pure::variants Integration for Capella Manual

pure-systems GmbH

Version 5.0.8.685 for pure::variants 5.0

2021

Table of Contents

1. Introduction

The pure::variants Integration for Capella provides a means to add variability information to Capella models. Based on this variability information pure::variants can then be used to flexibly create multiple differing variants from a single Capella master model. This provides a boost in development efficiency since instead of having to merge changes in slight variations of the base project, the change is applied once to the master project and then all relevant variants are automatically generated by pure::variants.

1.1. Software Requirements

The following software has to be present on the user's machine in order to support the pure::variants Integration for Capella:

Capella: Capella or Capella Studio of version 1.1.0 - 5.1.0 is required. Compatibility with other Capella releases is not guaranteed. The pure::variants Integration for Capella needs to be installed in Capella.

1.2. Installation

To install the pure::variants Integration for Capella, open Capella or Capella Studio and select **Help**->**Install New Software...**. Enter the address of your pure::variants update site. From the list of available features, select

"pure::variants - Integration for Capella" and make sure that **Contact all update sites during install to find required software** is checked. Then complete the wizard.

For installation in Capella, the standard Eclipse update site needs to be set up. Otherwise the installation will fail due to missing dependencies. In Capella 1.1.x the Eclipse update site is configured per default. In Capella 1.2.x, the Eclipse Neon update site still needs to be added. To do that, open **Window**->**Preferences**->**Install/Update**->**Available Software Sites** and add the update site `http://download.eclipse.org/releases/neon/` . For Capella 1.3.x, the update site is `https://download.eclipse.org/releases/oxygen/`. For Capella 1.4.x the update site is `https://download.eclipse.org/releases/2019-03/`. For Capella 5.x the update site is `https://download.eclipse.org/releases/2020-06/`

For more details on installation of pure::variants extensions, see section "Additional pure::variants Extensions" in the pure::variants User's Guide. You can find it in pure::variants at **Help**->**Help Contents**.

Please note that for transforming Capella master projects, pure::variants is used. In pure::variants the Connector for Capella Integration needs to be installed. See the pure::variants Connector for Capella Integration manual for details.

1.3. About this manual

The reader is expected to have basic knowledge about and experiences with pure::variants. The pure::variants manual is available in online help as well as in printable PDF format here.

2. Using pure::variants Integration for Capella

2.1. Supported Types of Variability Information

The main purpose of the pure::variants Integration for Capella is to support the user in adding and editing variability information in Capella master models. Based on this variability information, pure::variants can be used to derive multiple variants from the Capella master model. Each of the variants is derived from the master model by removing Capella elements not relevant in the respective variant. Removal of elements is controlled by pure::variants *conditions*. Conditions can be added to Capella elements by the master model author(s). The conditions are evaluated by pure::variants based on a variant model which captures the variant-specific feature selection. One element can be mapped to multiple conditions. If one of the element's conditions evaluates to 'false', the element is removed.

Apart from conditions, it is also possible to add a pure::variants *calculation* to an attribute of a Capella element. Like conditions, calculations are evaluated based on a variant model. However, instead of removing the attribute, its value is set to the evaluation result of the calculation.

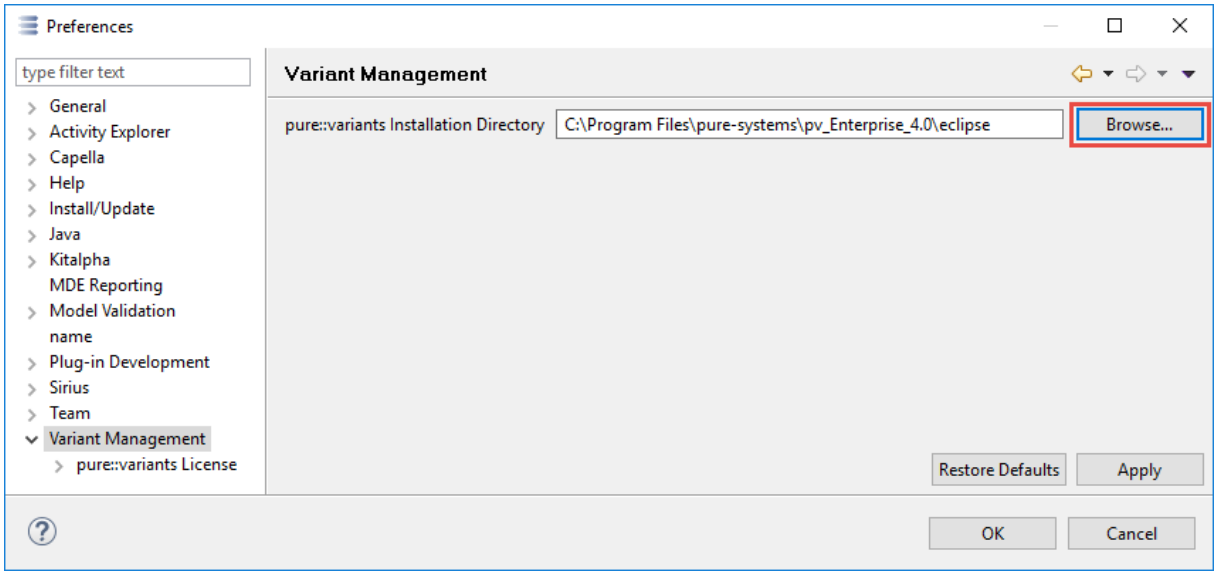
Each calculation also has a special condition assigned, which decides whether the calculation is evaluated when deriving a variant, or not. One attribute of an element can be mapped to multiple calculations. When mapping multiple calculations to the same element attribute, the calculations' condition should be used to make sure that only one calculation is evaluated during transformation of a variant. If multiple calculations apply to one element attribute during transformation, the value of the first calculation is used (based on the order of calculations in the Capella model below the Advanced Variability node).

2.2. Setup preferences for using Integration for Capella

Before starting to add variability information to Capella elements it is necessary to set the location of your pure::variants Installation and check whether the pure::variants License is set correctly. This is needed to be able to load pure::variants models.

To set the location, open the Capella preferences (**Window**->**Preferences**), and select **Variant Management**. Use the **Browse...** button to select the pure::variants install directory.

Figure 1. Setting the pure::variants Installation Directory



To set the pure::variants license, open preference page **Variant Management->pure::variants License** and, if a floating license server is used, also page **Variant Management->pure::variants License->License Server**.

3. Tasks

3.1. Preparing the Mappings View

For using the pure::variants Integration for Capella, the *Mappings* view is needed. To show it, select **Window->Show View->Other...** and choose the Mappings view from category **Variant Management**. This view lists all conditions and calculations of the current Capella model. To select a Capella model, just select any Capella element in the **Capella Project Explorer** or a Capella diagram.

For supporting auto-completion when writing pure::variants conditions or calculations, it is necessary to load a configuration space. To do this, use button **Select Configuration Space** () and select a configspace.xml file in the dialog that opens.

3.2. Creating Variability Information

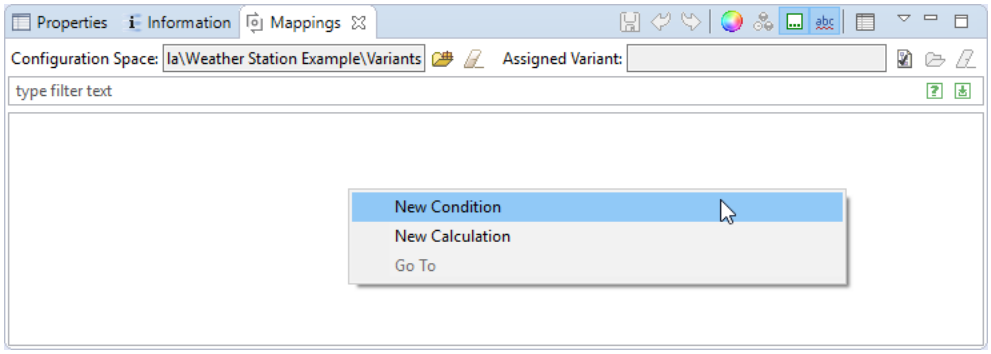
Creating a Condition on a Capella Element

There are multiple ways how to add an element to a condition: via the **Mappings** view context menu, via drag & drop onto a condition, or via the context-menu of a Capella element.

Using Context Menu in Mappings View

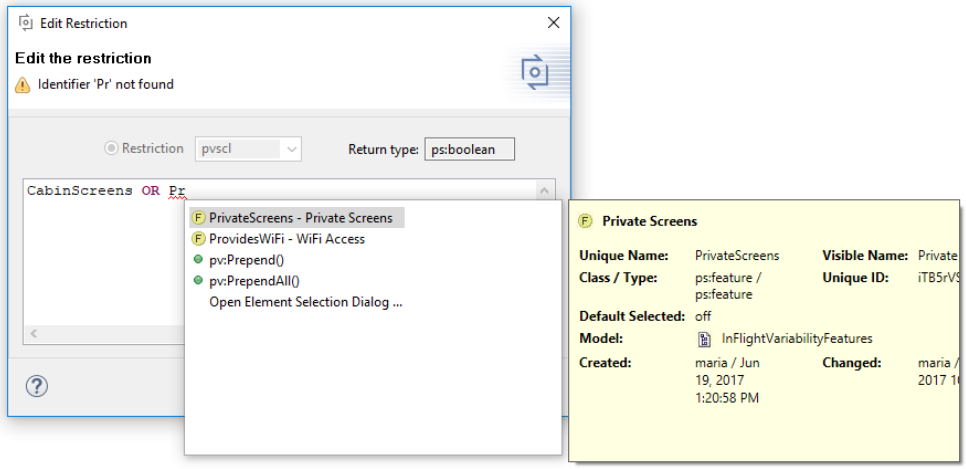
At first a condition must be added in **Mappings** view using the **New Condition** context menu.

Figure 2. Create Condition by Context Menu



The menu opens the condition editor. The editor supports writing conditions using auto-completion. To use the auto-completion, press `Ctrl+Space`.

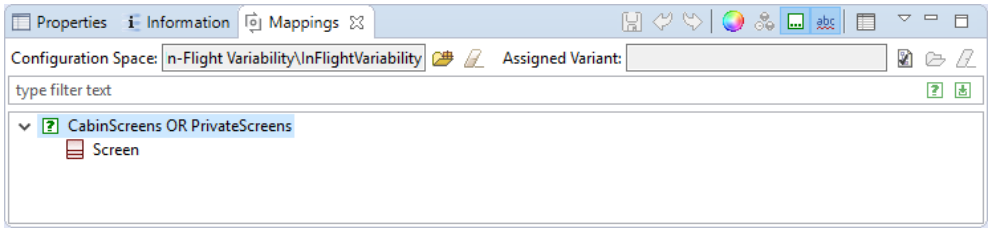
Figure 3. Condition Editor Showing the Auto Completion



Pressing **OK** will store the new or changed condition in the current Capella model. To add a Capella element to the newly created condition, select the element in the opened Capella model, right-click on the condition and select **Add ...**.

Alternatively, you can also drag & drop Capella elements onto the condition.

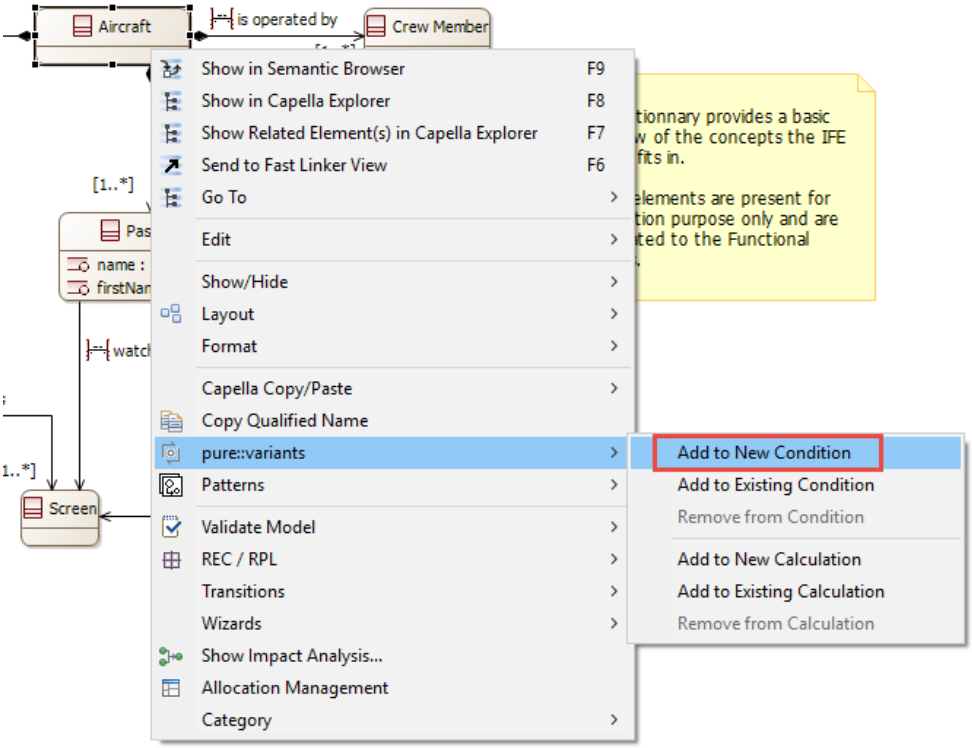
Figure 4. Restricted Capella elements in Mapping view



Using Context Menu in Capella model

Another way to add a condition to one or more Capella elements, is to right-click on the selected element(s) in the Capella model (e.g., in diagram or browser) and select **pure::variants->Add to New Condition**. This will open the condition editor to create a new condition and then add the selected element(s) to it.

Figure 5. Adding Capella Element to New Condition



Alternatively, you can select **pure::variants->Add to Existing Condition**. This will open a dialog to select an existing condition to which the selected element(s) will be added.

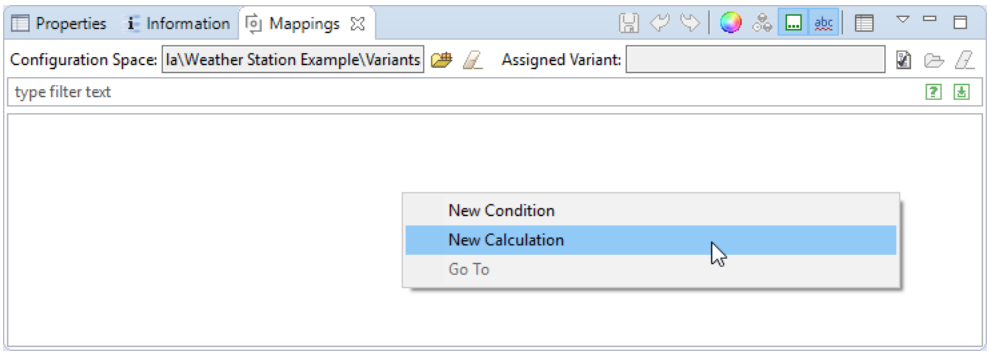
Creating a Calculation on an attribute of a Capella Element

Capella element attributes can be added to a calculation in similar ways as conditions: via the **Mappings** view context menu, via drag & drop onto a calculation, or via the context-menu of a Capella element.

Using Context Menu in Mappings View

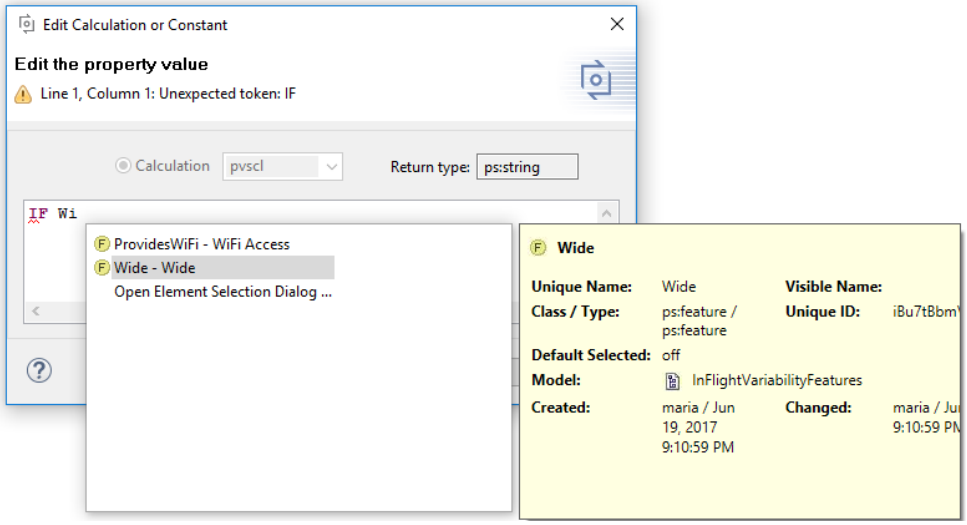
First a calculation needs to be created in the **Mappings** view using the **New Calculation** context menu.

Figure 6. Create Calculation by Context Menu in Mappings View



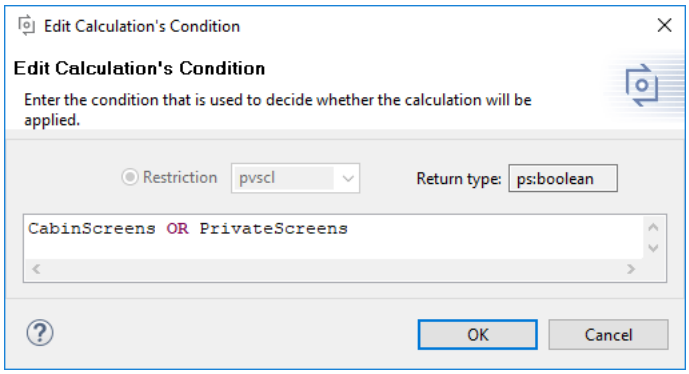
The menu opens the calculation editor, which works in the same way as the condition editor.

Figure 7. Calculation Editor Showing the Auto Completion



After pressing **OK** a second editor is shown, in which you can enter the calculation's condition. During transformation and preview, this rule is used to decide whether the calculation will be applied.

Figure 8. Calculation's Condition Editor

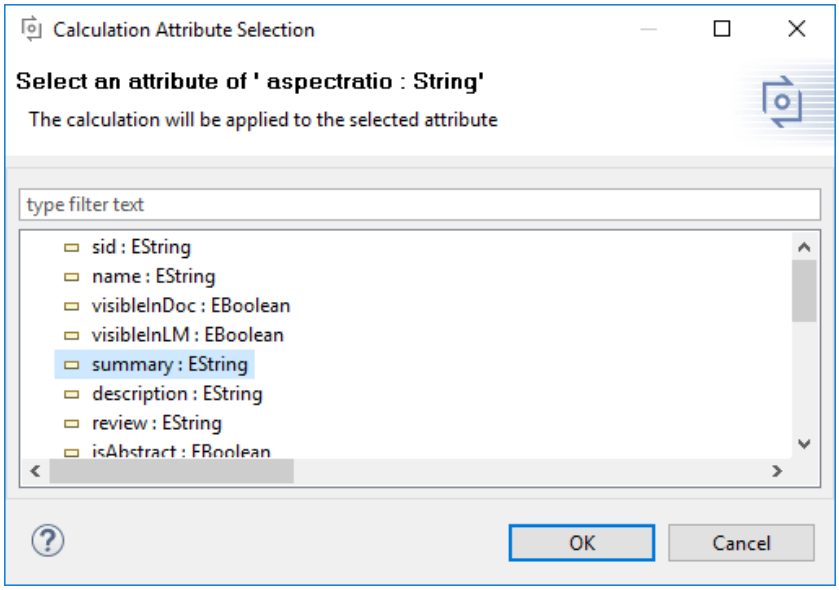


When both dialogs are closed, the new or changed calculation is stored in the current Capella model.

To add the attribute of a Capella element to the newly created calculation, select the element in the opened Capella model, right-click on the calculation and select **Add ...**.

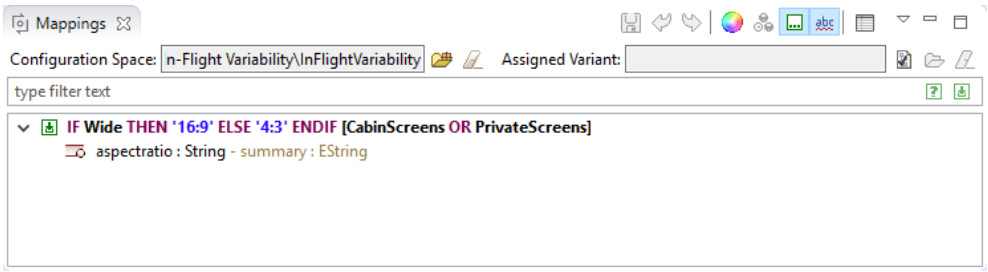
If this is the first element that is added to the calculation, a dialog opens that allows to select the attribute on which the calculation should be applied. All attributes of the current element are listed here. However, only attributes which are based on type *String*, *Boolean*, or a basic number type, such as *Integer*, *Float* or *Double* are currently selectable. If another element is already mapped to the calculation, the same attribute is used as for the other element(s).

Figure 9. Attribute Selection Dialog



Similar to conditions, you can also drag & drop a Capella element onto the calculation. This will also open the attribute selection dialog if necessary.

Figure 10. Restricted Capella elements in Mapping view



Using Context Menu in Capella model

Another way to add a calculation to an attribute of a Capella element is to right-click on the selected element in the Capella model (e.g., in diagram or browser) and select **pure::variants->Add to New Calculation**. This opens the attribute selection dialog and after pressing **OK**, opens the calculation editor and calculation condition editor for creating a new calculation. The selected attribute is added to the newly created calculation.

Alternatively, you can select **pure::variants->Add to Existing Calculation**. This also opens the attribute selection dialog and after pressing **OK** opens a dialog to select an existing calculation to which the selected attribute will be added.

3.3. Preview Variable Elements and Variant Results

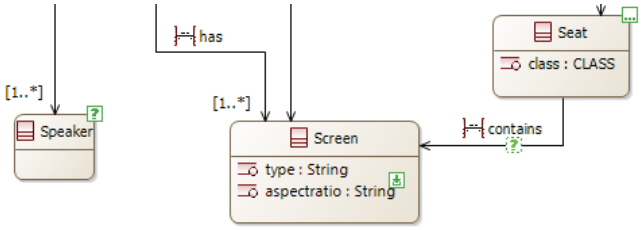
Identify Variable Elements in Capella Models

To find out whether an element is mapped to a pure::variants condition or calculation, activate the **Decoration** mode ( ) in the **Mappings** view. When decoration mode is enabled, Capella elements are marked as follows in Capella diagrams:

Decoration Icon	Shown when
	a Capella element is mapped to a condition

Decoration Icon	Shown when
	a Capella element is indirectly mapped to a condition, either because one of its parents is mapped, or due to propagation (see the section called “Enable Propagation Rules”)
	an attribute of this Capella element is mapped to a calculation
	mapped to both a calculation and a condition

Figure 11. Decoration Mode in Diagram



When hovering over the decoration icon, more information is shown. For conditions, the effective pvSCL condition is shown in brackets. This means that all conditions that apply to this element are concatenated with operator `AND`. For calculations, the pvSCL calculation that applies to this element is shown, and the attribute which is mapped to the calculation. After that, the calculation's condition is shown in brackets.

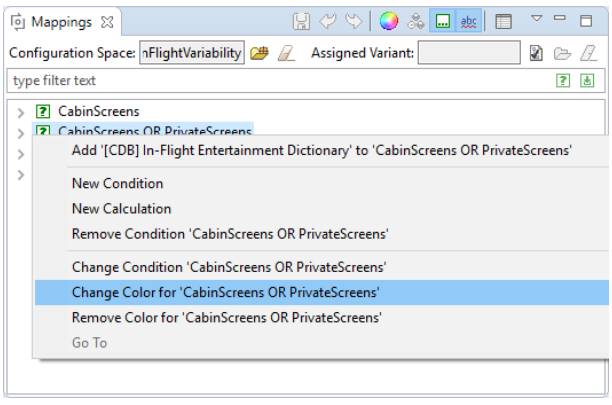
To add this text information to the diagram, select **Decoration Mode: Show Labels in Diagrams** from the drop-down menu of the **Mappings** view.

In the Capella project explorer, no icons are shown in decoration mode. However, the same text as in diagrams is added to elements.

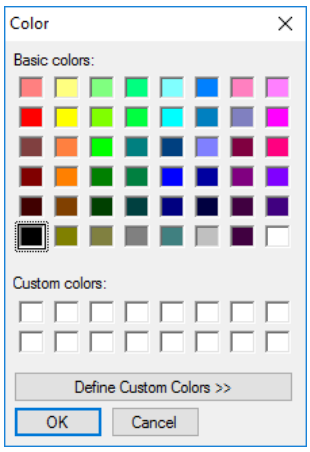
Color Elements

After activating the **Colorize** () mode in the **Mappings** view, all views and editors showing the Capella model are updated to colorize elements which have a conditon or calculation and this has set a color. To set a color, select a condition or calculation and press the **Change Color** context menu in **Mappings** view.

Figure 12. Set Color of Condition

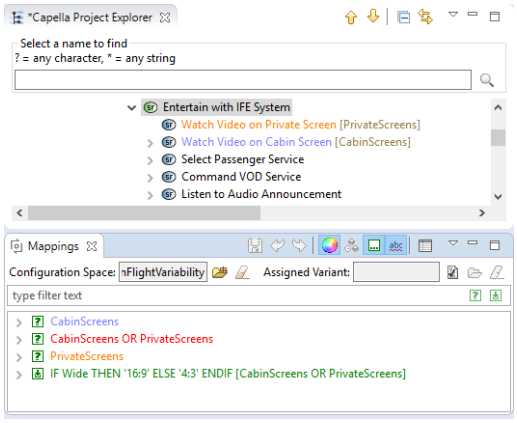


Use context menu to change color.

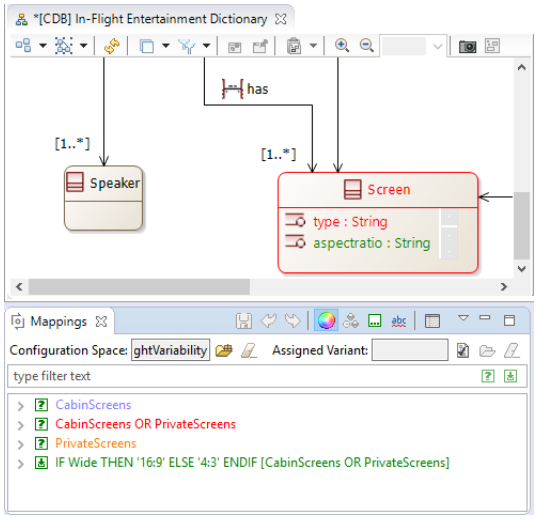


Choose the color.

Figure 13. Preview Variant: Colorize Elements



Visualization in Capella Project Explorer.



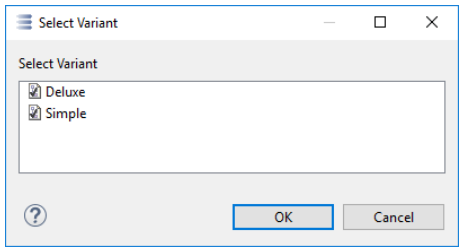
Visualization in Capella diagram.

Preview a Variant

To enable the preview of variants, a variant description model needs to be selected. To do this, open the **Mappings** view and press the **Assign Variant** ( ) button.

In the dialog that opens select the model and press **OK**.

Figure 14. Select Variant Description Model

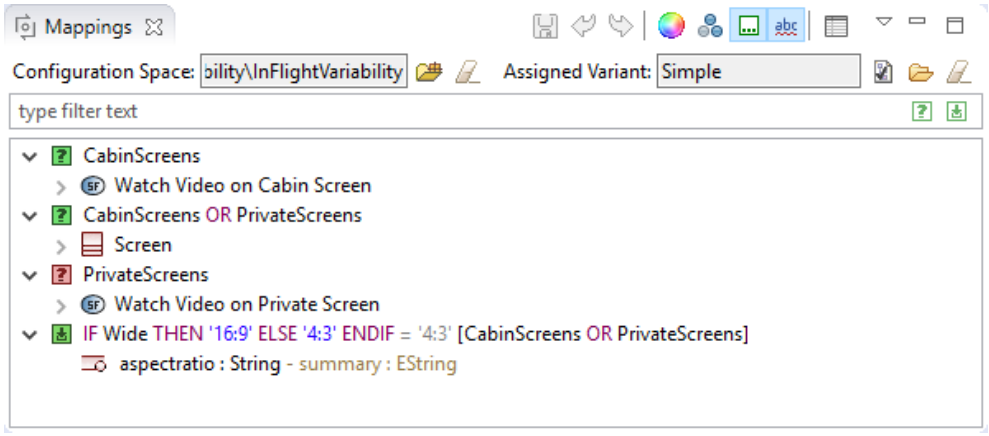


The variant description model is loaded and evaluated. The **Mappings** view changes the images in front of conditions and calculations. After loading a variant the images are:

Result Icon	Description
	green if the condition evaluates to `true`
	red if the condition evaluates to `false`
	green if the calculation's condition evaluates to `true`
	red if the calculation's condition evaluates to `false`

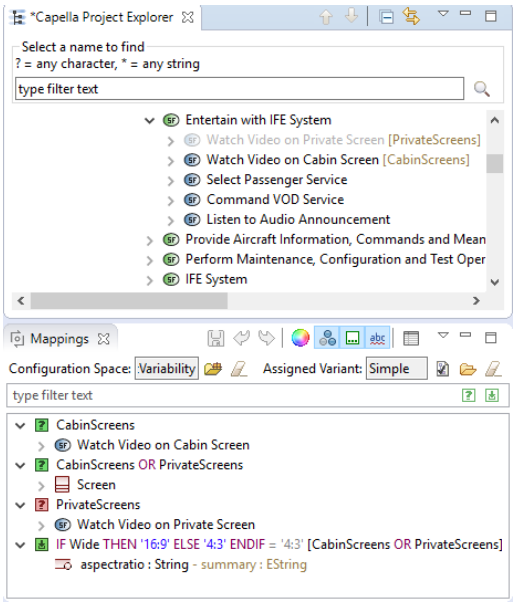
Furthermore, the evaluated value of the current variant is shown behind each calculation.

Figure 15. Mappings View with Loaded Variant

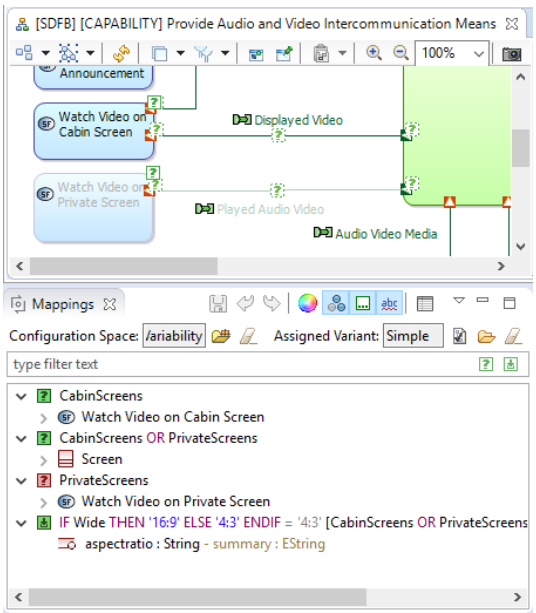


The preview mode is also enabled. To start the visualization in the Capella model, activate the **Variant** ( ) mode. All elements which are not in the currently selected variant are grayed out. Alternatively, you can hide all elements that are not in a variant, by selecting the other variant mode button ( ).

Figure 16. Preview Variant: Grayed-out Elements



Elements not in Variant are grayed-out in tree.



Elements not in Variant are grayed-out in diagram.

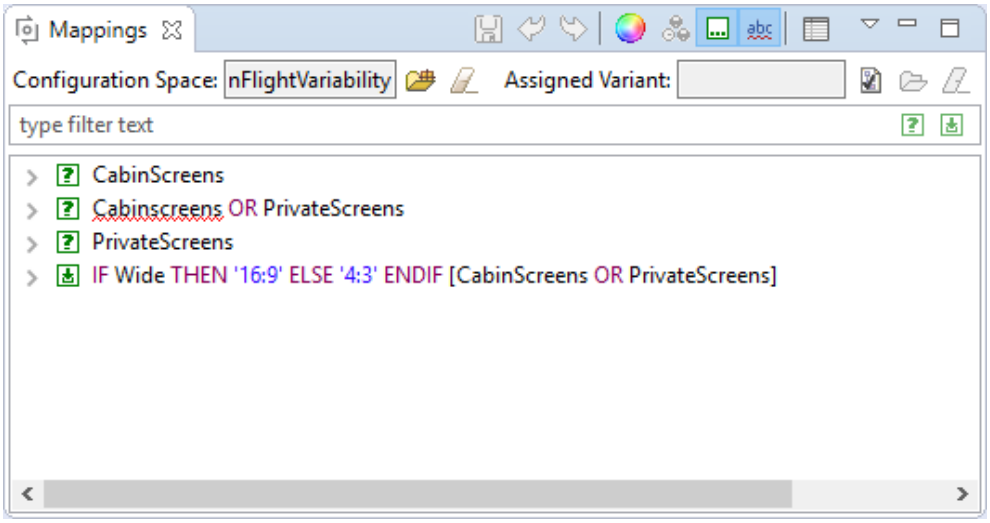
Highlight Condition of Capella Element

After selecting a Capella element, all conditions that it is mapped to are shown in bold letters in the **Mappings** view. Also all conditions that are propagated to this element are highlighted in bold letters (for details on propagation, see Section 4, “Capella Propagation Rules”).

Find Errors in Conditions or Calculations

Errors in conditions or calculations can be highlighted via button **Show Errors** ( ). For example, feature `Cab\inScreens` is misspelled in the second condition. Therefore it is underlined. Hover over the faulty condition or calculation to see the error message.

Figure 17. Underlining Errors in Conditions or Calculations

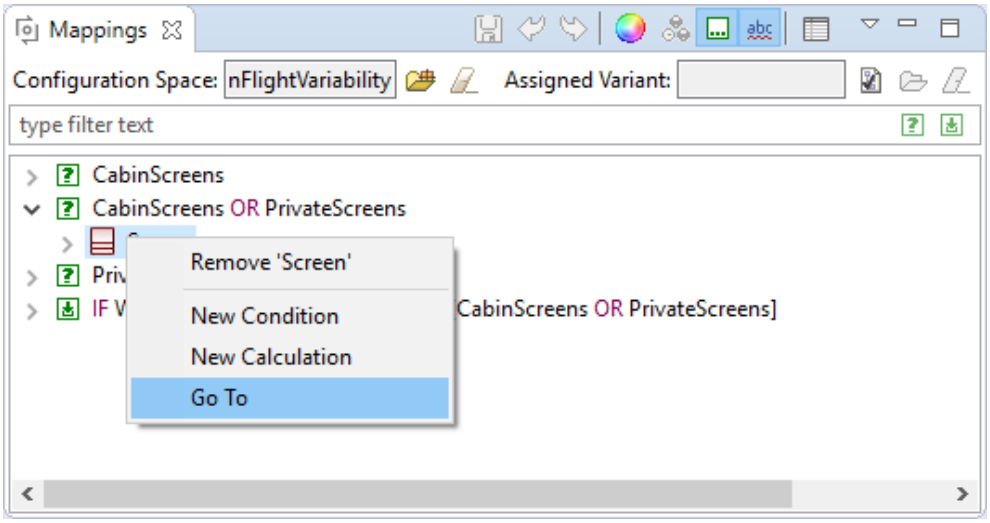


3.4. Useful when working with the Mappings View

Navigate to a Capella Element

You can navigate to any Capella element shown in the **Mappings** view by using the **Go To** context menu. Whereever the element is selectable, it will be shown and selected.

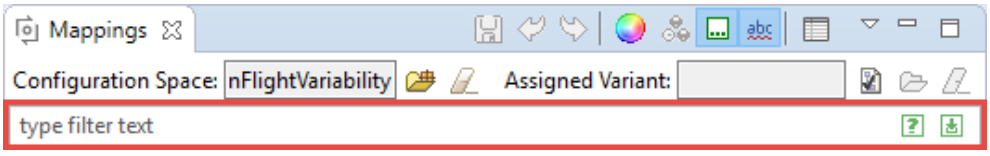
Figure 18. Navigate to Capella Element by Context Menu



Filter Conditions or Calculations

Conditions and calculations shown in the **Mappings** view can be filtered. To filter all elements shown in the mapping view by text, type part of the their text in the marked text box (can be either part of the pvSCL expression, or the name of a Capella element). Only the matching elements will be shown in the view.

Figure 19. Underlining Errors in Conditions or Calculations



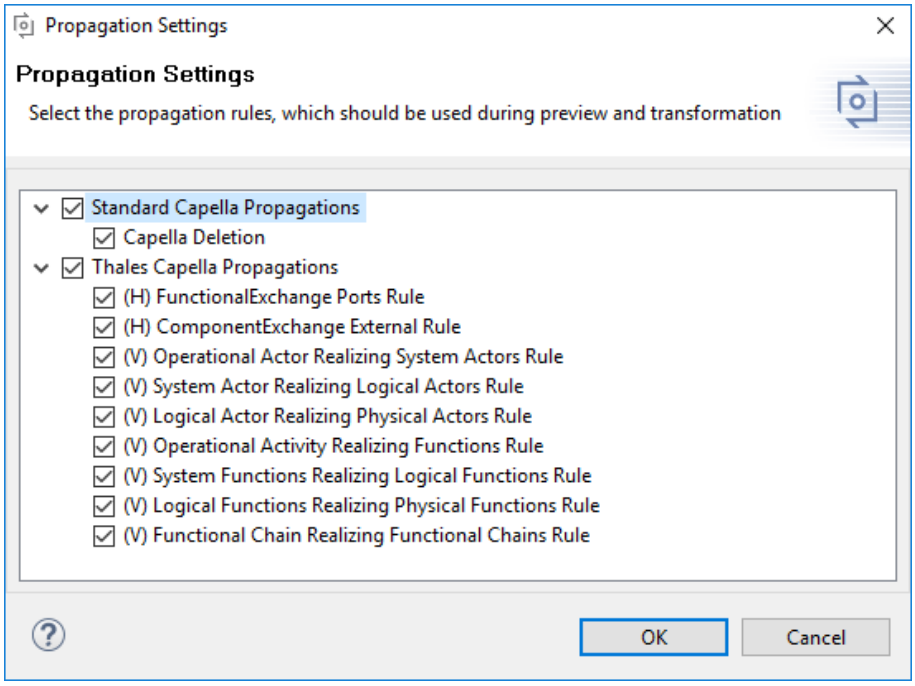
To show only conditions and hide all calculations, use button at the end of the text box. For showing only calculations, press .

Enable Propagation Rules

In Capella, different element types are semantically related to each other. Propagation rules utilize these semantic relations to simplify the mapping of Capella elements to conditions. Basically a propagation rule ensures that if Element A is removed during transformation, also Elements B, C, and D are removed. This influences also visualizations and decorations.

In Capella multiple propagation rules are supported. See Section 4, “Capella Propagation Rules” for a documentation of all rules. Each rule can be enabled or disabled in the **Propagation settings** of the **Mappings** view (accessible via button) .

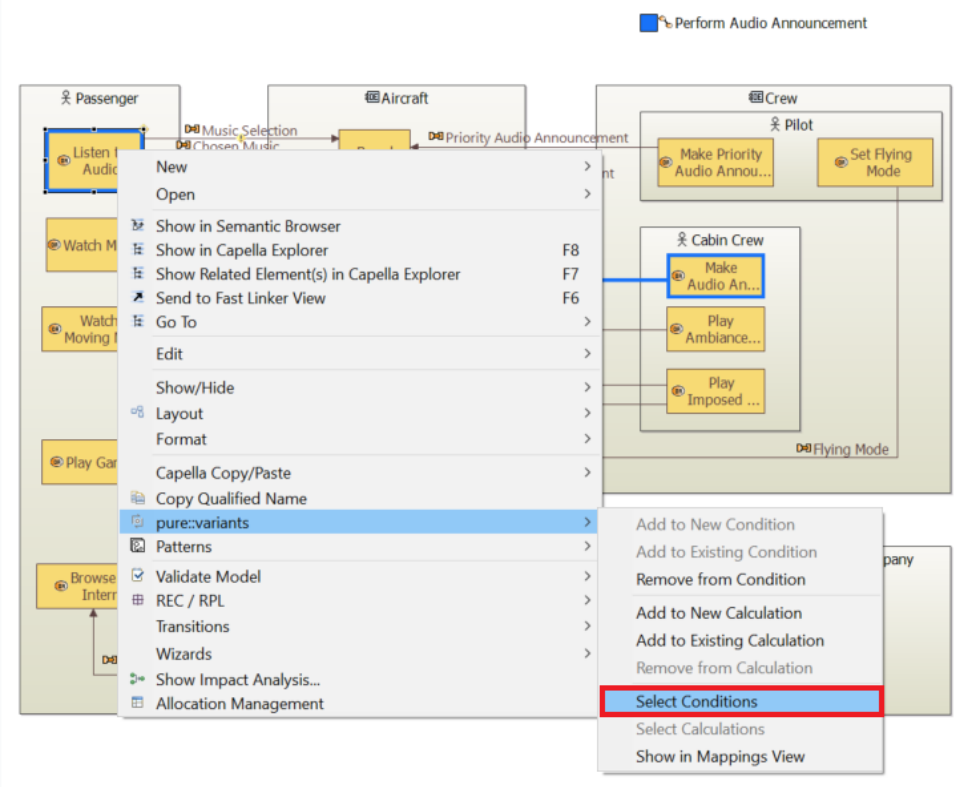
Figure 20. Propagation Settings



Select Calculations / Conditions

If an element is mapped to a condition or a calculation, it can be selected using the context menu in the EMF model. Also, all conditions and caluations that are propogated to this element are selected. To select the conditions or calculations, right-click on the selected element(s) in the EMF model (e.g., in diagram or browser) and select **pure::variants# Select Conditons** or **Select Calculations**. This will select the respective conditions or the calculations in the **Mappings** view.

Figure 21. Select Conditions on EMF element

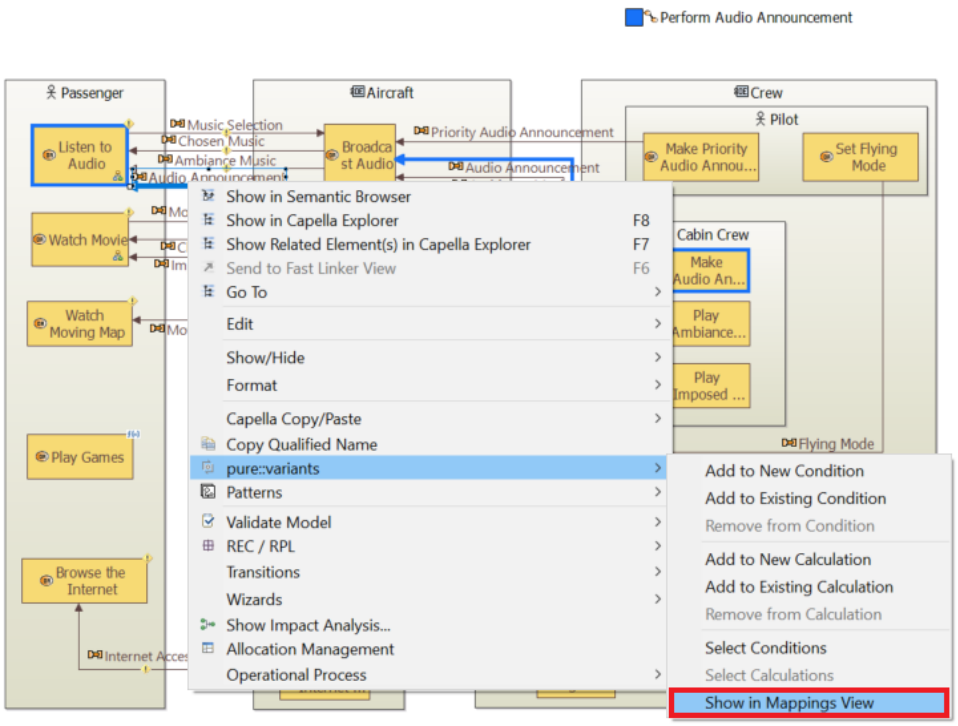


Alternatively, you can also right-click on the elements in the Capella Project Explorer to select the conditions and calulations.

Show in Mappings View

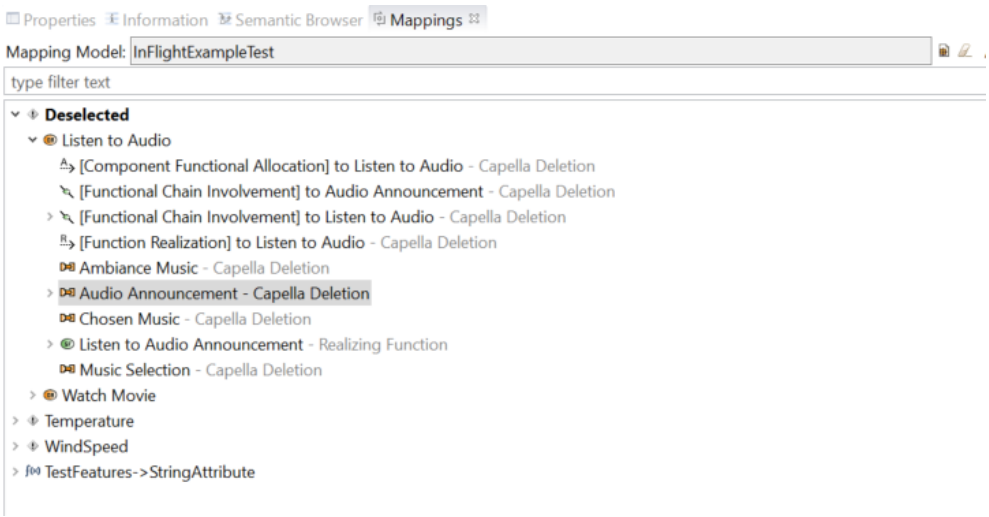
The elements mapped to the conditions or calculations can be selected in the **Mappings** view. The elements are selected under all the conditions and caluations that are propogated to this element. To select the element, right-click on the selected element(s) in the EMF model (e.g., in diagram or browser) and select **pure::variants# Show in Mappings View**.

Figure 22. Show in Mappings View on EMF element



Alternatively, you can also right-click on the elements in the Capella Project Explorer to select the elements in the mapping view.

Figure 23. Selected element shown in the Mappings view.

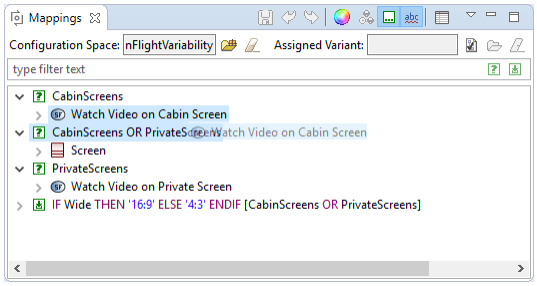


3.5. Changing Variability Information

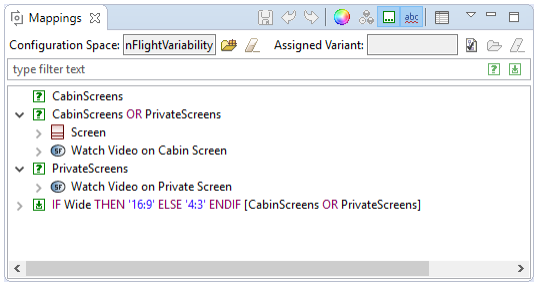
Change a Condition for a Single Element

Changing the condition on a single element can be done by dragging the mapped element from one condition to another. For this, a new condition can be created via the context menu (see the section called “Using Context Menu in Mappings View”) or an existing condition can be used.

Figure 24. Changing Conditon by Drag and Drop



Drag Element from a Condition and to new Condition

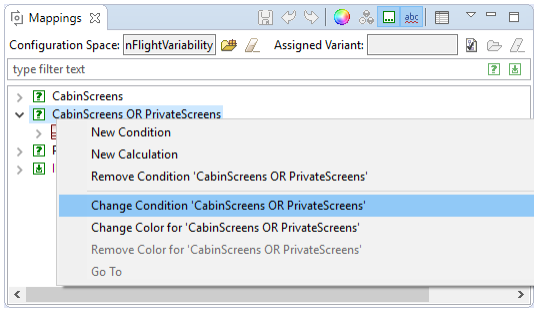


Drop Element on new Condition.

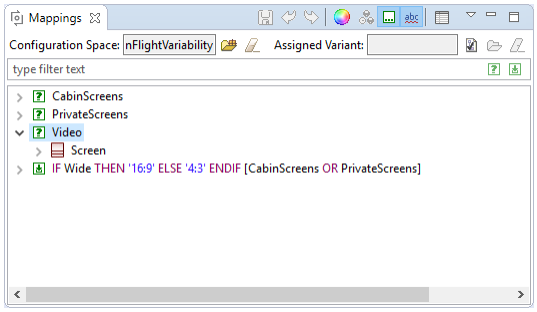
Change a Condition for all Elements

Select the condition in the **Mappings** view and double-click the condtion or select the **Change Condition** context menu. The condition editor with the current condition comes up. Edit the condition and press **OK**. The condition will be updated on all elements.

Figure 25. Changing Conditon via Editor



Open Context Menu on Condition
and Select 'Change Condition ...'



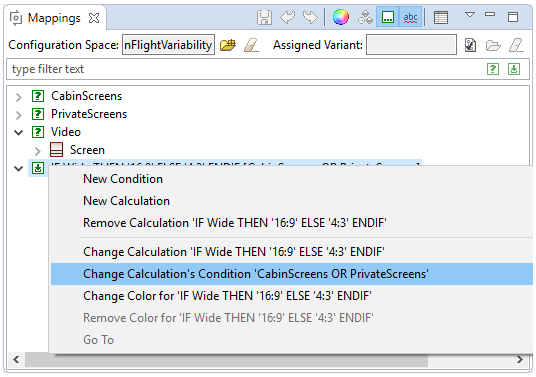
After editing Condition all Elements
are located beneath new Condition.

Changing a Calculation

Calculations can be changed in the same way as conditions. You can either drag & drop the mapped attribute from one calculation to another, or you can change the calculation itself via its **Change Calculation** context menu.

To change the calculation's condition, use context menu **Change Calculation's Condition**.

Figure 26. Changing a Calculation's Condition



4. Capella Propagation Rules

There are multiple propagation rules available for Capella. All propagation rules can be enabled or disabled separately. If an element is removed during transformation all enabled propagation rules are asked whether there are

any more elements to be removed. For the resulting elements the enabled propagation rules are applied again until no new elements are added to the result anymore.

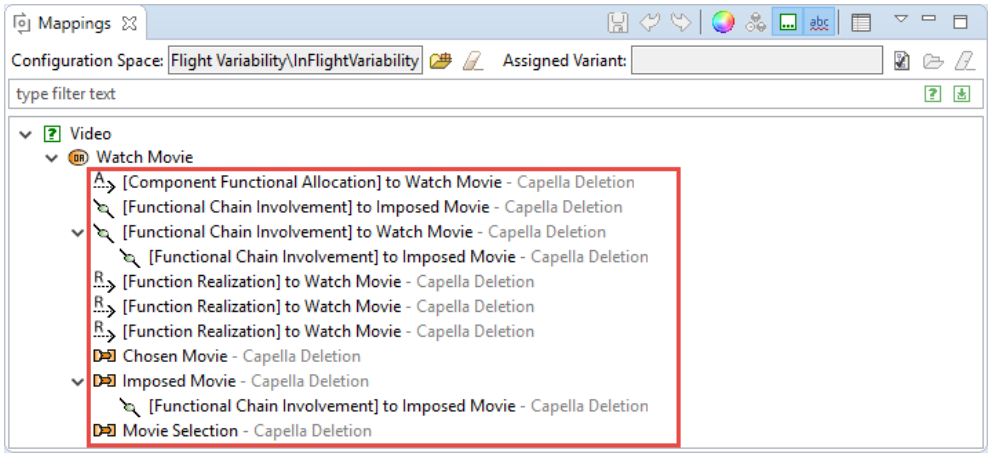
To understand which condition is propagated to which element and why, you can expand the mapped elements further in the **Mappings** view. This shows the next level of propagated elements and the name of the propagation rule. See Figure 27, “Capella Deletion Propagation in Mappings View” for an example.

All propagation rules below the "Thales Capella Propagations" category are prefixed with either an (H) or a (V). (H) stands for *horizontal*, which means that it relates to propagation of elements on the same Capella layer. (V) stands for *vertical*, which means that it relates to propagation of elements on multiple Capella layers.

4.1. Capella Deletion

The *Capella Deletion* propagation rule replicates the behavior of a Capella delete operation. It returns all other elements that are removed automatically by Capella when an element is deleted. Thus it is ensured that variant preview and transformation result match, even though during preview no actual deletion is done. This propagation rule is always enabled.

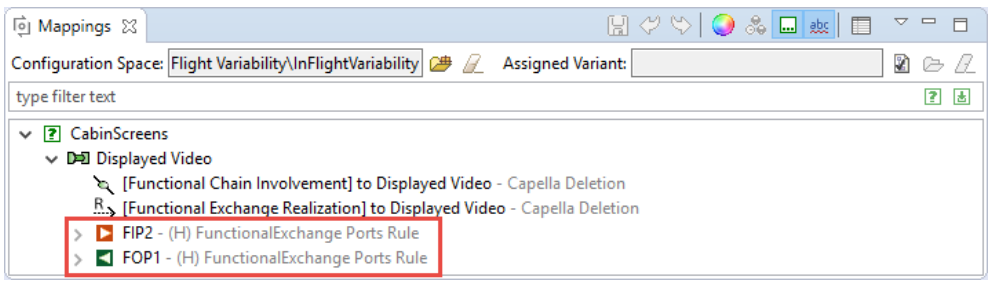
Figure 27. Capella Deletion Propagation in Mappings View



4.2. Functional Exchange Ports

The *(H) FunctionalExchange Ports Rule* ensures that for each functional exchange that is removed during transformation also the input and output ports are removed if no other functional exchange uses them.

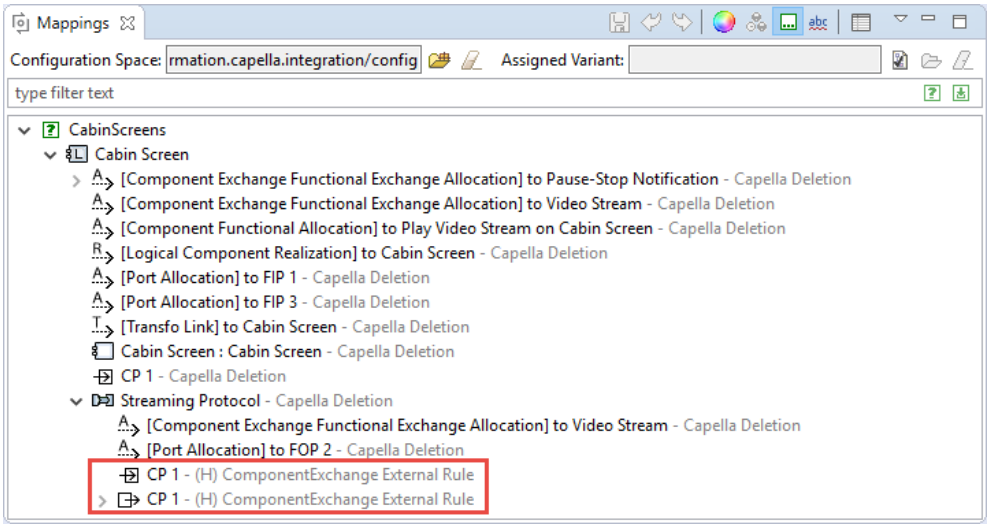
Figure 28. FunctionalExchange Ports Rule in Mappings View



4.3. Component Exchange Ports

The *(H) ComponentExchange External Rule* ensures that for each component exchange that is removed during transformation also the input and output ports are removed if no other component exchange uses them.

Figure 29. ComponentExchange Ports Rule in Mappings View

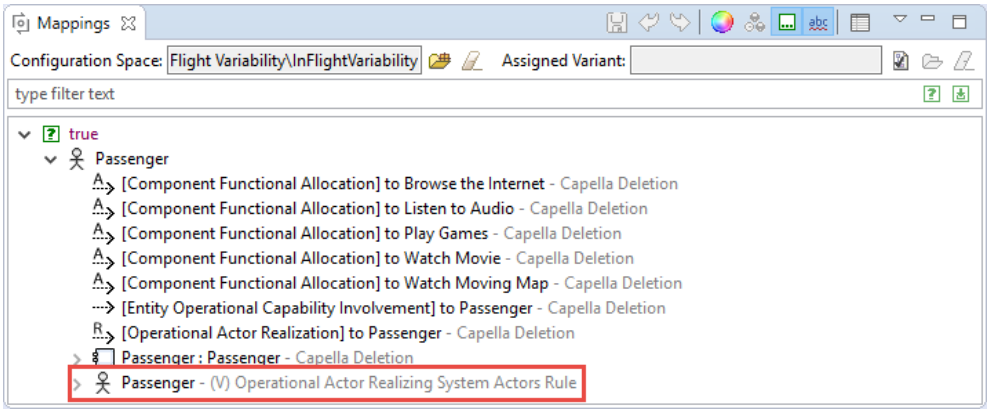


4.4. Realizing Actors

There are three rules concerning realizing actors: *(V) Operational Actor Realizing System Actors Rule*, *(V) System Actor Realizing Logical Actors Rule* and *(V) Logical Actor Realizing Physical Actors Rule*.

(V) Operational Actor Realizing System Actors Rule ensures that for each operational actor, that is removed during transformation, all realizing system actors are also removed. Only if the realizing system actor realizes another operational actor, that is still part of the transformation result, the system actor will stay in the output model. The other two rules ensure the same for layers System Analysis, Logical Architecture and Physical Architecture.

Figure 30. Operational Actor Realizing System Actors Rule in Mappings View

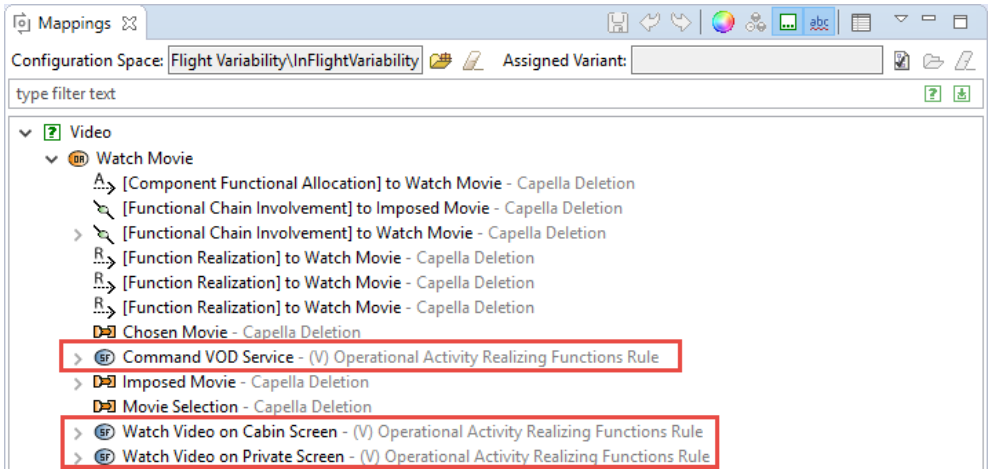


4.5. Realizing Functions

There are three rules concerning realizing functions: *(V) Operational Activity Realizing Functions Rule*, *(V) System Functions Realizing Logical Functions Rule* and *(V) Logical Functions Realizing Physical Functions Rule*.

(V) Operational Activity Realizing Functions Rule ensures that for each operational activity that is removed during transformation, all realizing system functions are also removed. Only if the realizing system function realizes another operational activity, that is still part of the transformation result, the system function will stay in the output model. The other two rules ensure the same for layers System Analysis, Logical Architecture and Physical Architecture.

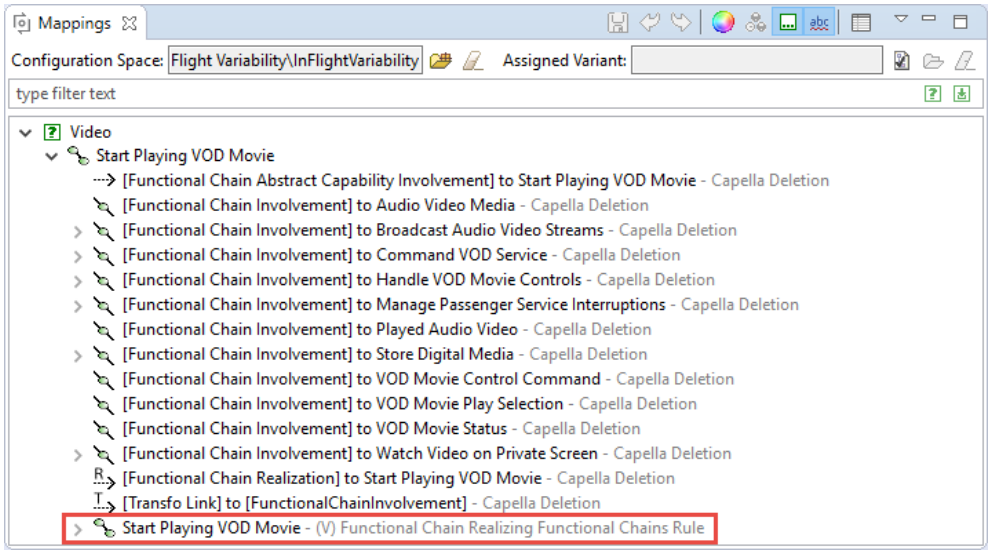
Figure 31. Operational Activity Realizing Functions Rule in Mappings View



4.6. Realizing Functional Chains

The *(V) Functional Chain Realizing Functional Chains Rule* ensures that for each functional chain that is removed during transformation also all realizing functional chains are removed. Only if the realizing functional chain realizes another functional chain, that is still part of the transformation result, the realizing functional chain will stay in the output model.

Figure 32. Realizing Functional Chains Rule in Mappings View

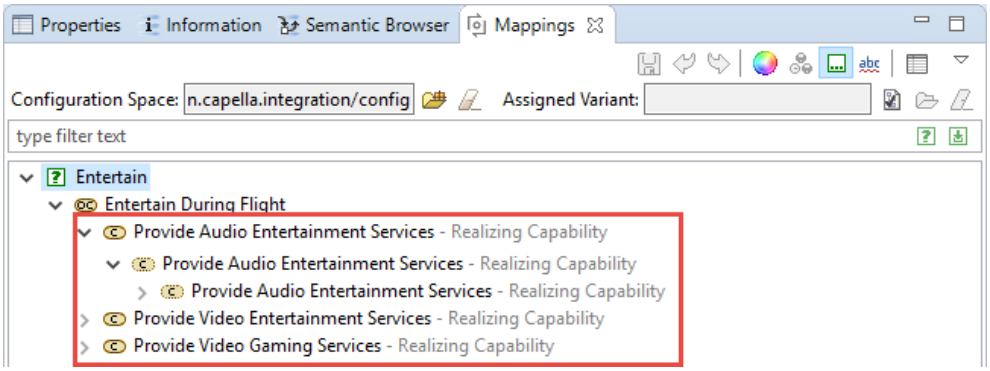


4.7. Realization Capabilities

The *Realizing Capability* propagation works in the same way as the Realizing Function, Actor and Functional Chain propagation. The only difference is that it applies to capabilities. When a capability is removed during transformation, each realizing capability is also removed if it does not realize any other capabilities.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 33. Realizing Capability Propagation

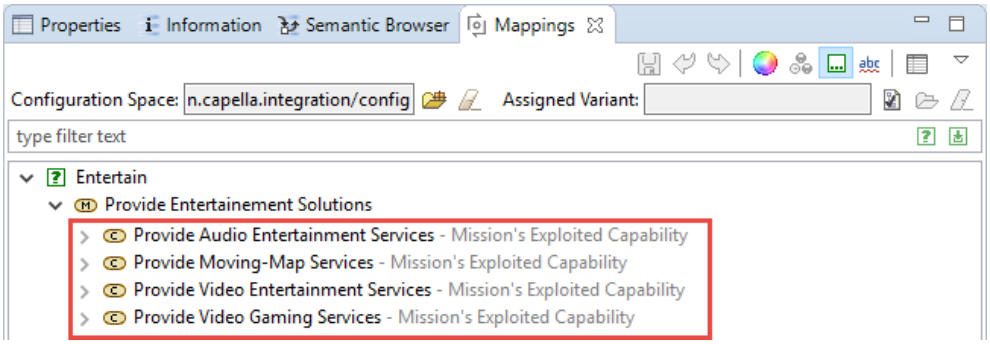


4.8. Mission's Exploited Capabilities

When a mission is removed during transformation, each exploited capability is also removed if it is not exploited by any other mission.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 34. Mission's Exploited Capabilities Propagation

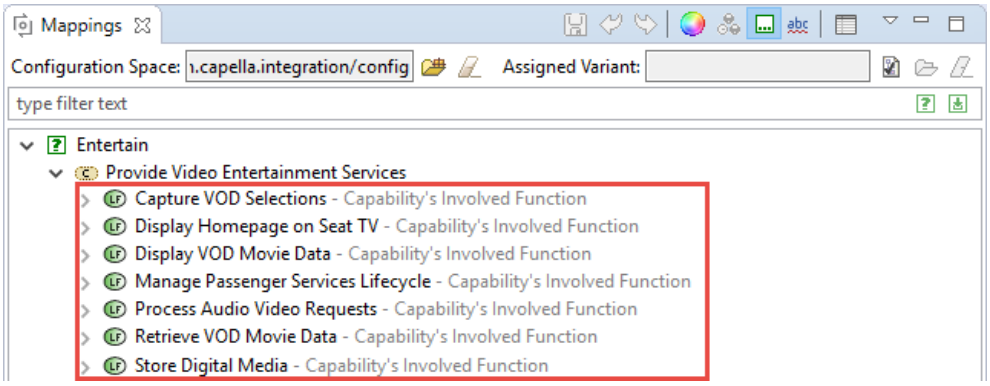


4.9. Capability's Involved Function

When a capability is removed during transformation, each of its involved functions is also removed if it is not involved in any other capability.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 35. Capability's Involved Function Propagation

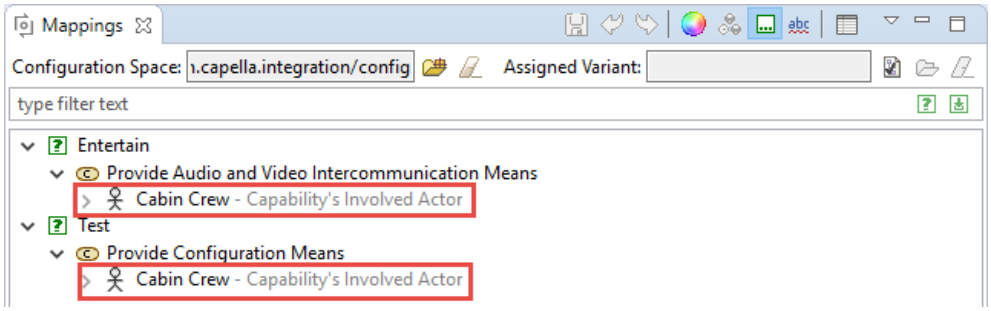


4.10. Capability's Involved Actor

When a capability is removed during transformation, each of its involved actors is also removed if it is not involved in any other capability.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 36. Capability's Involved Actor Propagation

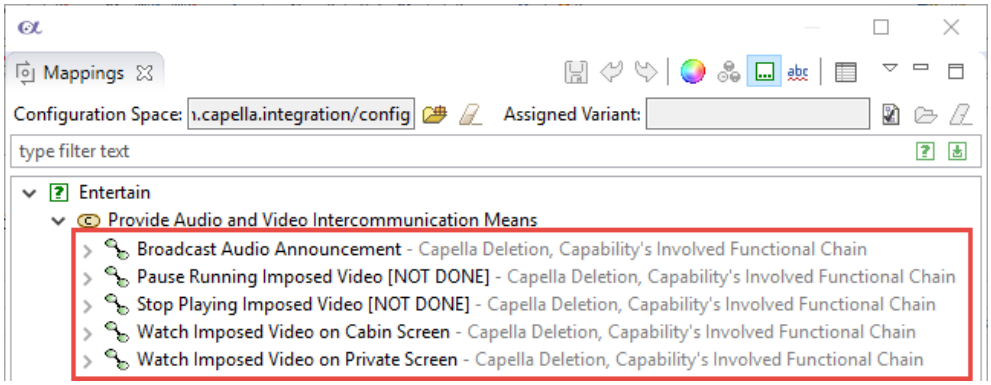


4.11. Capability's Involved Functional Chain

When a capability is removed during transformation, each of its involved functional chains is also removed if it is not involved in any other capability.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 37. Capability's Involved Functional Chain Propagation



4.12. Capability's Included Capability

When a capability is removed during transformation, each of its included capabilities is also removed if it is not included by any other capability.

Please note that for the sake of simplicity all Capella Deletion propagations have been removed from the image.

Figure 38. Capability's Included Capability Propagation