


Predefined Relation Maps

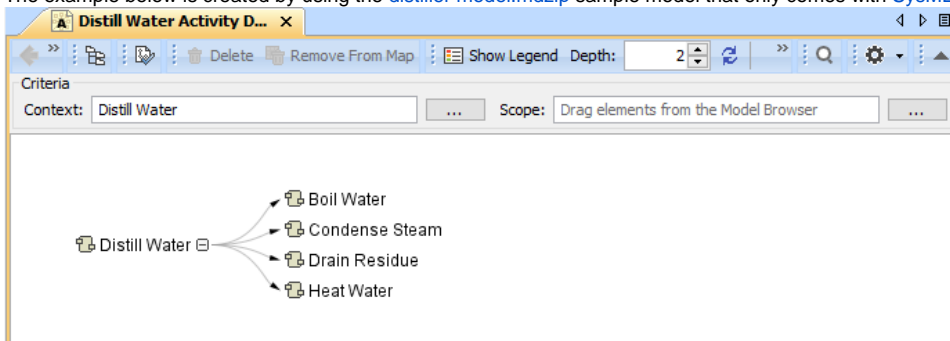
You can use all predefined relation maps to represent the traceability of system requirements and design elements. The main purpose of relation maps is to review and analyze relations among the elements and create new elements directly in the relation map. It is a special kind of diagram that automatically updates and renders an element's dependency tree according to predefined dependency criteria. You can create five kinds of predefined relation maps:

- [Activity Decomposition Map](#).
- [Structure Decomposition Map](#).
- [Instance Map](#).
- [Requirement Containment Map](#).
- [Requirement Derivation Map](#).

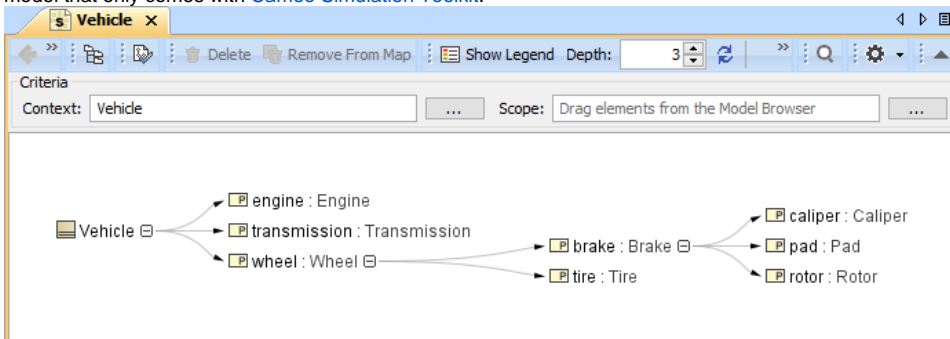
 You can create predefined relation maps only if you have the [SysML Plugin](#) installed. [How to install SysML Plugin >>](#)

The different purposes for each relation map are illustrated below:

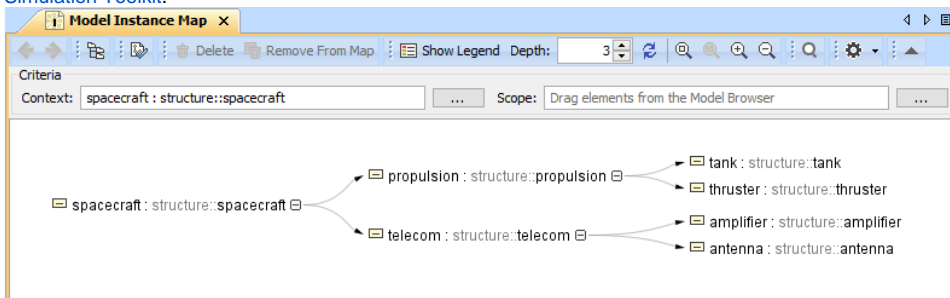
- **Activity Decomposition Map** displays an Activity decomposition of the selected context. You can review, analyze, and decompose the Activities. The example below is created by using the [distiller model.mdzip](#) sample model that only comes with [SysML Plugin](#).



- **Structure Decomposition Map** displays the decomposition of the [Part Properties](#) related through the [Composition relationship](#) of the specified context. You can review, analyze, and decompose the Part Properties. The example below is created by using the [VehicleStructure.mdzip](#) sample model that only comes with [Cameo Simulation Toolkit](#).



- **Instance Map** displays a hierarchy of the [Instance Specifications](#) of the selected context. You can review and analyze the hierarchy of Instance Specifications. The example below is created by using the [SpacecraftMassRollup_HTMLTable.mdzip](#) sample model that only comes with [Cameo Simulation Toolkit](#).



- **Requirement Containment Map** displays the decomposition of the [Requirements](#) related through the [Containment relationship](#) of the specified context. You can review, analyze, and decompose the Requirements. The example below is created by using the [hybrid sport utility vehicle.mdzip](#)

The screenshot displays the SysML Editor interface for a model named 'HSUV Specification Requirements'. The main workspace shows a hierarchical tree diagram of requirements. The root node is 'HSUV Specification' (represented by a folder icon). It branches into five nodes: '4 Capacity', '1 Eco-Friendliness', '3 Ergonomics', '2 Performance', and '5 Qualification'. Each of these nodes further branches into more specific requirements: '4 Capacity' branches into '4.1 CargoCapacity', '4.2 FuelCapacity', and '4.3 PassengerCapacity'; '1 Eco-Friendliness' branches into 'R1.2.1 Emissions'; '3 Ergonomics' has no children; '2 Performance' branches into '2.4 Acceleration', '2.1 Braking', '2.2 FuelEconomy', and '2.3 OffRoadCapability'; and '5 Qualification' branches into '5.1 SafetyTest'. The interface includes a toolbar with icons for navigation and editing, and a 'Criteria' section with 'Context' and 'Scope' fields.

```

graph LR
    HSUV[HSUV Specification] --> 4[4 Capacity]
    HSUV --> 1[1 Eco-Friendliness]
    HSUV --> 3[3 Ergonomics]
    HSUV --> 2[2 Performance]
    HSUV --> 5[5 Qualification]
    4 --> 4.1[4.1 CargoCapacity]
    4 --> 4.2[4.2 FuelCapacity]
    4 --> 4.3[4.3 PassengerCapacity]
    1 --> R1.2.1[R1.2.1 Emissions]
    3 --> 
    2 --> 2.4[2.4 Acceleration]
    2 --> 2.1[2.1 Braking]
    2 --> 2.2[2.2 FuelEconomy]
    2 --> 2.3[2.3 OffRoadCapability]
    5 --> 5.1[5.1 SafetyTest]
  
```

-
- The screenshot displays the Hsuv Specification Requirements tool interface. At the top, there is a toolbar with various icons for navigation and editing. Below the toolbar, a search bar is visible. The main area shows a hierarchical tree of requirements. The root node is '2 Performance', which is expanded to show its sub-requirements: '2.1 Braking', '2.2 FuelEconomy', and '2.3 OffRoadCapability'. Each of these sub-requirements is further expanded to show its own set of sub-requirements. For example, '2.1 Braking' has '2.1.1 RegenerativeBraking' and '2.1.2 RegenerativeBraking'. '2.2 FuelEconomy' has '2.2.1 RegenerativeBraking', '2.2.2 Range', and '2.2.3 PowerSourceManagement'. '2.3 OffRoadCapability' has '2.3.1 RegenerativeBraking', '2.3.2 Range', and '2.3.3 PowerSourceManagement'. A legend in the top right corner indicates that blue lines represent 'Derived' requirements and green lines represent 'Owned' members. The interface also includes a 'Criteria' section with a 'Context' dropdown set to 'Performance' and a 'Scope' dropdown set to 'Drag elements from the Model Browser'.

Additional relation map
You can also create a Relation Map Diagram if you want a rapid review, analysis, and creation of relationships among the elements of the entire model.

The following procedures explain how to work with relation maps.

- \$body