you can skip this step.



If you keep the **Expression** field empty, the modeling tool **will infer** the expression for you. If your variation point depends on one or more *Boolean* features, for example, X, Y, and Z, and you keep the expression field empty, the modeling tool will infer the expression: "X and Y and Z" automatically.