ENTERPRISE ARCHITECT

用户指南系列

数学模拟

Author: Sparx Systems
Date: 13/11/2024
Version: 17.0

创建于 ENTERPRISE ARCHITECT

目录

数学模拟

Enterprise Architect提供了广泛的选项，可将高级数学工具和功能引入您的模拟。

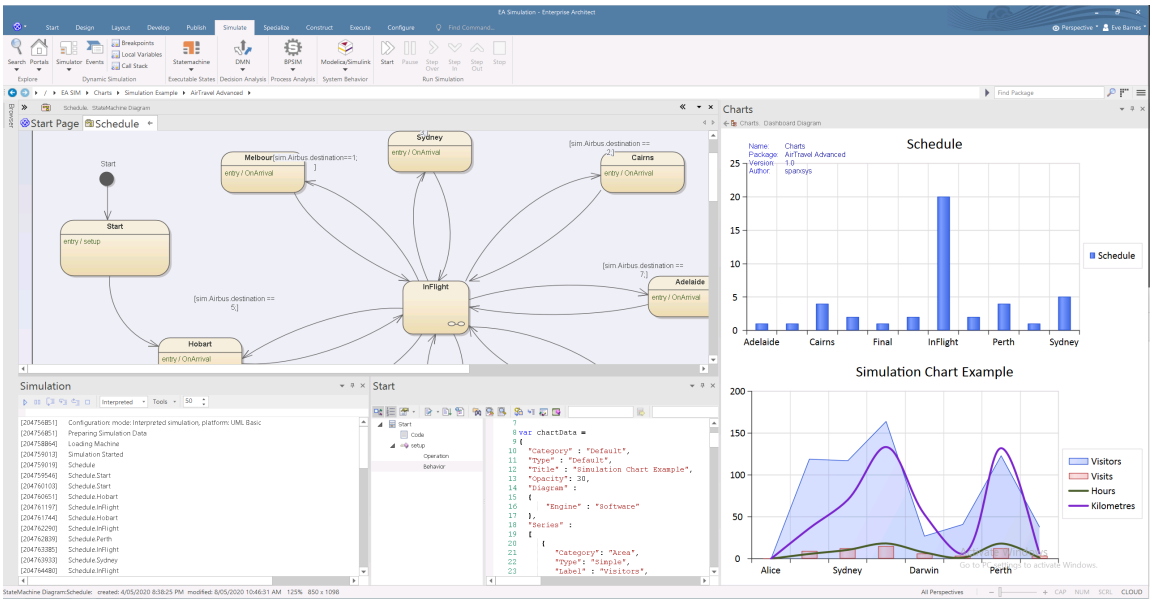
您可以通过使用求解器类将 MATLAB 等集成外部工具的强大功能引入您的模型，也可以导出您的模型以在其他外部工具（例如 MATLAB Simulink、状态流和 Simscape 或 OpenModelica）中执行。

Enterprise Architect在JavaScript引擎中包含一个广泛的数学函数库，提供显着扩展的仿真功能的好处。

Enterprise Architect还提供范围广泛的动态图表；无需外部工具，您可以配置这些图表以从Enterprise Architect中直接执行的模拟中提取和绘制信息。

探索：

- Enterprise Architect中的求解器类调用 MATLAB 或 Octave 将复杂的数学合并到基于模型的仿真中
- 基于流行的 Cephes函数库的广泛的内部数学库
- 与 OMG SysPhS 标准集成，使您能够配置模型以导出到常用工具
- 支持将模型导出到 MATLAB Simulink、Simscape 和状态流；您可以在Enterprise Architect中创建模型并在 MATLAB 中执行它
- 对 Modelica 的广泛支持；您可以在Enterprise Architect中创建和配置您的模型并在 Modelica 中执行它
- 在专用的图形演示工具中或通过Enterprise Architect的动态图表功能以图表格式演示建模和模拟的结果



可用的集成

这些数学建模工具可在Enterprise Architect ：

产品	描述
MATLAB	MATLAB 是一种流行且广泛使用的数值计算环境和编程语言，由 MathWorks 开发。它提供了丰富的数学表达式和公式，可以在应用程序本身内处理或调用到其他应用程序（如Enterprise Architect ）中。 Enterprise Architect的 MATLAB 集成通过 MATLAB API 连接，允许您的 Enterprise Architect仿真和其他脚本根据所选 MATLAB 函数和表达式的值执行。您可以通过 Solver 类调用 MATLAB，或将您的模型导出到 MATLAB Simulink、Simscape 和/或状态流。 注记：与 MATLAB集成需要 MATLAB R2018b 或更高版本。

Simulink	Simulink 是一个核心 MATLAB 应用程序，用于在 Block 之间运行定向消息的 SysML 仿真。 Enterprise Architect可以将 SysML模型转换为 Simulink 格式，自动运行仿真，并将所选变量的输出绘制为图表。您还可以直接在 Simulink 中打开生成的 Simulink 文件，允许您修改和微调仿真设置和输出功能。 您可以直接从Enterprise Architect Simulink 模式中拖放常见的内置 Simulink 库模块，或者使用新的 SysPhS 标准原型参数引用您自己的定制模块。 Simulink 是 OpenModelica 的替代选项，用于在Enterprise Architect中开发和运行仿真。
Simscape	Simscape 是 MATLAB Simulink 的可选扩展，允许您对物理系统进行模型，并指示 MATLAB 使用 Simscape 跨许多不同物理域的大量库模块来模拟和绘制请求的输出。 Enterprise Architect可以将 SysML内部块图转换为 Simscape。
状态流	状态流也是 MATLAB Simulink 的可选扩展，提供生成 MATLAB状态流图以在运行下运行的能力。在Enterprise Architect中，这可以帮助您使用在 Enterprise Architect中建模的状态机来指导您的 SysML 模拟，这些状态机被转换为状态流图。
模型	Modelica 是一种用于建模、模拟、优化和分析复杂动态系统的开放语言标准。它定义并提供了一个文件结构，可以由源（免费开放源）和 Dymola 和 Wolfram Modeller（市售；这些可以与Enterprise Architect一起使用但尚未与 Sparx Systems软件测试或集成）等应用程序访问和操作）。
OpenModelica	OpenModelica是一个基于Modelica开放语言标准的免费源环境；OpenModelica 使您能够读取、编辑和模拟 Modelica 文件。 Enterprise Architect与 OpenModelica 集成，并支持在 SysPhS 标准下使用它来定义状态机图和参数图模拟中的常量和变量。 您还可以在 OMEdit - OpenModelica 连接编辑器中的Enterprise Architect中显示模型中的 SysML块图，该编辑器显示块的别名和注记。 OpenModelica 是 Simulink 的替代选项，用于在Enterprise Architect中开发和运行仿真。
GNU Octave	GNU Octave是一个数学函数库。通过Enterprise Architect的JavaScript引擎，您可以与 Octave 解释器集成以使用任何可用的 Octave 函数。 Octave 提供了 MATLAB 函数的替代方案，特别强调序列和矩阵。
JavaScript Math Library	JavaScript Math Library是图表数学库的一个实现，它直接内置到Enterprise Architect中的JavaScript中，以方便在脚本仿真（或基于动态图形、模型的插件 ）中使用高级数学函数插件 或许多其他场景）。

求解器

产品	描述
求解器类	Solver类为各种外部工具提供通用 API；它在Enterprise Architect使用的任何 JavaScript引擎中都可用，并且在从 MATLAB 或 Octave 调用数学函数时具有特殊价值。您可以在外部工具中审阅处理结果，或将它们带入JavaScript引擎以在Enterprise Architect中呈现。

MATLAB	MATLAB 求解器在您的计算机上安装了 MATLAB 时可用。求解器使用 MATLAB API 提供对大量可用 MATLAB 函数的访问。
八度	Octave 求解器在您的计算机上安装 Octave 时可用。求解器直接与 Octave 解释器通信，以允许您访问 Octave 环境中的函数和数据。

配置模拟

类型	描述
配置工件	工件配置是一种专门设计的Enterprise Architect中指定工件模拟的特征和参数。您可以通过配置SysML仿真窗口设置规范。
SysML扩展用于物理和信号流仿真（交互）标准	SysPhS 标准提供了一种更简单、基于模型的方法来共享模拟、在每个元素中定义变量、常量和初始值，而不是通过配置文件。这使得建立模拟的可视化方法成为可能，因为变量、常量和初始值可以在 SysML 块的其他隔间中的图表中可见。
定义多个数据集	可以针对在参数模型中的仿真配置中使用的 SysML 块定义多个数据集。这允许使用相同的 SysML模型进行可重复的模拟变化。

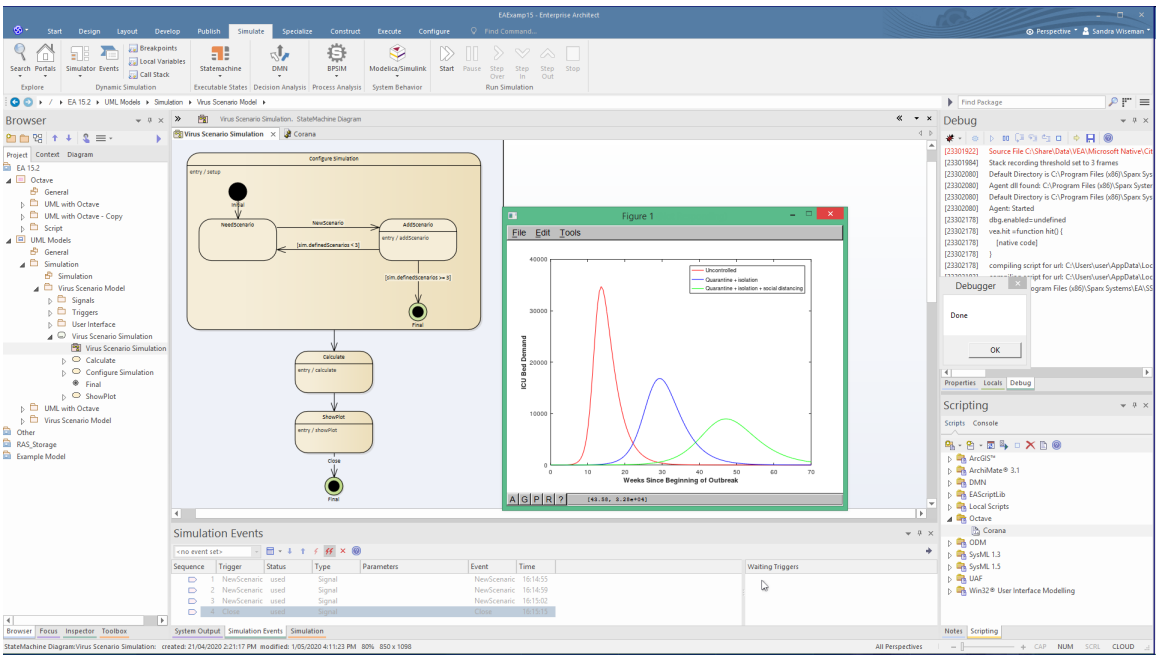
公共使用案例

名称	描述
求解器控制台	• 快速测试要在脚本或模拟中使用的命令 • 调用一个 MATLAB函数来查看它是否返回您所期望的并且运行时没有错误 • 剪切和粘贴只需要运行一次的JavaScript片段，而不是创建脚本然后删除它
模拟中的求解器	• 调用一个复杂的数学函数，定义为 Octave函数 • 调用MATLAB的API例程来确定决策流程
SysML SysPhS仿真	• 在Enterprise Architect模型中模拟新的汽车 ABS 系统并使用 Simulink 进行仿真 • 在Enterprise Architect中设计模型和液压系统，并使用现有的 Modelica 组件库在 OpenModelica 中模拟系统
在生成中生成 StateChart 并进行细化和调试	• 创建SysML状态机，快速定义用户反复开启和关闭系统的动作；生成模拟并在状态流中打开以 "实时"查看状态参数并调整状态流设置
模型和测试一个 StateChart	• 在状态流模型中模拟之前的模型StateChart

求解器

从Enterprise Architect 15.2 版开始，您可以使用JavaScript调用 求解器”，它提供与 MATLAB 和 Octave 等外部工具的集成。 Solver类提供与您的外部数学工具的实时连接，在Enterprise Architect中执行模拟期间，您可以通过该工具调用复杂的高阶数学函数。

求解器可帮助您将 MATLAB 和 Octave 的计算能力带入Enterprise Architect的仿真、基于模型的插件和自定义脚本中。 Solver类是Enterprise Architect的JavaScript引擎中的内置结构。



使用开始器开始

Solver类变量可以在每次模拟中创建一次，或者，如果需要，可以创建多个 Solver类实例。请注记，启动一个新的 Solver类实例有时可能需要几秒钟。通常建议在模拟开始时创建一个 Solver 实例，并在需要时重用它。

要使用 Solver类，您需要了解首选数学库中可用的函数以及它们使用的参数，如库产品文档中所述。

您首先定义要在脚本中使用的数学库（或库）。对于 MATLAB，您键入：

var matlab = new Solver('matlab');

对于八度，您键入：

var octave = new Solver('octave');

然后，无论您需要在脚本中使用数学函数的什么地方，对于 MATLAB，您键入：

matlab.exec('complexMathsFunction', 参数);

对于 Octave，您键入：

octave.exec('complexMathsFunction',参数);

这两行脚本在适当的工具中执行函数并在那里显示结果。如果要将结果带回Enterprise Architect ，请在该行之前添加：

var resultFrom'工具名';

那是：

var resultFromMatlab = matlab.exec('complexMathsFunction', parameter1, parameter2);或者

var resultFromOctave = octave.exec('complexMathsFunction', parameter1, parameter2);

作为JavaScript引擎的一部分，Solver Classes 也可以立即被插件

访问插件

创建基于模型的JavaScript插件的作家。

注记：如果在一段JavaScript中多次调用创建一个新的 Solver，模拟性能会大大降低（例如，在多次进入的状态机节点上）。

有关这两个求解器的更多信息，请参阅*GNU Octave Solver*和*MATLAB Solver*帮助主题。

求解器构造函数

构造函数	描述
Solver(string solverName)	创建一个连接到指定助手应用程序的新实例的新求解器。

求解器方法

方法	描述
get(string name)	从求解器环境中检索命名值。
set(string name, object value)	为求解器环境中的命名变量分配一个新值。
exec(string name, string	执行一个命名函数。实际功能将取决于所使用的求解器类型。

arguments, int returnValues)	

求解器控制台

Solver Console 是一个易于访问且易于使用的编辑器。它主要用于管理 MATLAB 和 Octave Solvers，提供对这些外部应用程序支持的测试功能的简单快速访问。 Solver Console 可用于创建、检查和维护模拟中使用的调用，以及使用JavaScript编写的脚本。

访问

功能区	仿真> 控制台> 求解器> MATLAB 仿真> 控制台 > 求解器 > Octave

求解器初始化

选择的 Solver 会自动设置为以下之一：

MATLAB：

- var matlab = new Solver("matlab");

八度：

- var octave = new Solver("octave");

如果可以访问后台 Octave 或 MATLAB，控制台将提示启动细节并准备好接受任何输入。

用途

控制台接受将一次执行一个的单行JavaScript命令。在底部的文本输入部分输入新命令，然后按 Enter 键执行命令。

执行的命令将被添加到主输出面板，以及命令的任何输出。

然后可以使用求解器类函数访问求解器；例如：

软件：

- matlab.set('<变量名>', <值>)
- matlab.get('<变量名>')
- matlab.exec(<函数>, <参数>)

八度：

- octave.set('<variable_name>', <value>)
- octave.get('<variable_name>')
- octave.exec(<函数>, <参数>)

有关详细信息，请参阅八度求解器和*MATLAB* 求解器帮助主题。

可以通过按向上箭头键或向下箭头键重新使用以前输入的命令。

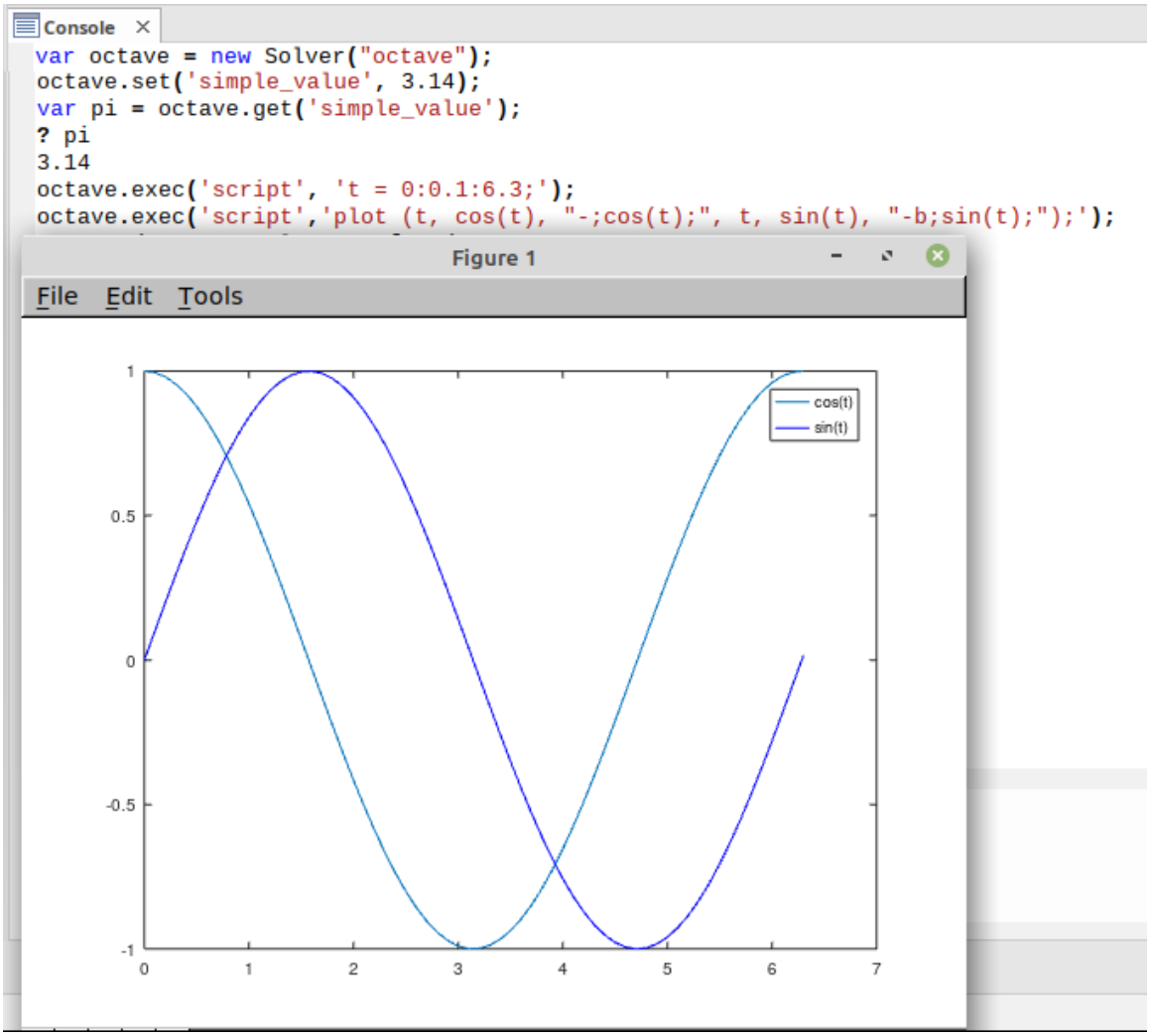
输出

要输出变量或函数的值，您可以使用：

- ? <变量名>
- ? <功能>

示例

此图显示了在 Octave 控制台中使用 Octave Solver 的 Set、Get 和 Exec 以及使用 ? 的输出的一些示例：



Octave 图是两个 ocatave.exec() 语句的生成输出。

控制台命令

控制台命令前面有 !字符并指示控制台执行操作。控制台命令包括：

- !clear - 清除控制台显示
- !save - 将控制台显示保存到文件
- !help - 打印命令列表
- !close - 关闭控制台
- !include <scriptname> - 执行指定的脚本项； scriptname 的格式为 GroupName.ScriptName（名称中允许使用空格）
- ? - 列出命令（与!help相同）
- ? <变量或函数> - 输出值

模拟中的求解器

模拟中A求解器可帮助您包含来自数学模型的行为，根据需要查询状态以调整相应UML模型的行为。

创建和初始化求解器

求解A最好在模拟开始时创建一次。这使得它可以在整个模拟过程中使用，而不会导致重复构建时间的延迟。这也是加载任何所需模块、定义功能和定义模型初始状态的好时机。

在状态机模拟中，执行此初始化的好地方是影响转移

从初始状态开始，或在模拟期间未重新输入的状态的条目中。

在活动模拟中，您需要将行动添加到动作的影响中。

更新求解器模型

随着模拟的进行，您可以使用任何影响字段来更新数学模型的参数。这可能包括更新率或算法之间的变化。在状态机模拟中：

- A转移
  当特定流程需要更改时，可以使用影响
- 状态条目A用于确保在模拟处于某个状态时更改行为，无论它是如何到达那里的；这包括任何嵌套状态
- 状态退出A用于确保在模拟离开状态时更改行为，无论它如何离开
- 状态Do活动A行为类似于进入动作。

在活动模拟中，只有行动效果。

查询求解器模型

带着一个转移

守卫，调用*solver.get()* 。你也可以调用*solver.exec()*来调用一个没有副作用的函数。

GNU Octave求解器

GNU Octave是一个数学函数库，特别强调序列和矩阵。您可以将每个函数调用到用JavaScript编写的脚本中。

您可以在运行时从运行调用任意数学函数，使用一个名为 Solver类的简单构造，用JavaScript ；用于 Octave 的 Solver类可以调用外部 Octave 工具并将高级数学函数直接链接到您正在运行的模拟中。例如：

var octave = new Solver(octave);

var resultFromOctave = octave.exec('complexMathsFunction', parameter1, parameter2);

请参阅模拟中的帮助主题求解器。

特征包括：

- 从 Octave 调用向量、矩阵、数字和字符串
- Octave 向量作为JavaScript一维数组返回（而JavaScript一维数组作为 Octave 向量返回）
- Octave 矩阵作为JavaScript二维数组返回（ JavaScript二维数组作为 Octave 矩阵返回）
- 您可以使用以下方法从 Octave 检索变量值：
  octave.get(<名称>)
- 你可以使用JavaScript值调用任何 Octave函数：
  octave.exec(<name>, [<arguments>], 0/ 1)
- 所有参数都在JavaScript数组中传递
- 您也可以在JavaScript中使用结果；如果你想要一个结果，传递1 ，如果你不想要结果，传递 0
- 您可以使用以下命令执行任何 Octave 语句：
  octave.exec("脚本", <语句>, 0/ 1)

Octave 可作为 MATLAB 库的替代品，并且可以在与 MATLAB 相同的所有上下文中使用。

设置和配置

安装 Octave 后， Enterprise Architect将从注册表中读取位置以提供自动集成。

用途

任务	描述

赋值	使用命令；例如：
	octave.set("simple_value", 3.14);
	octave.set("example_sequence", [0, 1 , 2]);
	octave.set("身份", [[1 , 0], [0, 1]]);

建造	在JavaScript编辑器中，通过将 őctave″传递给 Solver 构造函数，创建一个连接到 Octave 的新 Solver。 var octave = new Solver("octave");

检索值	使用命令；例如：
	var simple_value = octave.get("simple_value");
	var example_sequence = octave.get("example_sequence");
	var identity=octave.get("identity");

调用函数	将函数的名称作为第一个参数传递给 Solver.exec。 将所有参数作为第二个参数传递给数组中的那个函数。 当你想要一个JavaScript内部函数返回的值时，传递一个非零值作为第三个参数。例如： var序列= octave.exec("linspace", [0, 10, 1001], 1);
执行语句	将 脚本"作为第一个参数的名称传递给 Solver.exec。 将string中的整个 Octave 语句作为第二个参数传递。 八度。执行 ('脚本' ， 't = 0:0。 1 ：6.3 ；'); 八度。 exec ('脚本' , 'plot (t, cos (t), "-; cos (t);", t, sin (t), "-b; sin (t);");');

跟踪()	跟踪（语句）方法允许您在模拟中的任意点打印出跟踪语句。这是在执行期间用附加输出信息补充仿真log的极好方法。
	跟踪() 输出被写入仿真窗口。
	这是使用跟踪() 方法的示例：
	octave.set('simple_value', 3.14);
	var pi = octave.get('simple_value');
	跟踪("PI = " + pi); // 将该值输出到仿真窗口

视频

Sparx Systems提供了一个使用 Solver 通过 Octave 生成绘图的 YouTube 视频。看：

- 用 Octave 绘图

MATLAB 求解器

MATLAB 是一种数值计算环境和编程语言，包含一个大型数学函数库，每个函数都可以从用JavaScript编写的脚本中调用。

您可以在运行时使用一个简单的构造（称为 Solver类）从 MATLAB 调用任意数学函数，该构造用JavaScript 。MATLAB A Solver类可以通过其 API 调用外部 MATLAB 工具，并将数学函数直接链接到您正在运行的仿真中。例如：

```
var matlab = new Solver("matlab");
var resultFromMatlab = matlab.exec('complexMathsFunction', parameter1, parameter2);
```

请参阅模拟中的帮助器帮助主题。

特征包括：

- 从 MATLAB 检索向量、矩阵、数字和字符串
- MATLAB 向量作为JavaScript一维数组返回（而JavaScript一维数组作为 MATLAB 向量返回）
- MATLAB 矩阵作为JavaScript二维数组返回（而JavaScript二维数组作为 MATLAB 矩阵返回）
- 您可以使用以下方法从 MATLAB 中检索变量值：
  matlab.get(<名称>)
- 您可以使用以下方法调用任何带有JavaScript值的 MATLAB函数：
  matlab.exec(<名称>, [<参数>])
- 所有参数都在JavaScript object中传递
- 您还可以在JavaScript中使用结果
- 您可以使用以下命令执行任何 MATLAB 语句：
  matlab.exec("脚本")

MATLAB 可作为GNU Octave库的替代品，并且可以在与GNU Octave相同的所有上下文中使用。

注记：与 MATLAB集成需要 MATLAB R2018b 或更高版本。

设置和配置

安装 MATLAB 后， Enterprise Architect将从注册表中读取位置以提供自动集成。

如果 MATLAB 没有自动加载，请将路径（如配置*SysML*仿真窗口帮助主题）设置为通过在 MATLAB 控制台中运行 "matlabroot"获得的值。

用途

使用	讨论
建造	通过将*"matlab"*传递给 Solver 构造函数，创建一个连接到 MATLAB 的新 Solver。那是： var matlab = new Solver('matlab');

赋值	使用 matlab.set函数赋值。例如：
	matlab.set('simple_value', 3.14);
	或者
	var myString = "这是一个示例string ";
	matlab.set('myString', myString);

检索值	使用 matlab.get函数从 MATLAB 中检索结果。例如：
	var simple_value = matlab.get('simple_value');
	var myString = matlab.get('myString');

调用函数	将函数的名称作为第一个参数传递给 Solver.exec。
	任何一个：
	• 对于采用单个参数的函数，只需将值作为第二个参数传递即可；例如：
	var result = matlab.exec(' ceil ', 7.4);
	或者
	• 如果需要两个或多个参数，则将所有参数作为JavaScript物件作为第二个参数传递；这可以内联完成，例如：
	var 结果 = matlab.exec('减号', {0: 8, 1 : 4.5});
	注记：参数的顺序由object名称的字母顺序决定
	可以在这里A帮助函数来避免错误：
	// 将可变数量的参数包装到要传递给solver.exec的object中
	函数args() {
	变量 obj = {};
	for (var i = 0; i < arguments.length; i++) {
	obj[i] = 参数[i];
	}
	返回对象；
	}
	var 结果 = matlab.exec('减号', args(8, 4.5));

跟踪()	跟踪（语句）方法允许您在模拟中的任意点打印出跟踪语句。这是在执行期间用附加输出信息补充仿真log的极好方法。
	跟踪() 输出被写入仿真窗口。例如：
	matlab.set('simple_value', 3.14);
	var pi = matlab.get('simple_value');
	跟踪("简单值 = " + pi);

视频

Sparx Systems提供了一个使用 MATLAB 控制台创建 MATLAB Solver 的 YouTube 视频。看：

- MATLAB 控制台

另外两个视频说明了 MATLAB Solver 在执行状态机仿真中对疫苗试验和划的使用。看：

- Matlab 求解器
- 状态机仿真-疫苗试验
- 状态机仿真-彩票划

状态机求解器示例

本主题提供了一些简单的示例，说明在状态机模拟中在何处以及如何应用求解器，其中包括在状态机操作中以及在转移

中使用脚本转移

连接器。本主题中的图像显示 Octave Solver 示例；但是，除非另有说明，否则相同的脚本可用于 MATLAB Solver 中的 MATLAB。

初始化求解器

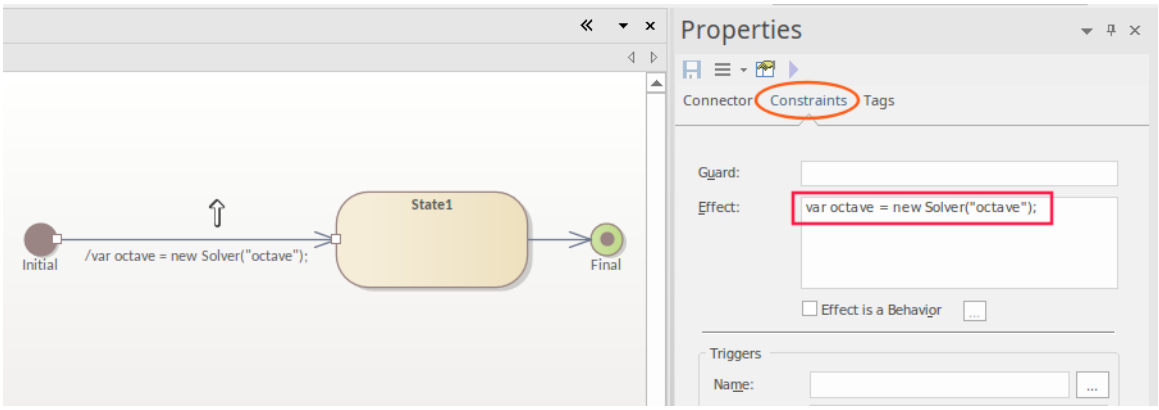
在状态机中初始化求解器的脚本可以放在转移

的影响中转移

或者在某个状态的Entry操作中。通常放在第一个转移

的影响转移

退出初始。



赋值和执行命令

要分配一个值，请使用 octave.set() 或 matlab.set()函数。

对于状态机，该方法可以放在转移

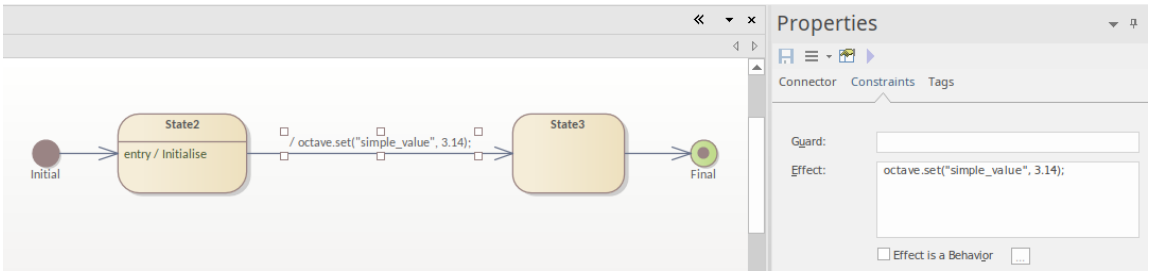
的影响中转移

或在操作的行为（进入/执行/退出）中。

编辑转移

的影响转移

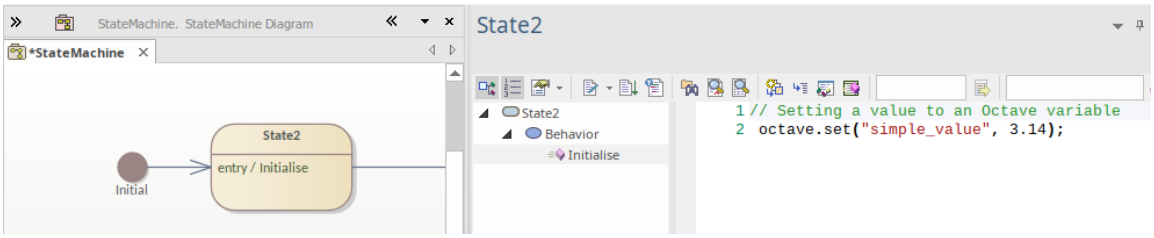
您使用 属性”对话框的 约束”页面。



要将方法放入操作中，请打开操作的行为脚本：

1. 通过右键单击元素并选择 特征>操作"上下文菜单选项，从图表或浏览器窗口在状态中创建新的条目操作。
2. 单击新的条目操作，然后按 Alt+7。

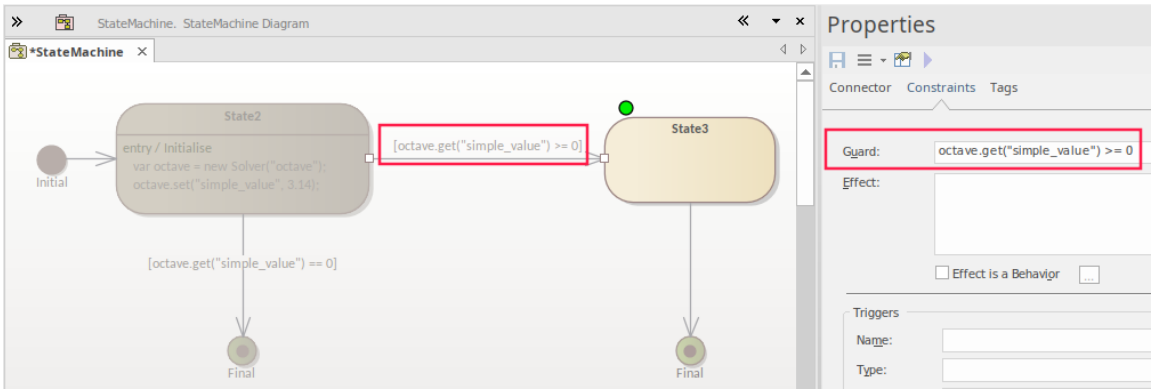
这将在编辑器中打开行为脚本。



当使用影响()函数执行 MATLAB 或 Octave 命令时，这些命令可以放在转移
的影响中转移
或操作的行为脚本。

条件分支

在状态机中使用条件分支时，条件可以放在转移
的守卫条件中转移
并且可以包含调用 MATLAB 或 Octave 函数的脚本。例如：



注记：

- 条件也可以使用选择来设置；条件语句的位置是一样的，都是在现有转移
  的守卫条件中定义的转移
- 该示例显示了在 State3 上设置的断点处的模拟

获得结果

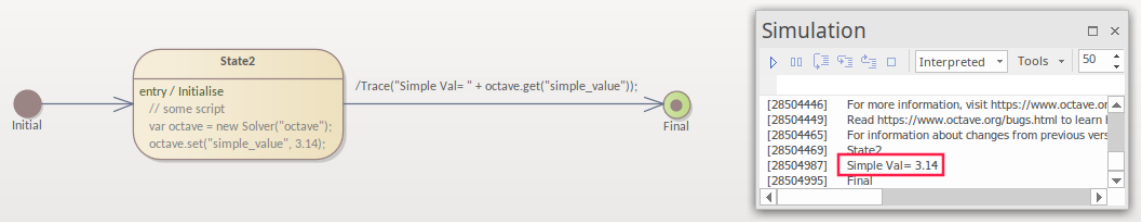
从外部函数调用返回结果时，您可以使用 Solver 的 get()函数在脚本中返回结果。然后将结果从脚本传递给用户
有三个关键选项：

- 跟踪()
- 阴谋
- Win32显示

对于跟踪和 Win32 选项，您必须使用 Solver 的 .get() 命令在JavaScript中返回本地副本。作为使用 .get() 命令的
示例，请参见守卫条件的守卫条件。

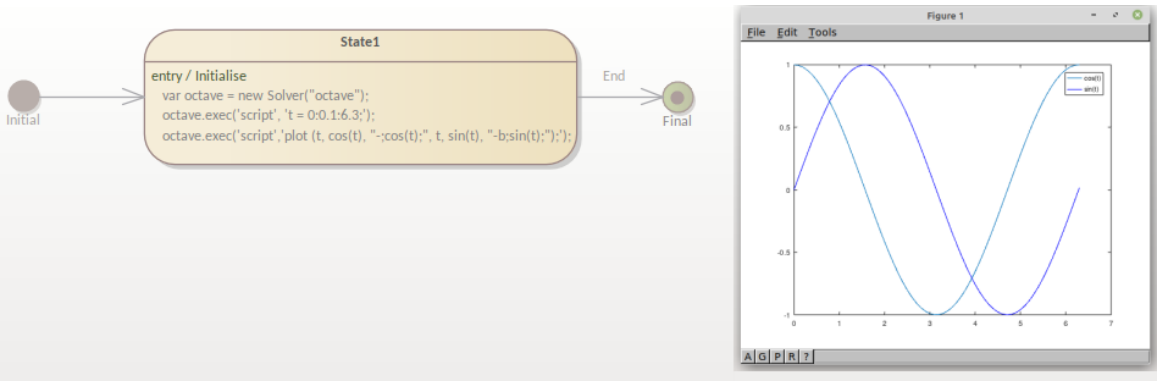
使用跟踪

跟踪() 命令在最初编写和调试模拟时很有用，因为它允许您检查脚本在不同阶段的结果。结果在仿真窗口中输出。



执行绘图

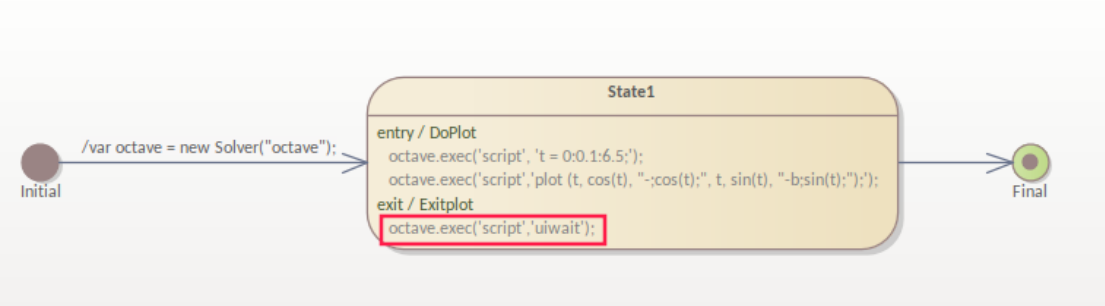
Octave 和 MATLAB 非常强调生成绘图的能力，因为这是输出结果的关键方法。要生成绘图，您可以使用 Solver 的 .exec()函数来调用绘图生成函数。



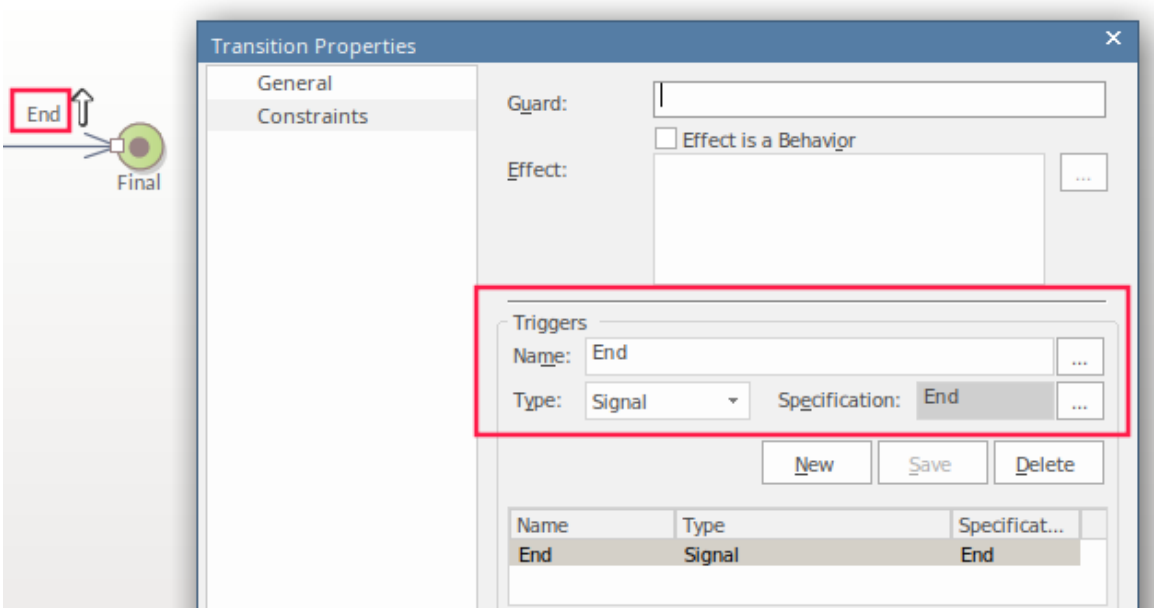
掌握情节

在模拟中执行绘图时，如果模拟没有暂停，则该绘图只能在短时间内查看。因此，对于状态机，有两个选项可以在查看绘图时暂停流程：

1. 在 Octave 或 MATLAB 中使用 uiwait。例如，这可以在退出操作或转移
   中设置转移
   影响。以下是使用 Octave 的示例：
   octave.exec('脚本','uiwait');



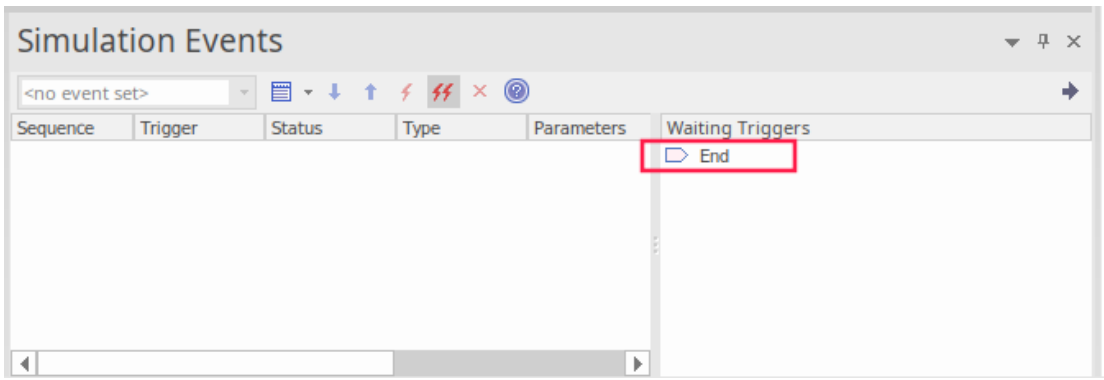
2. 设置触发器/事件序列以暂停转移
   中的模拟转移
   不在生成情节的状态。在示例情况下，这是在最后的转移
   .



超越转移

您可以：

- 点击仿真事件窗口中的触发器展示或
- 从 Win32 屏幕中的一个按钮使用BroadcastSignal()（这将在接下来的使用 *Win32*接口部分中讨论）



使用 Win32接口

使用 Win32 接口时，要执行的主要步骤如下：

- 创建 Win32 对话框
- 设置脚本行以打开对话框
- 从对话框中的字段获取值
- 将该值传递给求解器
- 使用按钮触发剧情

配置Win32接口的步骤如下：

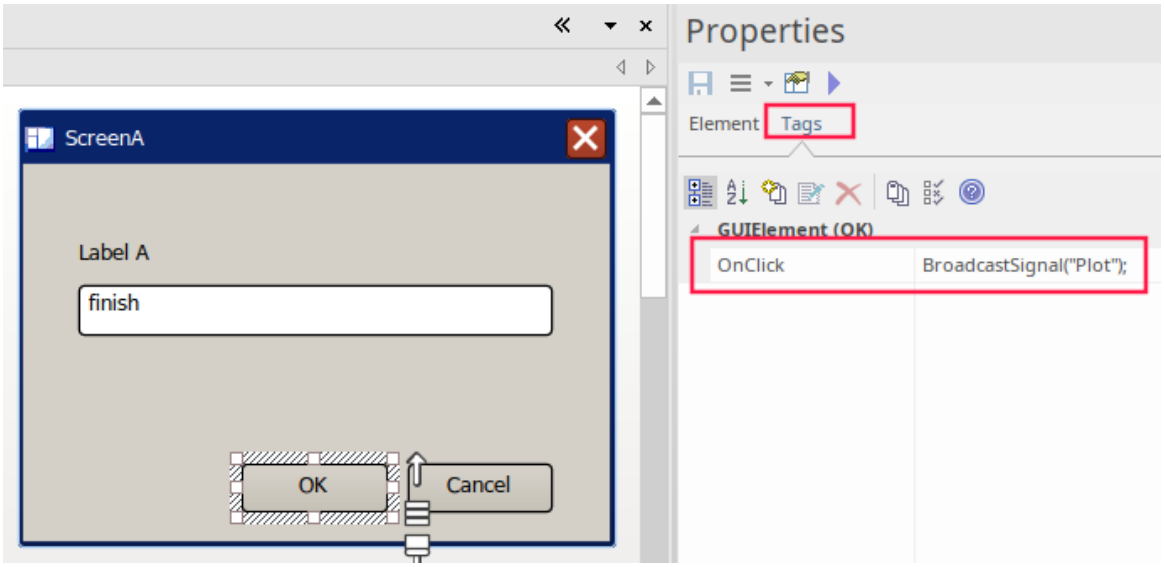
- 使用模型构建器创建 "Starter Win32 Model" - 选择 "UX设计"蓝图集和 "Win32 UI Models"蓝图
- 将 <<Win32屏幕>> 的名称更改为 "ScreenA"
- 将 "Edit控件A"的名称更改为更有 义的名称，例如 "finish"
- 单击 State1 并按 Alt+7 在 Entry Operation 的脚本中添加打开对话框的调用：
  对话框**.ScreenA .**显示= true ；

要获得有效的用户输入值，用户必须单击确定按钮，因此您必须从确定触发器广播一个事件以启动该过程。要接收触发器您必须有信号并设置转移

由该广播信号触发。

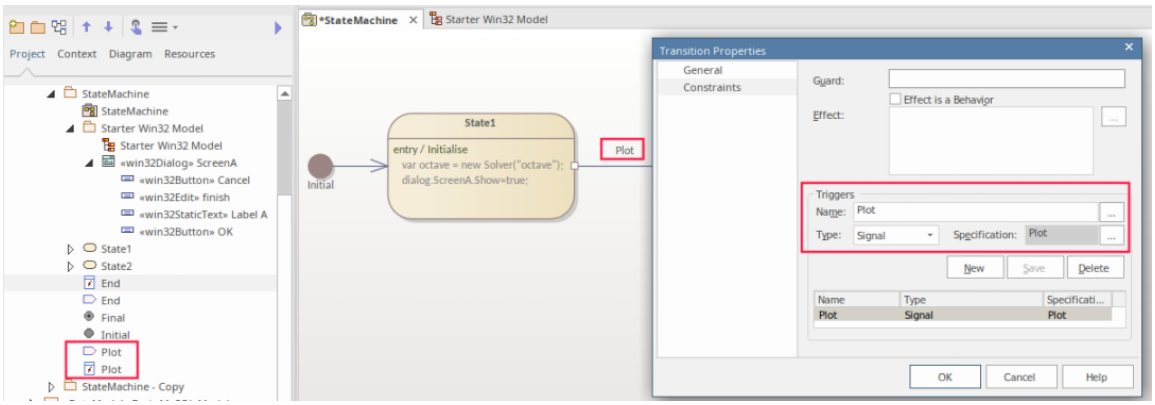
请参阅前面使用触发器保存绘图部分中创建和设置触发器和信号 "End"的步骤。在本例中，我们设置了相同的转移

，但是对于一种称为 "情节"的新触发器。这是使用按钮上的 OnClick标记值发送 BroadcastSignal('Plot')：



转移

退出 State1 设置为由 BroadcastSignal 触发：



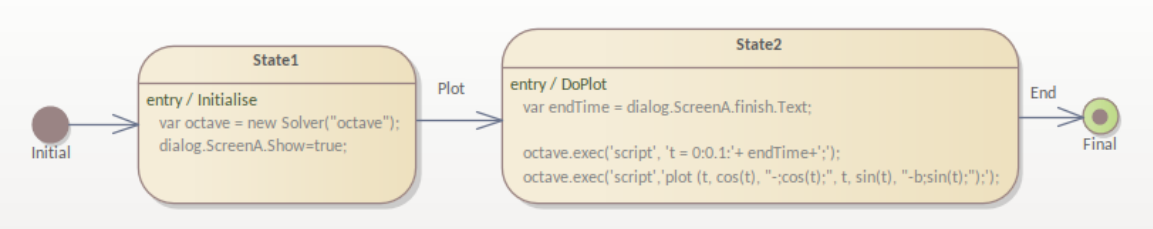
在转移

您必须创建一个触发器，将触发类型设置为信号并创建信号。有关详细细节请参阅*Win32*用户接口仿真帮助主题。

运行绘图的脚本将位于 State2 的 Entry 操作中。在模拟过程中，只有在单击 Win32确定按钮并发送 Plot 触发器后才会转换到 State2。因此我们现在：

- 添加 Entry Operation 并使用 Alt+7 打开行为
- 在这个状态的 Entry Operation 中，我们使用以下方法获取字段的值：
  变量 结束时间 = 对话框.ScreenA.完成.文本;
  这将设置一个变量，其中包含要发送给 Octave 进行绘图的最后一个参数的值
- 在 octave.exec() 语句中，我们放置变量来设置要绘制的秒数：
  octave.exec('脚本' , 't = 0:0. 1 :' + 结束时间+ ';');

这为 Entry操作中的脚本提供了两个状态：

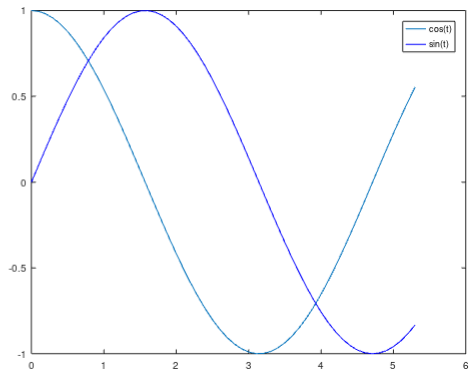


运行仿真

运行仿真：

- 选择仿真功能区，点击 运行仿真>开始”

当您输入一个值并单击确定按钮时，将返回：

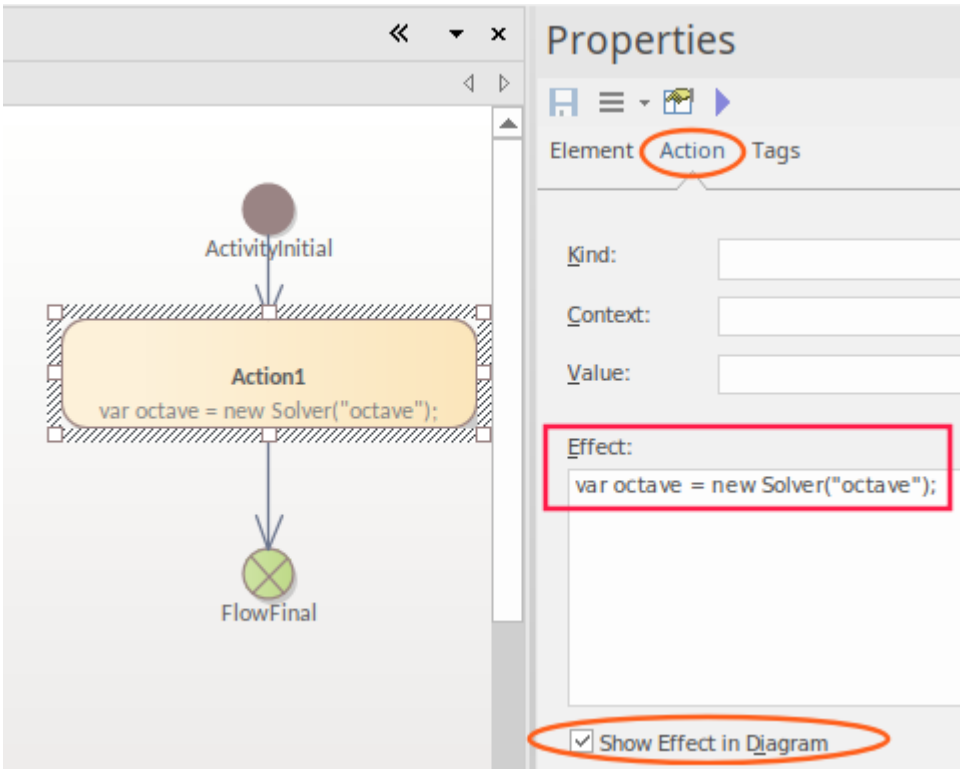


活动图表Solver Example

在本主题中，我们运行通过一些简单的示例来说明在活动图表模拟中在何处以及如何应用求解器。这包括在行动效果以及 ControlFlow Guards 中使用脚本。本主题中的图像显示 Octave Solver 示例；但是，通过将 Octave Solver 替换为 MATLAB Solver，可以将相同的脚本用于 MATLAB。

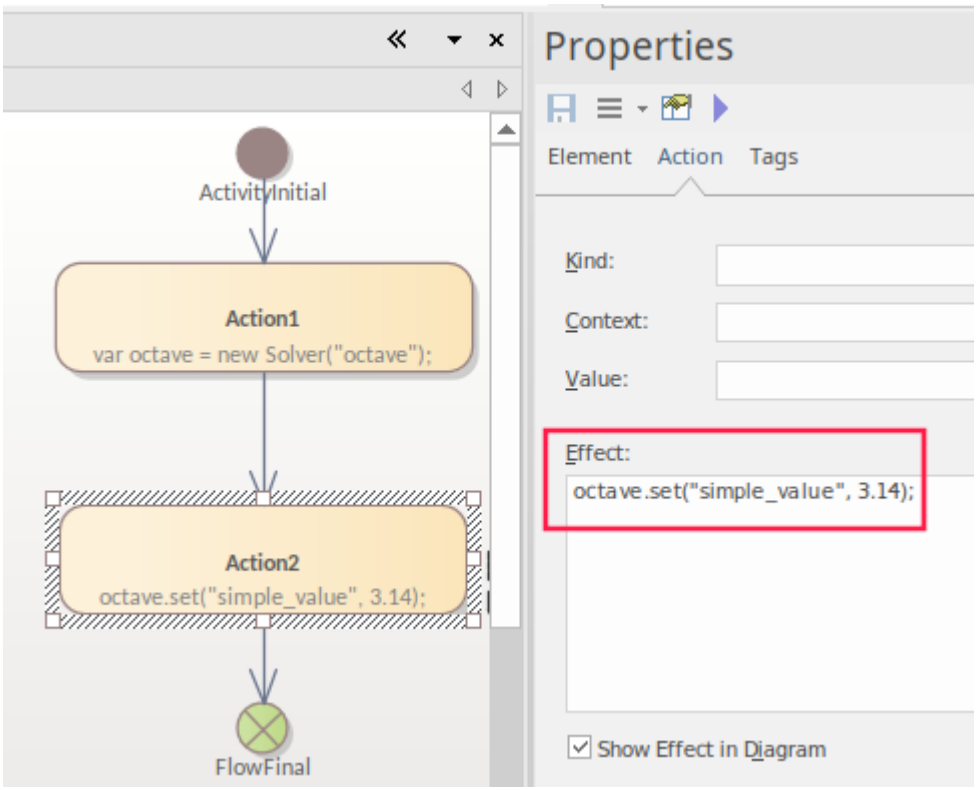
初始化求解器

初始化活动图中的影响的脚本可以放在一个行动的影响中，通常是第一个行动ActivityInitial 的动作的影响。



赋值和执行命令

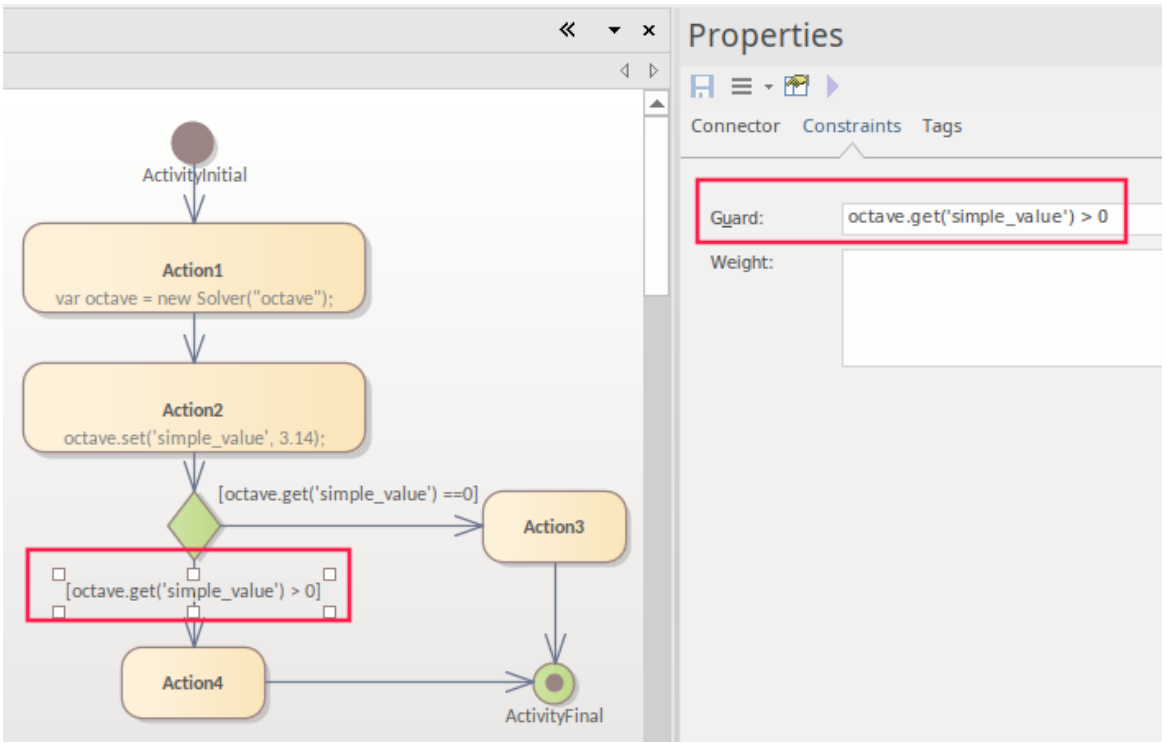
要分配一个值，请使用 octave.set() 或 matlab.set() 函数。对于活动图，函数可以放在行动的影响中。



当使用行动()函数执行 MATLAB 或 Octave 命令时，这些命令可以再次放置在行动的影响中。

条件分支

对于状态机中的条件分支，条件可以放在守卫条件的守卫条件中，并包含调用任何 MATLAB 或 Octave函数的脚本。例如：



获得结果

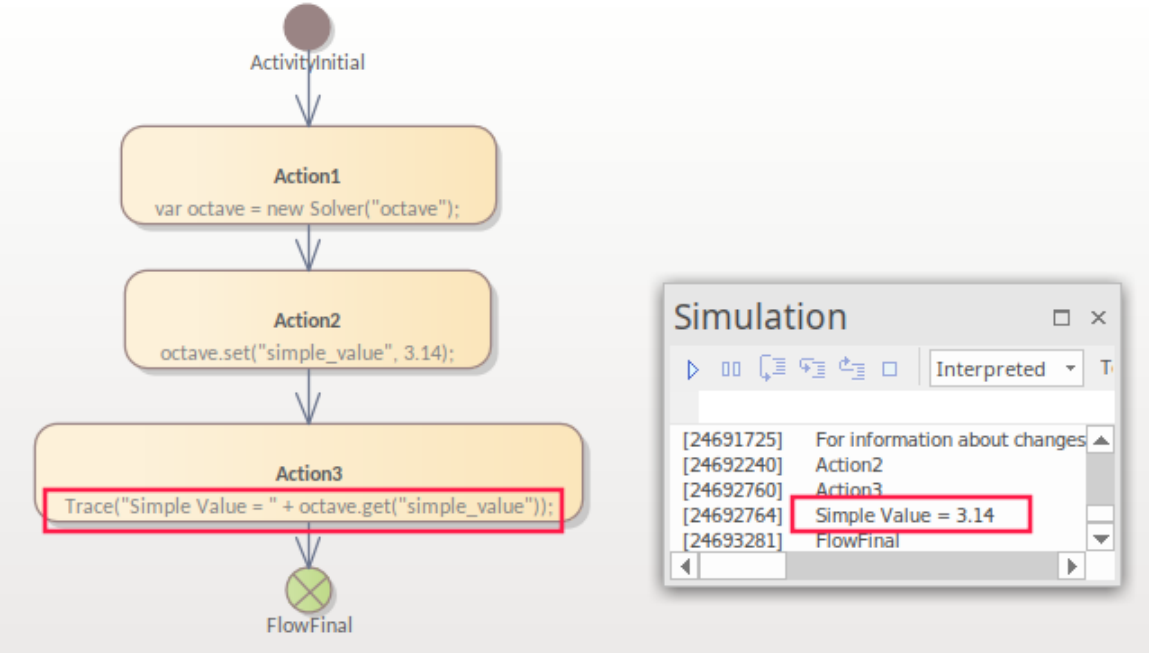
要从外部函数调用返回结果，请使用 Solver 的 .get()函数。对于如何将结果从脚本传递给用户，有三个关键选项：

- 跟踪()
- 阴谋
- Win32显示

使用跟踪和 Win32 选项，您必须使用 Solver .get()函数在JavaScript中返回本地副本。上图包含一个在守卫条件中使用 .get()函数的示例。

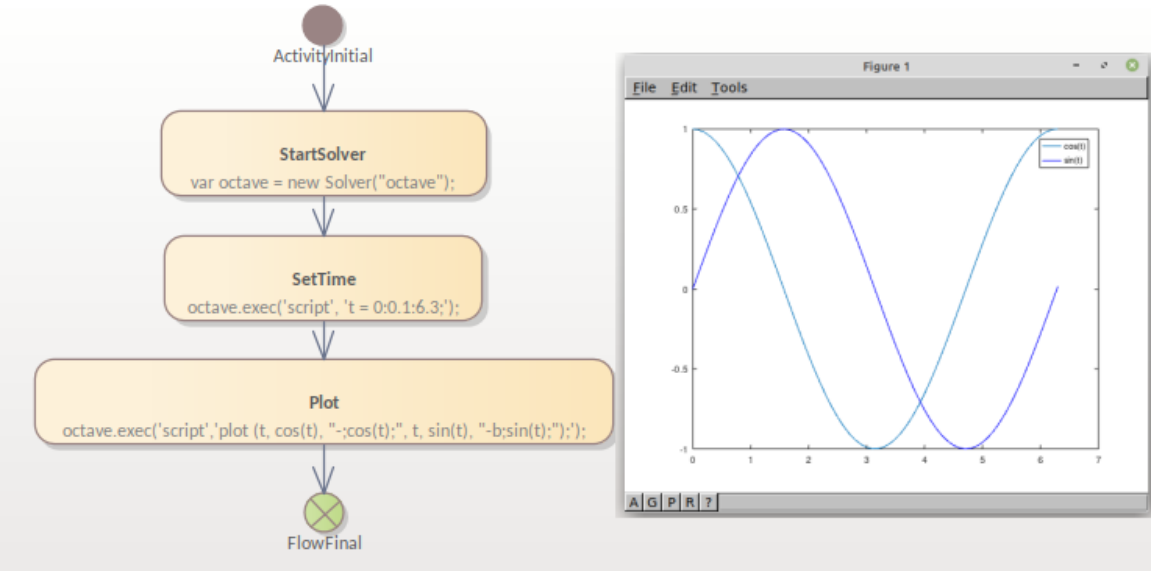
使用跟踪

跟踪() 命令在最初编写和调试模拟时很有用，因为它允许您检查脚本在不同阶段的结果。结果在仿真窗口中输出。



执行绘图

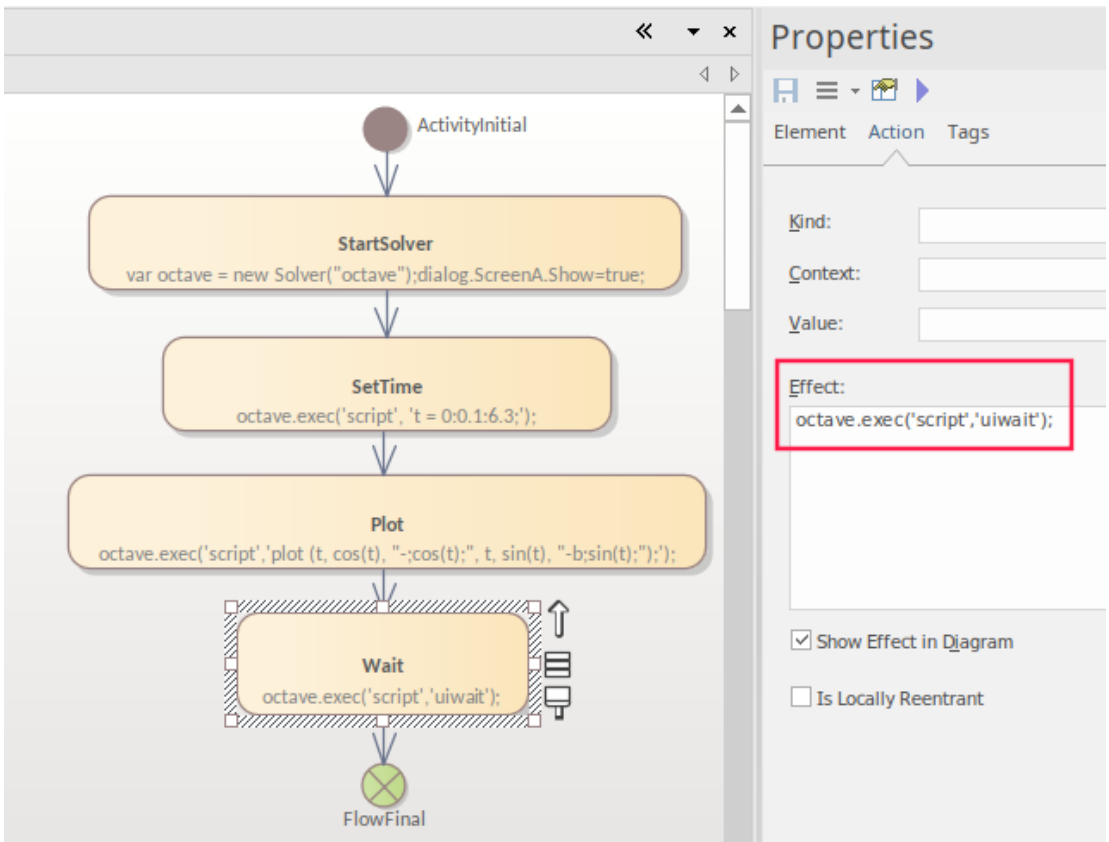
Octave 和 MATLAB 非常强调生成绘图，因为这是输出结果的关键方法。要生成绘图，您可以使用 Solver 的 .exec()函数来调用绘图生成。



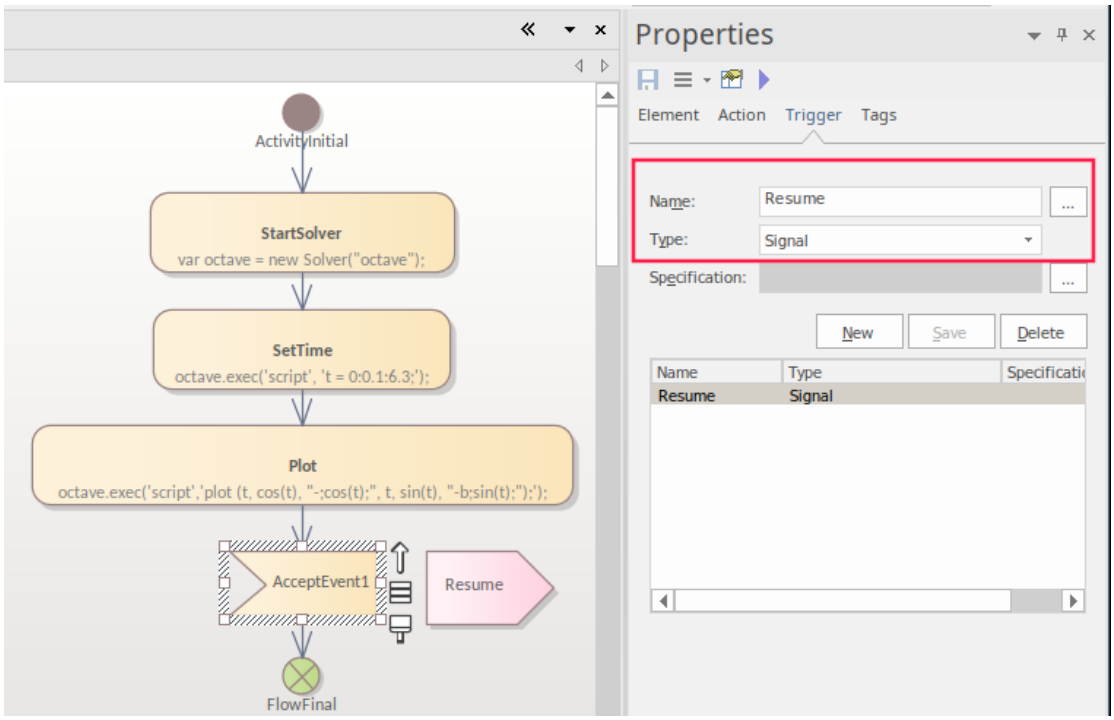
掌握情节

在模拟中执行绘图时，如果模拟没有暂停，则绘图只会短暂可见。对于活动图表，有两个选项可以在查看情节时暂停流程：

1. 使用 Octave 或 MATLAB 中的影响函数，将其设置在行动的影响中。以下是使用 Octave 的示例：
   octave.exec('脚本','uiwait');

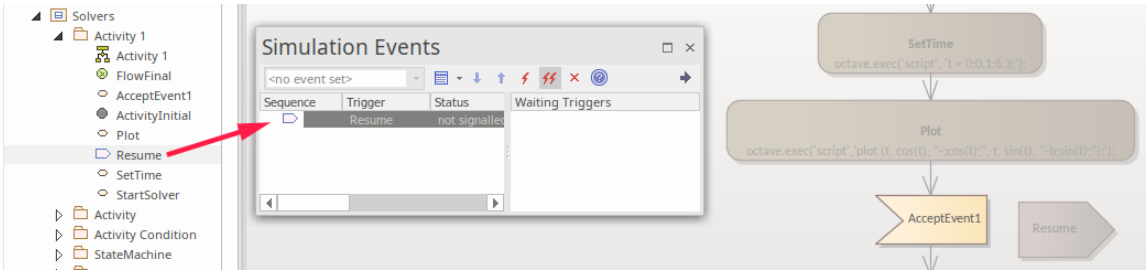


2. 将行动类型的动作设置为在生成绘图后暂停，这需要在触发器上设置对模拟活动的引用。这可以使用显示选择触发器”对话框的[...]按钮来创建和引用。点击加新按钮来创建触发器。



要超越 AcceptEvent，您：

- 将触发器（简历）从浏览器拖到仿真事件窗口
- 双击该过渡



使用 Win32接口

使用 Win32 接口时，大致要执行的步骤如下：

1. 创建一个 Win32 对话框。
2. 设置脚本行来打开它。
3. 从对话框中的字段获取值。
4. 将该值传递给求解器。
5. 使用按钮来触发情节。

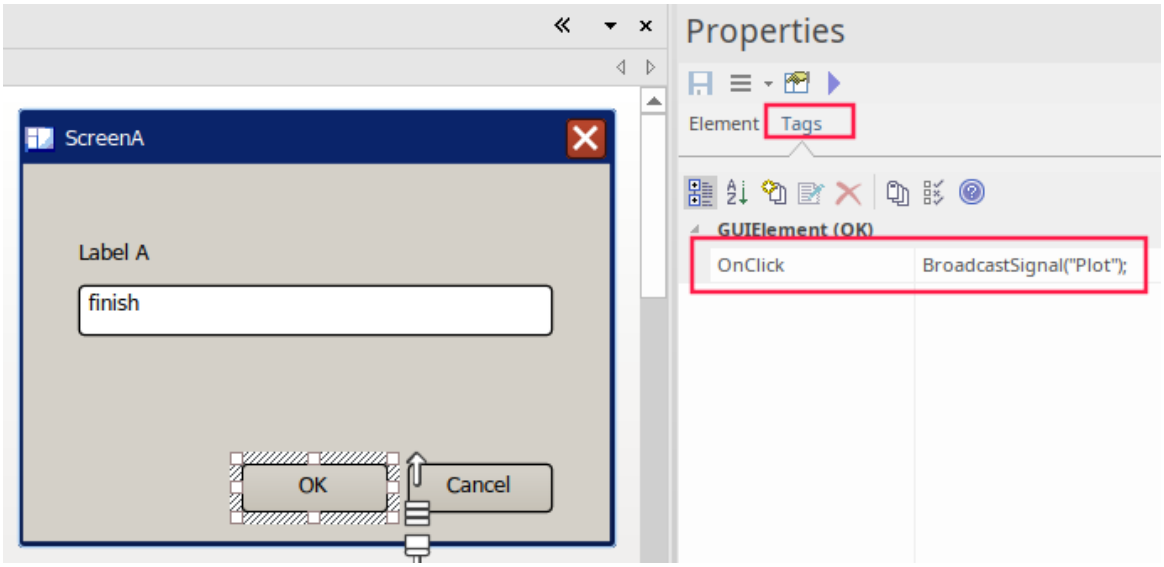
以下是配置 Win32 接口的步骤：

1. 使用 模型生成器"对话框（Ctrl+Shift+M）创建 Starter Win32 模型"。
2. 将 <<Win32屏幕>> 的名称更改为 ScreenA"。
3. 将 Edit控件A"的名称更改为更有 义的名称，例如 finish"。
4. 在 State1 上按 Alt+7 添加打开对话框的调用，在 Entry Operation 的脚本中：
   对话框.ScreenA.显示=true;

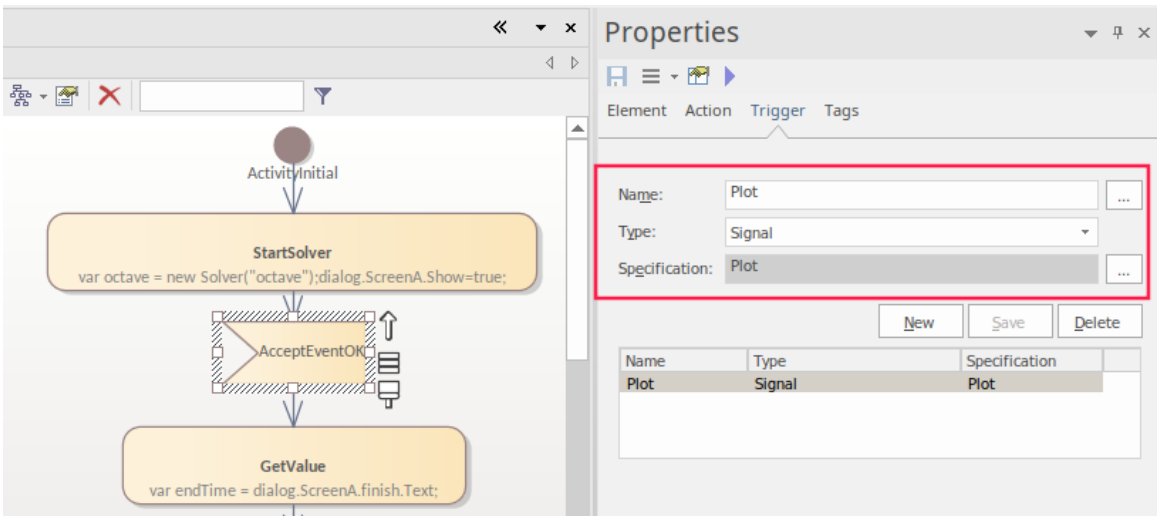
要获得有效的用户输入值，用户必须单击确定按钮，因此您需要从确定触发器广播一个事件以启动该过程。要接收触发器您设置一个信号和一个转移

由该广播触发。

请参阅创建和设置触发器和信号 "End"的步骤，如前面的 "保持绘图"中所示。在本例中，我们设置相同的内容，但针对的是名为 "Plot"的新触发器。此图使用按钮上的 OnClick标记值发送 BroadcastSignal('Plot')。



在退出 StartSolver 的 ControlFlow 上，现在有一个行动 "AcceptEvent"。它设置为按如下所示触发：



在属性窗口 "触发器"选项卡上，您创建一个触发器，将 "触发类型"设置为 "信号"并为其创建信号。有关详细细节请参阅*Win32*用户接口仿真帮助主题。

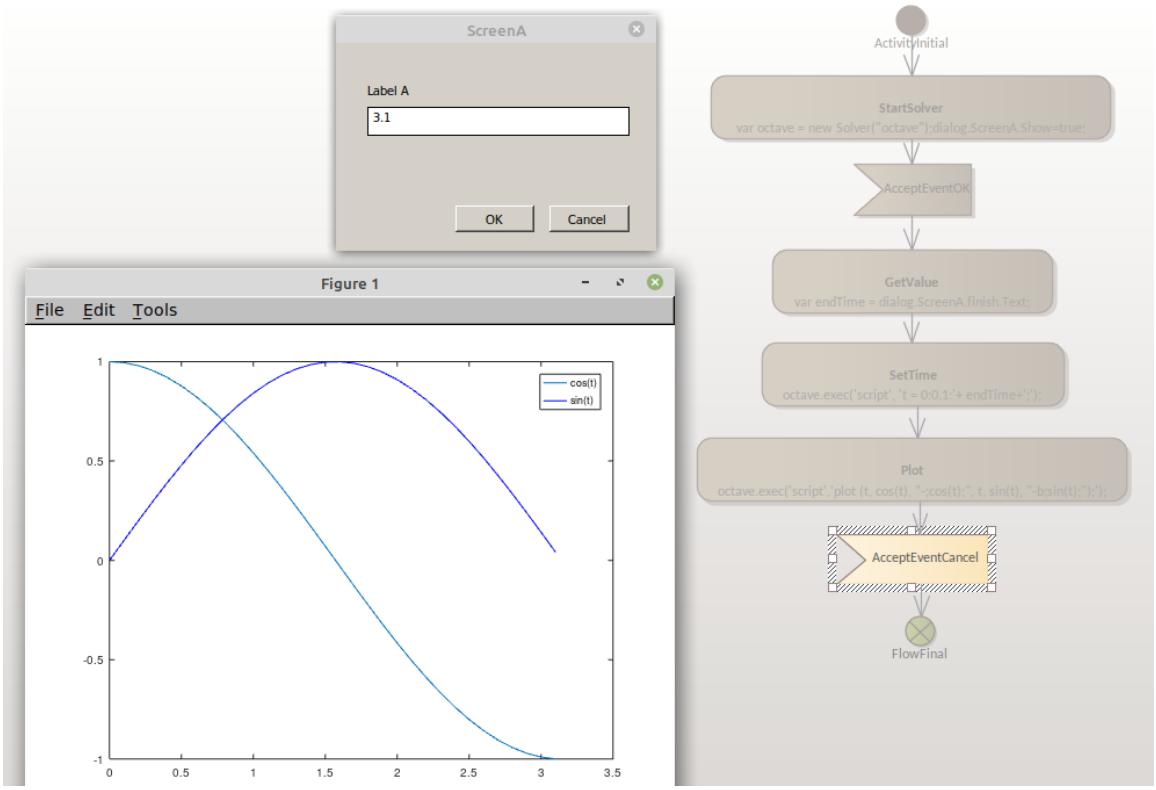
取消按钮的配置方式与确定按钮（带有广播信号）相同，并且在 AcceptEventCancel 上设置触发器参考以使用 "取消"触发器/信号。

运行仿真

运行仿真：

- 选择仿真丝带
- 点击 "运行仿真>开始"

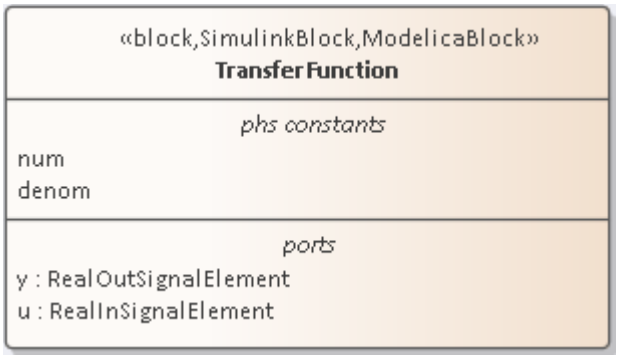
当您输入一个值并单击确定按钮时，将返回：


ScreenA
Label A
3.1
OK
Cancel
Figure 1
File Edit Tools
cos(t)
sin(t)
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
2.5
3
3.5
Activity Initial
StartSolver
var octave = new Solver("octave");dialog.ScreenA.Show=true;
AcceptEventOK
GetValue
var endTime = dialog.ScreenA.finish.Text;
SetTime
octave.exec('script', 't = 0:0.1:'+ endTime+';');
Plot
octave.exec('script','plot (t, cos(t), "-;cos(t);", t, sin(t), "-b;sin(t);");');
AcceptEventCancel
FlowFinal

SysPhS仿真

交互扩展用于定义物理和信号仿真仿真规范（Ph），是物件管理组 (OMG) 规范的扩展，旨在为一致的模型提供通用建模平台。这些模型可以转换为两个关键仿真平台之一，Modelica 和 MATLAB 的 Simulink/Simscape。

OMG SysPhS 原型帮助您在模型本身而不是在模拟配置规范中定义模型模拟的特征。它们在浏览器窗口和属性窗口中以及在带有属性类型和初始值的特定元素隔间的图表中提供了更大的object或属性类型的可见性。



该标准在Enterprise Architect中由 OMG SysPhS配置文件表示，以及：

- SysPhS 用于信号流和物理交互的元素库（在 SysPhS 标准下执行模拟所必需的）
- 专门A工具箱页面
- 广泛A组件元素模式，可从中生成常见的仿真元素，例如电子、逻辑和流体组件；模式引用 OpenModelica 或 Simulink 标准库中的库模块
- 特征用于使用 Modelica 或 MATLAB 的 Simulink、Simscape 和状态流进行仿真绘图。

SysPhS特征

特征	描述
引用 SysPhS 库	使用 SysPhS 的关键资源是 SysPhS仿真库，其中包括您必须在模型中引用的可重用资源。
工具箱	图表工具箱的图表页面包含 OpenModelica 和 MATLAB Simulink 的基本 SysML 元素。
SysPhS模式	SysPhS模式提供引用等效 MATLAB 和 Modelica 组件的预定义 SysPhS 模块。在使用 SysPhS 模型时，这些简单的块可以用作启动器。
SysPhs 组件	SysPhS 组件使您能够设置对 Modelica 和 Simulink 组件的引用。
仿真	您可以使用附加信息定义 IBD 或参数模型以驱动仿真，然后使用仿真配置在 Modelica、Simulink 或 Simscape 中生成模型，以生成结果图。
SysPhS 示例	有几个使用 SysPhS 设置模拟的例子。
为 SysPhys 更新 SysMLSim	您可以更新旧的模拟配置（ Enterprise Architect 15.2 之前），以反映 SysPhS 标准的使用。

选项

变量和常量的额外选项，例如 isContinuous 和 isConserved，自动设置为标记值，再次避免了在配置规范中定义它们的需要。这些选项在块本身和停靠的属性窗口中也可见。

视频

- 用于电路仿真的SysPhS模式
- 使用仿真和 Modelica 仿真数字电子设备

引用 SysPhS仿真库

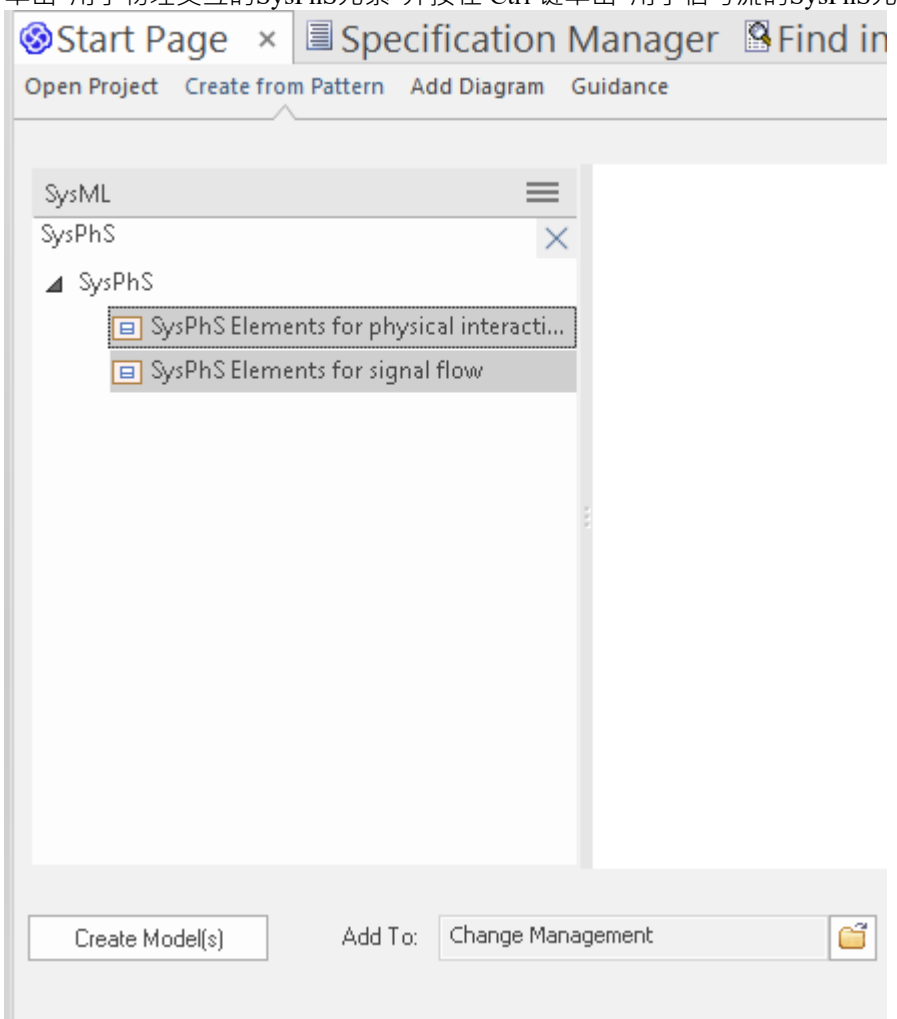
SysPhS 规范提供了独立于平台的部件库的定义，该部件库由两个组件组组成：信号流和物理交互。由于此组件库是用于建模的底层资源，因此您正在处理的存储库中必须包含此库的副本，以供 SysPhS 模型参考。

为了根据 SysPhS 标准执行模拟，您必须从模型生成器下载两个 SysPhS 库：

- SysPhS元素用于物理交互和
- SysPhS信号流元素
- 这些库可用于存储库中创建的多个 SysPhS 模型。建议将它们下载到一个唯一的包中，然后在任何 SysPhS模型中设置对该包的单个引用。

下载库：

1. 创建一个包来保存这些库。
2. 在浏览器里选择包。
3. 将蓝图（ ，工作区右上角）设置为 系统工程> SysML"。
4. 按 Ctrl+Shift+M 打开模型生成器对话框。
5. 在过滤器栏中的蓝图名称下方，键入 "SysPhS"。
6. 单击 用于物理交互的SysPhS元素"并按住 Ctrl 键单击 用于信号流的SysPhS元素"。



7. 单击创建模型按钮。

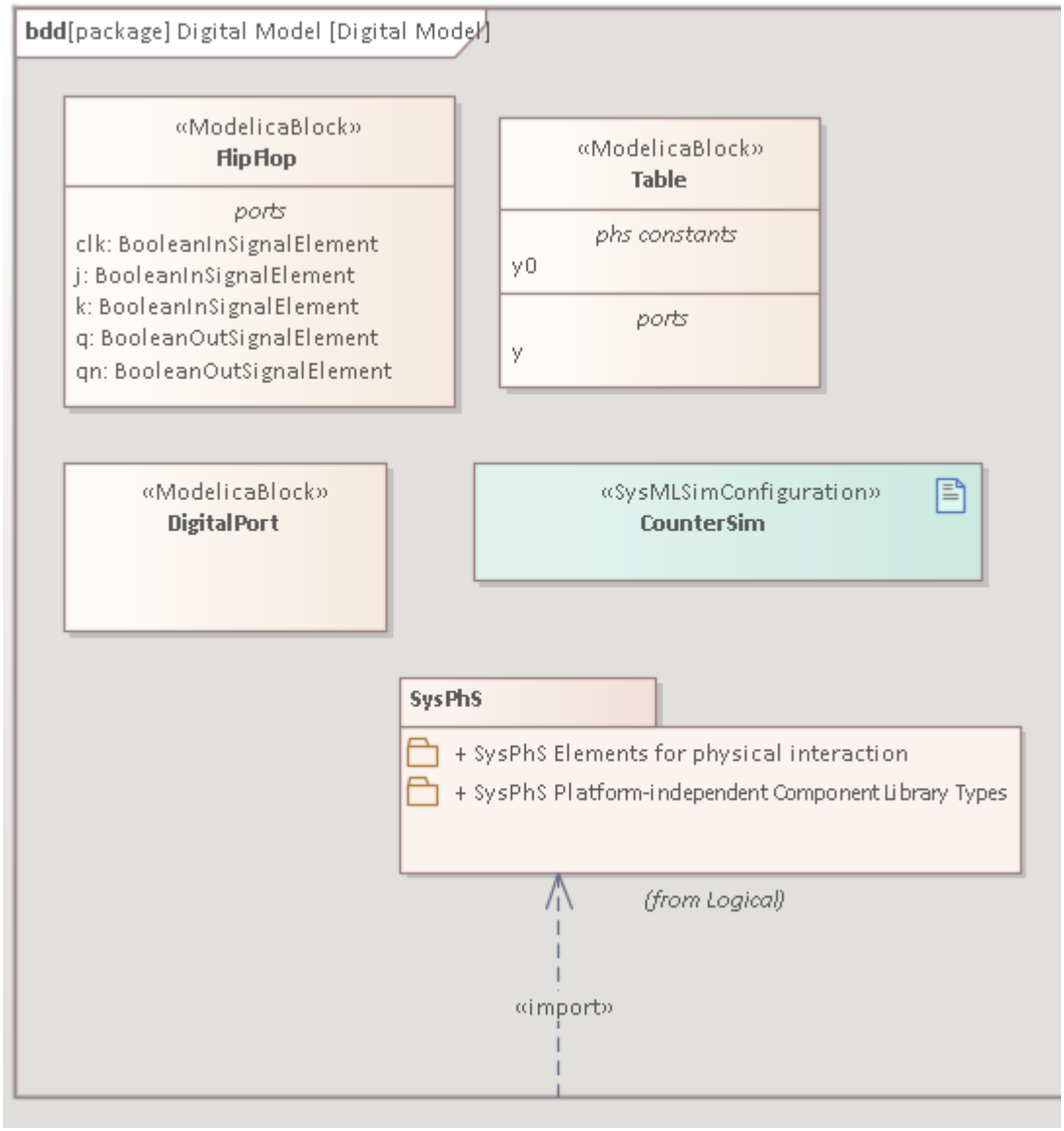
这些库被加载到所选的包中。

对于使用 SysPhS 的每个 SysML 模拟，块定义图表框架必须具有对库包引用。要设置它，请使用导入连接器将

库包链接到 SysML 图表框架：

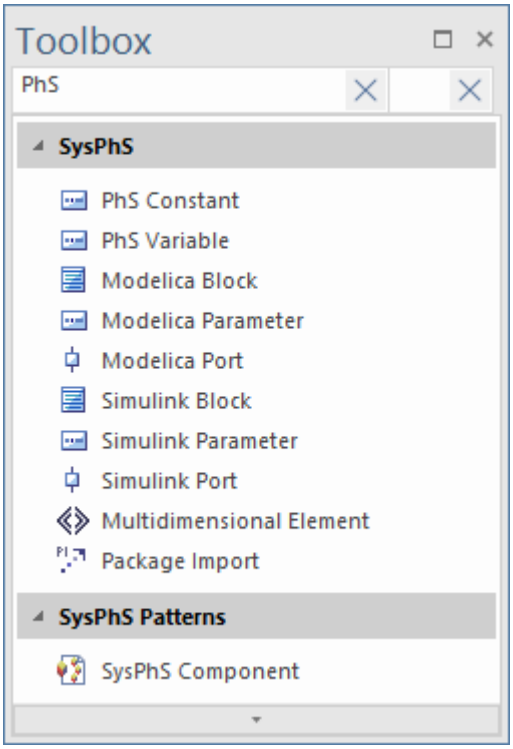
1. 为模拟创建一个块定义图。
   这将自动在元素周围产生一个边界框。
2. 将 SysPhS 库包拖到图表上。
   提示A
3. 选择 包元素"选项。
   库包元素被添加到边界框架内的图中。
4. 右键单击边界框以显示上下文菜单，并确保 可选"选项被勾选。
5. 将工具箱更改为 SysPhS"页面：
   单击 包导入"关系图标，并在边界框和库包之间拖动光标以创建导入连接器。
6. 保存图表（按 Ctrl+S）。

这是使用 <<import>> 连接器的 SysPhS包引用的示例：



使用工具箱

SysPhS 配置文件提供了一个专用的工具箱页面，其中包含用于 OpenModelica 和 MATLAB Simulink 的基本 SysML 元素，以及在 "SysPhS模式"页面上的 "SysPhS部件"图标。



系统元素

元素Icon	描述
PhS常数	运行常数定义了在仿真运行期间保持不变的值。 对于 Modelica，这些对应于参数变量。 对于 Simscape，这些对应于（常量）参数。
PhS 变量	运行变量定义在仿真运行期间可变的值。 对于 Modelica，PhSVariables 具有： • isContinuous=true 对应Modelica连续组件 • isContinuous=false 对应离散元件 对于 Simulink，PhSVariables 对应于 Simscape变量。
模型块	Modelica块元素类型用于在 Modelica 库中定义相应的组件。要将 Modelica块定义为对应于 Modelica 库中的组件，需要将 Modelica SysPhS名称属性的值设置为 Modelica 中的完全限定组件名称（类路径）。
Modelica参数	Modelica参数element-type 用于定义Modelica 库组件A参数。这些在属性窗口>元素选项卡 > «ModelicaParameter»（来自 SysPhS）中定义，它有两个参数： • 名称：对应Modelica参数名称（Parameters Name）

	• 价值： 该参数的类型及其初始值可以在属性>属性选项卡的字段中定义： • 类型 • 最初的
模特端口	Modelica端口对应于端口库中定义A端口。 对应的 Modelica端口名称定义在：属性>> « *ModelicaPort»*（来自 *SysPhs*）元素名称
Simulink块	Simulink块元素类型用于在 Simulink 库中定义相应的组件。要定义与 Simulink 库中的组件对应的 Simulink块，需要将 Simulink属性名称的属性值设置为来自 Simulink 的完全限定组件名称（在 Simulink模型资源管理器中作为 ″内容″找到）。
Simulink参数	Simulink参数element-type 用于定义 Simulink Library 组件A参数。这些在属性窗口 >元素选项卡 > «SimulinkParameter»（来自属性）中定义，它有两个参数： • 名称：对应Simulink库中的参数名称 • 价值： 该参数的类型及其初始值可以在属性>属性选项卡的字段中定义： • 类型 • 最初的
Simulink端口	Simulink端口对应于端口库中定义A端口。 对应的 Simulink端口名称定义在：属性>元素>*«SimulinkPort»*（来自 *SysPhs*）名称>。
多维元素	在系统建模中，虽然可以使用多重性来指定向量，但对于多维数组，我们需要多重多重性。 MultidimensionalElement 原型提供了这一点，有效地支持定义维度的多重性数组。这通过拖动多维元素图标应用于块。
导入包	需要导入包连接器来设置从主块到导入仿真库的引用。有关更多详细信息，请参阅参考 *SysPhS*仿真库帮助帮助。

SysPhS模式

在 OMG SysPhS 规范中有一个 Modelica 和 Simulink 通用的 SysPhS 组件列表。 Enterprise Architect将这一系列组件作为一组 SysPhS模式提供。这些核心组件是创建新模型时有用的起点。有关使用这些的更多详细信息，请参阅使用 *SysPhS*模式帮助帮助。

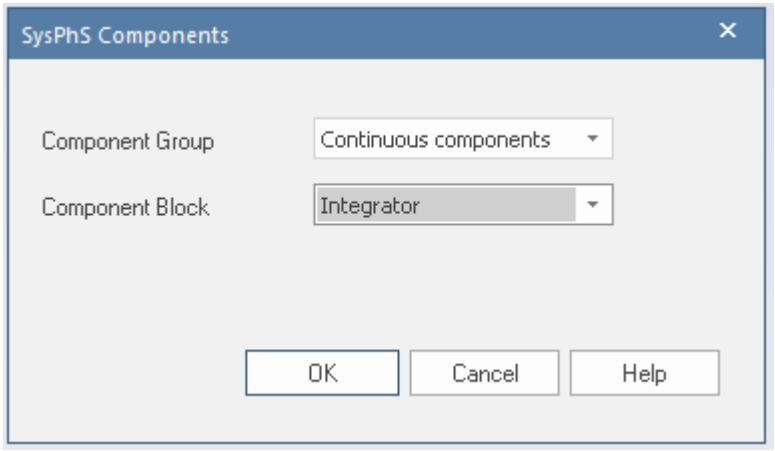
使用 SysPhS模式

OMG SysPhS 规范列出了一系列 SysPhS 组件作为平台独立部件库。 Enterprise Architect *SysPhS*模式提供的是 Modelica 和 Simulink 共有的这一系列组件。选择一个项目时，会创建一个为Modelica和Simulink键入的块。然后，当运行设置为 Modelica 或 Simulink 的仿真时，将在相应的工具中创建此项目类型。

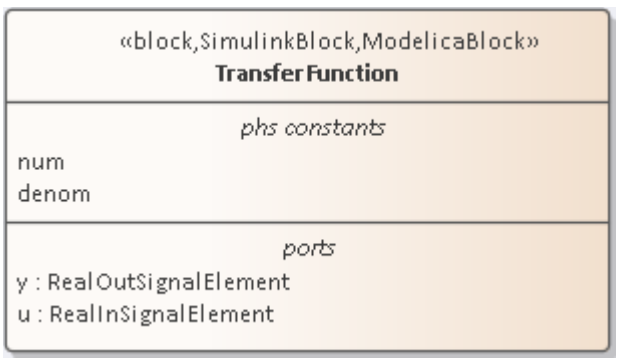
这些组件的一般分组是

- 连续组件
- 离散组件
- 非线性组件
- 数学成分
- 源和汇
- 路由组件
- 逻辑组件
- 电气元件

当您将 "SysPhS部件"图标拖到图表上时，将显示 "SysPhS 组件"对话框。



在此对话框中，您首先选择要创建的元素类别，然后选择要创建的元素类型，两者都通过从 "部件组"或 "部件块"字段的下拉列表中进行选择。每个字段中的下拉列表都是从 OpenModelica 和 Simulink 标准库中填充的。例如：



请注意，元素具有 Simulink 和 Modelica 的构造型，因此可以在这两种工具中使用。对于每个元素，元素类型的正确属性和端口自动包含在元素中，作为元素隔间中的项目。如果您更喜欢图表上的实际存在，您可以将实际的结构元素从浏览器窗口拖到元素上。

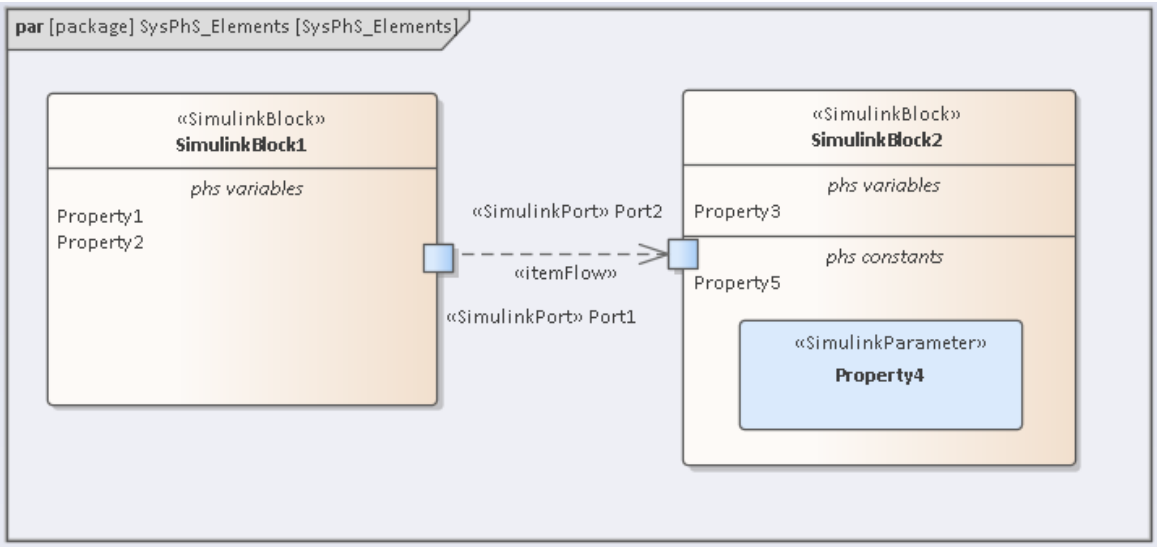
有关所有受支持组件的列表和更多详细信息，请参阅部分： *OMG SysPhS 1 .0 PDF*中的*11.3.2* 实值组件。

使用 SysPhS 组件

如果您正在使用 Modelica 或 Simulink 中的示例，并且您想在Enterprise Architect中引用该模型中已经存在的组件，您可以从模型中拖动适当的块、参数或端口元素工具箱到创建引用元素的图表。

显示属性和部件

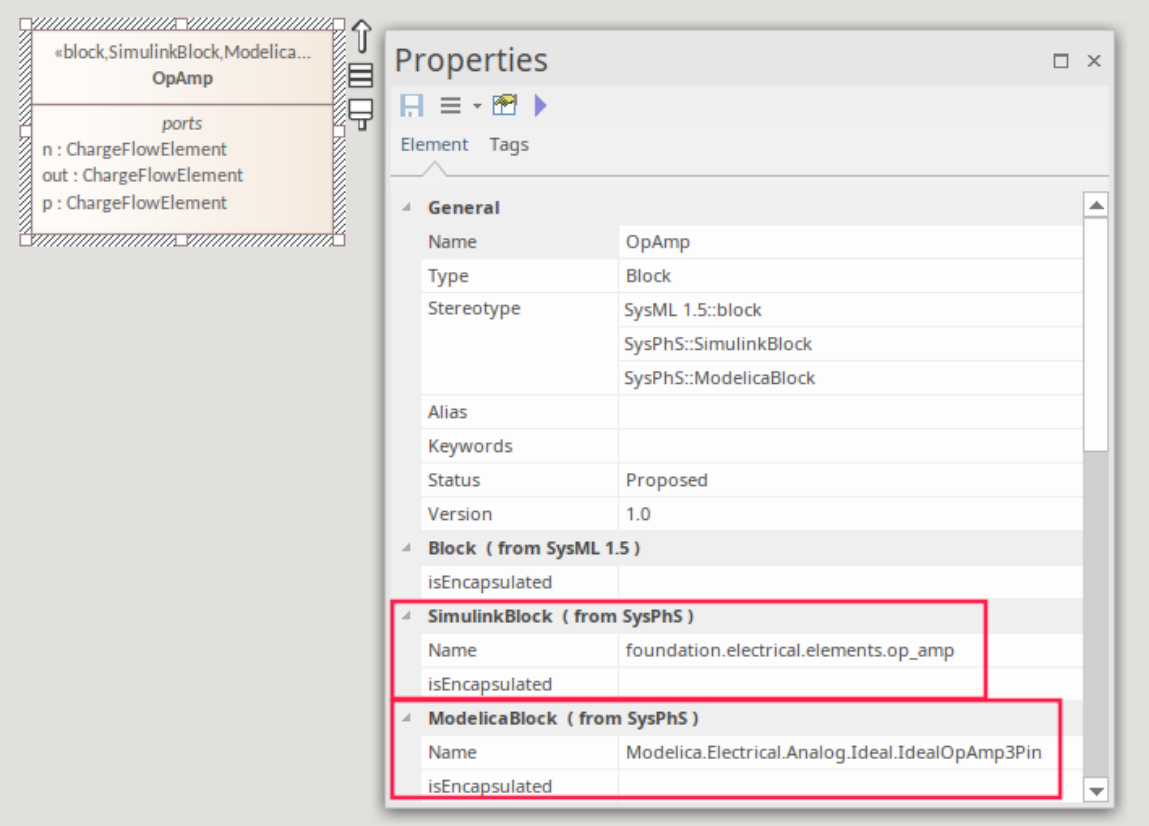
在注记图中使用属性和部件时需要注意A一点是，它们的默认显示为部件object 。它们可以留在该渲染中或设置为在隔间中显示为文本。以下是两者的示例：



属性1和 5 已作为元素被拖到图表上，然后从图表中删除（但不是从浏览器窗口中删除）。删除时，它们在图表上被各自的块中的文本条目替换。属性4 作为部件元素留在图表上；如果从图中删除，它也将成为隔间中的文本条目。

块元素型

将 Modelica 或 Simulink 模块类型拖到图表上会创建一个非特定的 SysPhS块。要将此块设置为特定的 Modelica 或 Simulink块，请在属性窗口中的**SimulinkBlock (SysPhS)**或**ModelicaBlock (SysPhS)**段下设置 名称”字段。

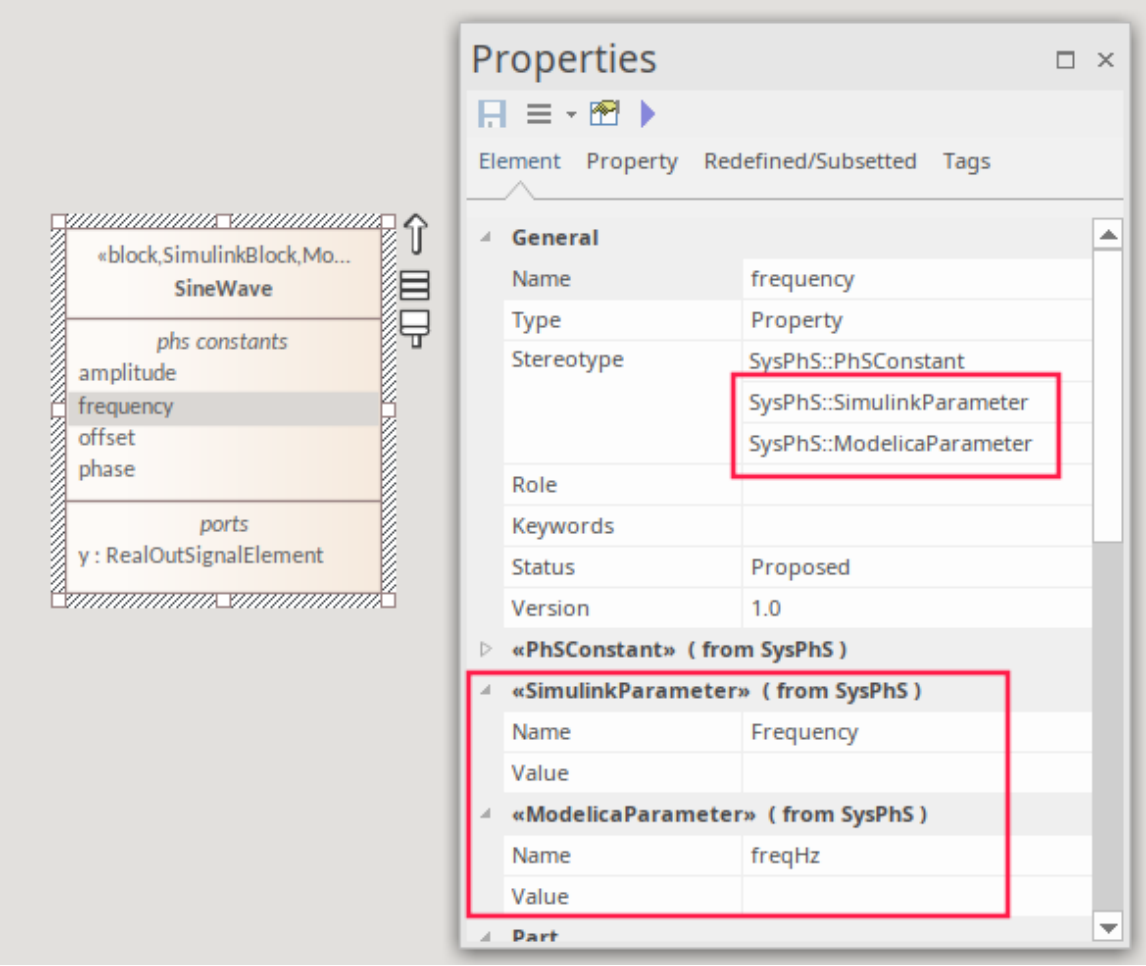


有关在这些外部工具中引用 Simulink 和 Modelica属性的更多详细信息，请参阅创建 *Modelica* 特定模块和创建 *Simulink* 特定模块帮助主题。

注记：如图所示，如果要在两个外部工具中模拟模型，可以同时应用两个 SysPhS构造型。请参阅将模块设置为 *Modelica* 和 *Simulink*帮助主题。

参数Element-type

参数元素类型使用 SimulinkParameter 或 ModelicaParameter 构造型创建属性元素。如果从图中删除元素，它们会列在父块元素的*phs* 常量隔间中。这是一个为 Modelica 和 Simulink 设置的 SysPhS参数示例，显示了每个参数的构造型以及在 名称"字段中对各自产品参数名称的引用。

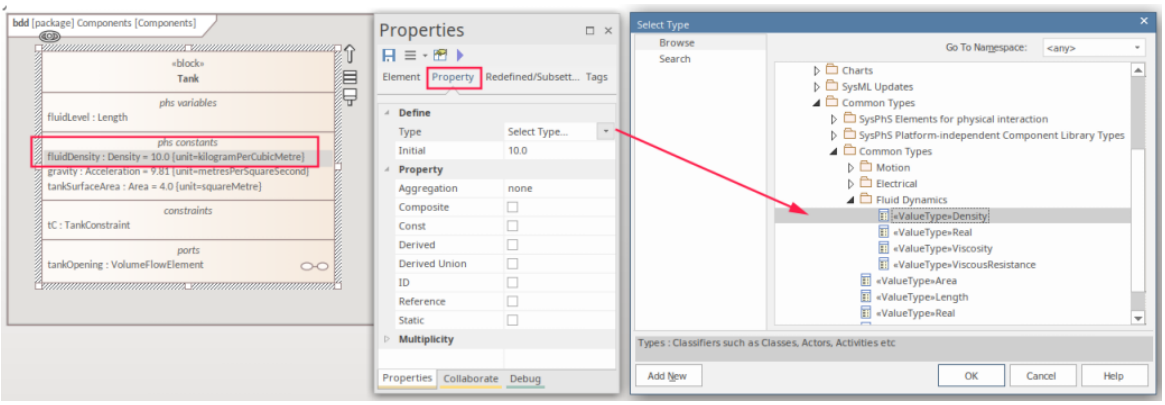


PhsConstant 和 PhsVariable

要定义一个元素的常量和变量属性，您可以将 块常量"和 PhS 变量"图标拖到图中的某个元素上。同样，如果您从图中删除元素，它们将列在*phs* 常量或*phs* 变量隔间中。

在模型中设置常数值时，可以在块中设置值，也可以在从块派生的部件中设置值。例如，在重力作为绝对常数的情况下，这最好在块中定义。部件或块的值可以在仿真属性中进一步更改。

下图显示了水的流体密度集，它可以在部件或模拟中被覆盖以定义另一种流体（例如油）的密度。如果一个块以不同的值重复使用 - 例如一个电阻为 3.3 kohms，另一个电阻为 5.6 kohms，则初始值最好在 IBD 中的特定部分中定义，这些部分源自没有初始值的块.



属性>属性"选项卡有两个字段：

- 类型
- 最初的

该类型可以设置为标准类型，或者在本例中设置为模型中引用的 SysML *ValueType* 。

注记如果 PhsConstant 或 PhsVariable 设置了初始值，它会显示在元素的初始值隔间中的图表上。

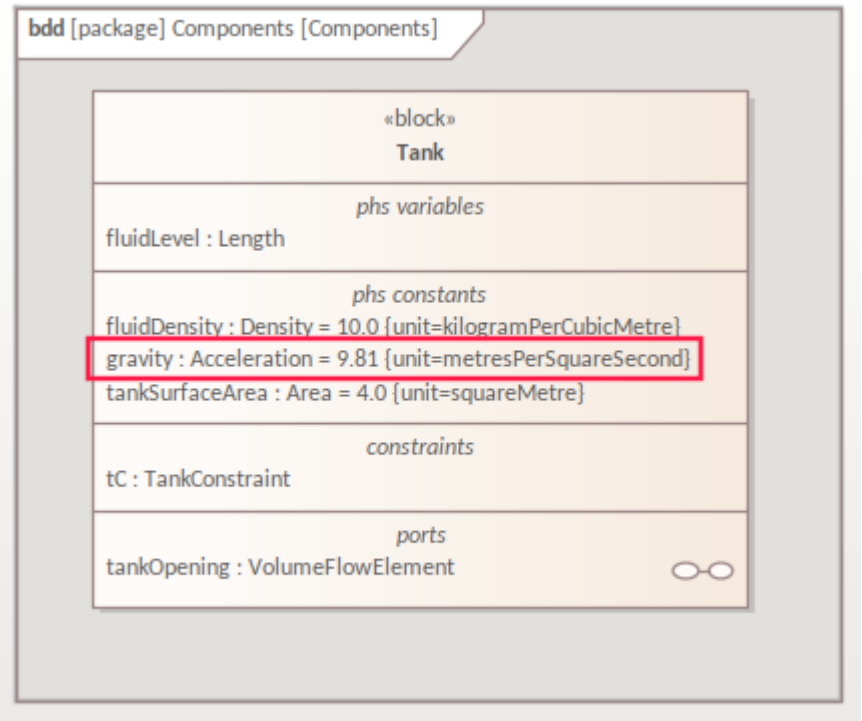
设定值

在 Modelica 或 Simulink 中设置引用组件的块或部件后，您希望能够为该外部组件的属性或参数设置值。在块中而不是在从该块部件的元件中放置值的决定取决于每个部件的该值是否存在任何变化。

块属性

在水箱的示例中，其中定义了常数 'Gravity' 并为每个水箱实例固定，该值最好放在源块中。

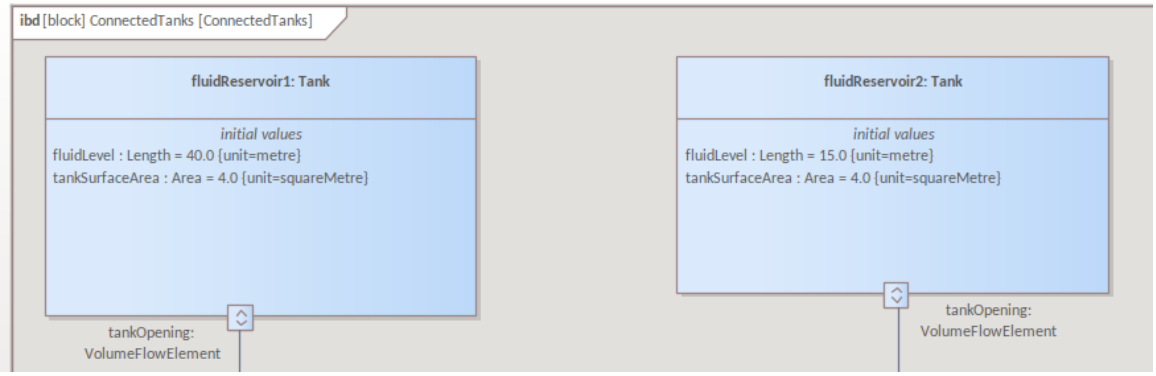
此图还显示了水的流体密度集 (10 kg/m ^{3})，可以在部件中或在模拟中覆盖它以定义另一种流体（例如油）的密度。



部件属性

如果一个块以不同的值重复使用 - 例如表示一个 3.3 kΩ 的电阻器和另一个 5.6 kΩ 的电阻器 - 那么初始值最好在块中定义为空白，在从块派生的特定部分中设置单独的值。

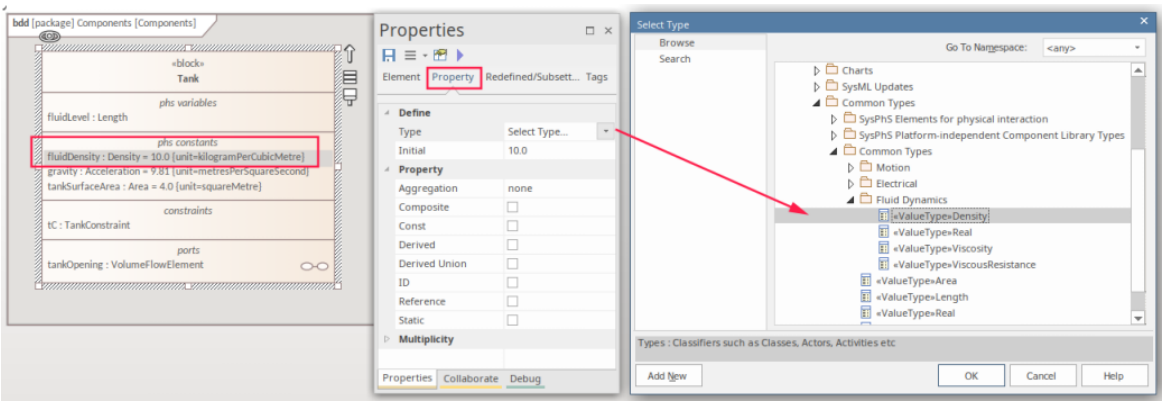
在此示例中，我们有两个半径相同但深度不同的罐，因此在此 IBD 中创建的两个罐部件的流体水平（罐的深度）是不同的。



注记，Type 派生自块类型类型。

设置类型

您可以在块中定义属性以引用特定的值类型。这是在属性> 属性"选项卡上的 类型"字段中设置的；最好使用 … 按钮引用值类型。



有关更多详细信息，请参阅使用值类型建模数量帮助主题。

在仿真中设定值

在仿真中，经常需要在运行时而不是在运行模型中设置变量。这就是数据集可用于在模拟的一系列变体中布置一组值以运行的地方。有关详细信息，请参阅模型分析使用数据集帮助主题。

创建 Modelica 特定的模块

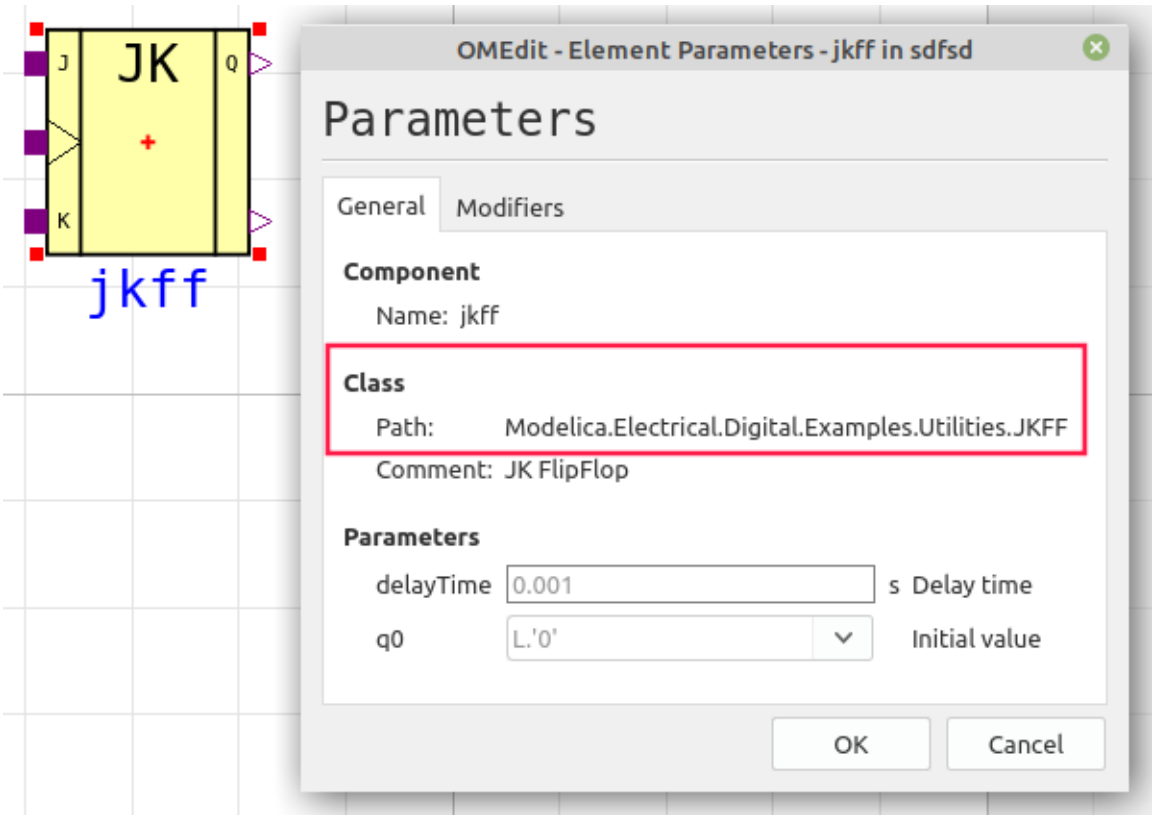
鉴于可在 Modelica 中使用的各种不同零件类型，在某些情况下，您需要模型Modelica 组件和您无法从部件模式中提供的基本块派生的零件。在这种情况下，您可以设置Enterprise Architect组件来引用 Modelica 组件。

为块使用 Modelica类路径

在Enterprise Architect SysPhS 组件中引用 Modelica 组件的过程是：

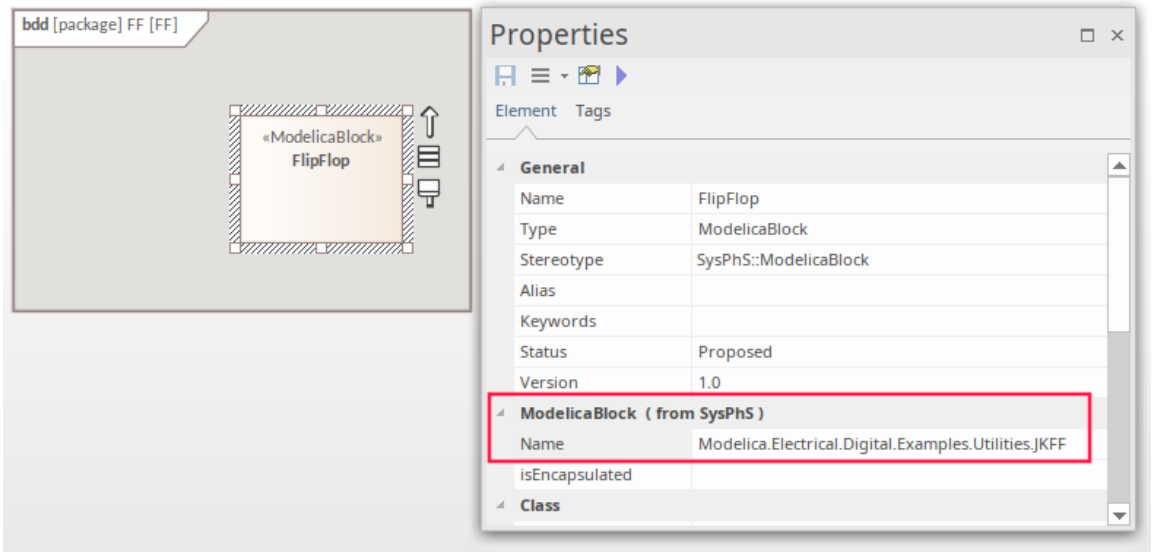
- 在 Modelica 中，访问 Modelica 组件的类路径
- 复制这段文字
- 在Enterprise Architect中，将复制的文本粘贴到 Modelica块的 名称"字段中

此插图是 Modelica 组件（JK 触发器）及其 参数"对话框的示例，显示了类路径。



- 注记：可以通过在路径上单击并拖动光标，然后在文本块上按 Ctrl+C 来选择 Class-Path。

将复制的文本放在属性Modelica块的属性窗口中，在 ModelicaBlock 下的 名称"字段中**（来自 SysPhS）**：。



设置端口

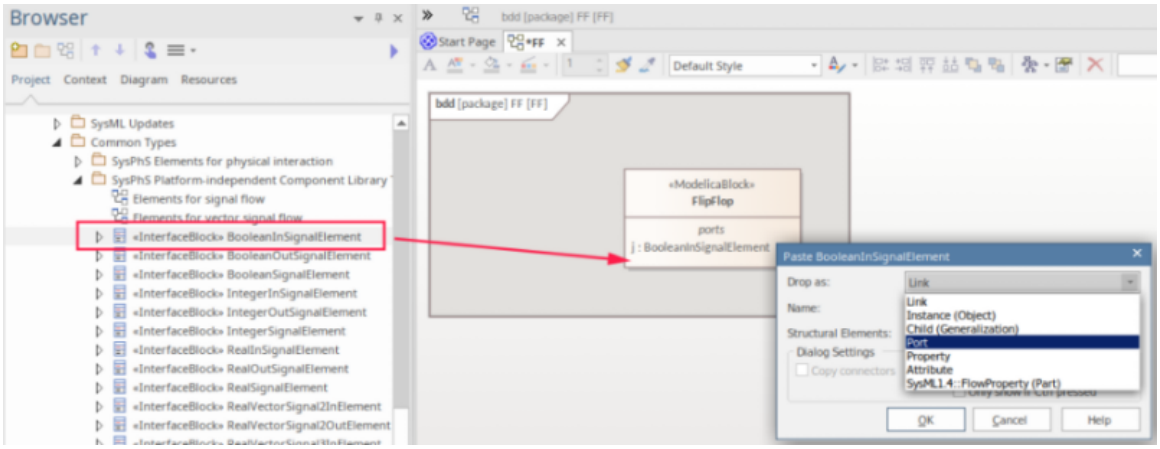
端口上的块可以是非类型的端口或预定义的端口类型。

通过将 Modelica端口从工具箱拖到图表上来创建非类型端口。

引用端口预定义的端口类型 - 例如布尔信号输入端口或模拟信号输出端口：

- 访问浏览器窗口中的访问平台独立部件库中的端口类型
- 将端口类型从库中拖到图表上的块上
- 将其设置为块上的端口

例如，此图显示了在 Flip-端口上创建的 “端口 j”和 “k端口”，在此过程中， “端口j”被设置，“k”被定义为端口。



创建 Simulink 和 Simscape 特定模块

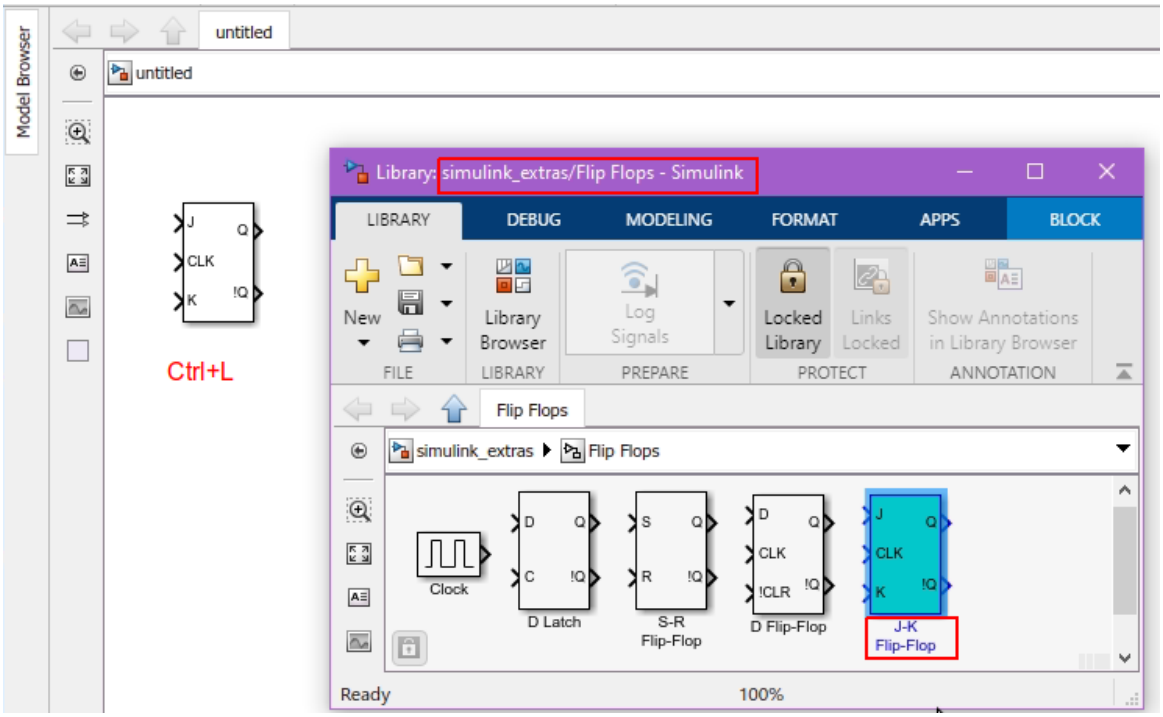
鉴于可在 Simulink 中使用的各种不同零件类型，在某些情况下，您需要模型Simulink 和 Simscape 零件，而这些零件无法从部件模式中提供的基本块派生出来。在这种情况下，您可以设置一个在 SysPhS块中引用该 Simulink 组件类型。这可能是一个块、一个部件或一个部件的部件。

对块使用 Simulink类路径

在Enterprise Architect SysPhS 组件中引用 Simulink 组件的过程是：

- 在包含该组件的 Simulink 图中，单击该组件并按 Ctrl+L 以访问库中的该组件类型
- 在库窗口中，注记窗口标题中的路径，以及窗口正文中部件元素下的部件名称
- 在Enterprise Architect中，在**SimulinkBlock**下的 名称”字段（来自属性）的属性窗口中键入部件路径和名称：

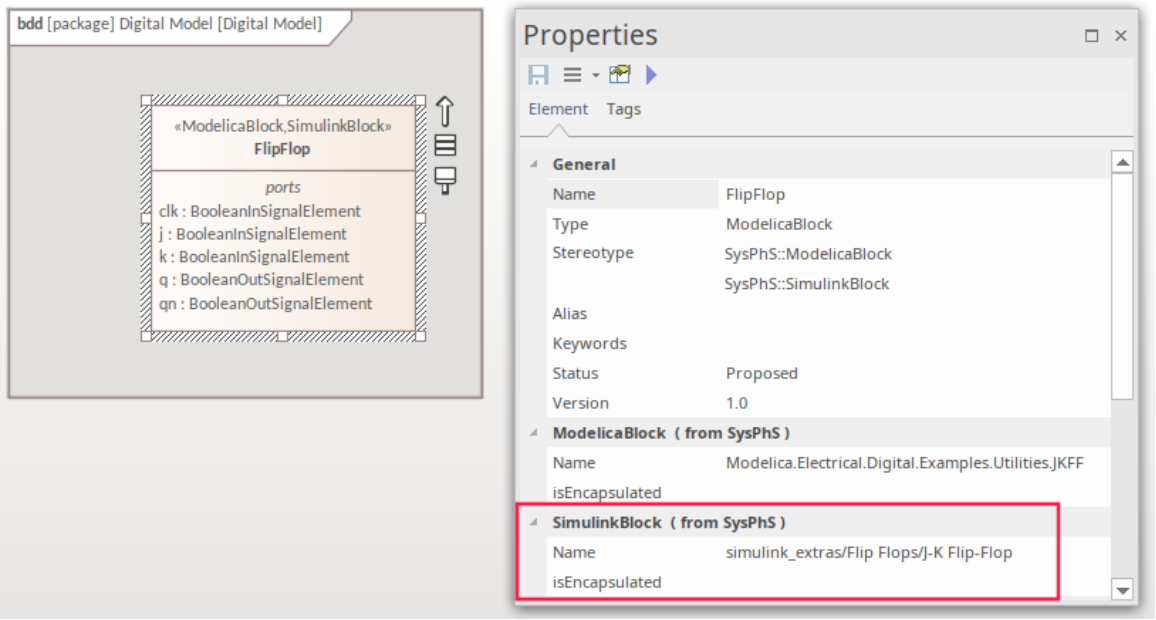
这是 Simulink JK 触发器和 参数”对话框的示例，显示了类路径。



在这种情况下，路径/名称是：

- simulink_extras/触发器/JK触发器

此图显示了添加到属性SimulinkBlock 的属性窗口中的 名称”字段中的文本。



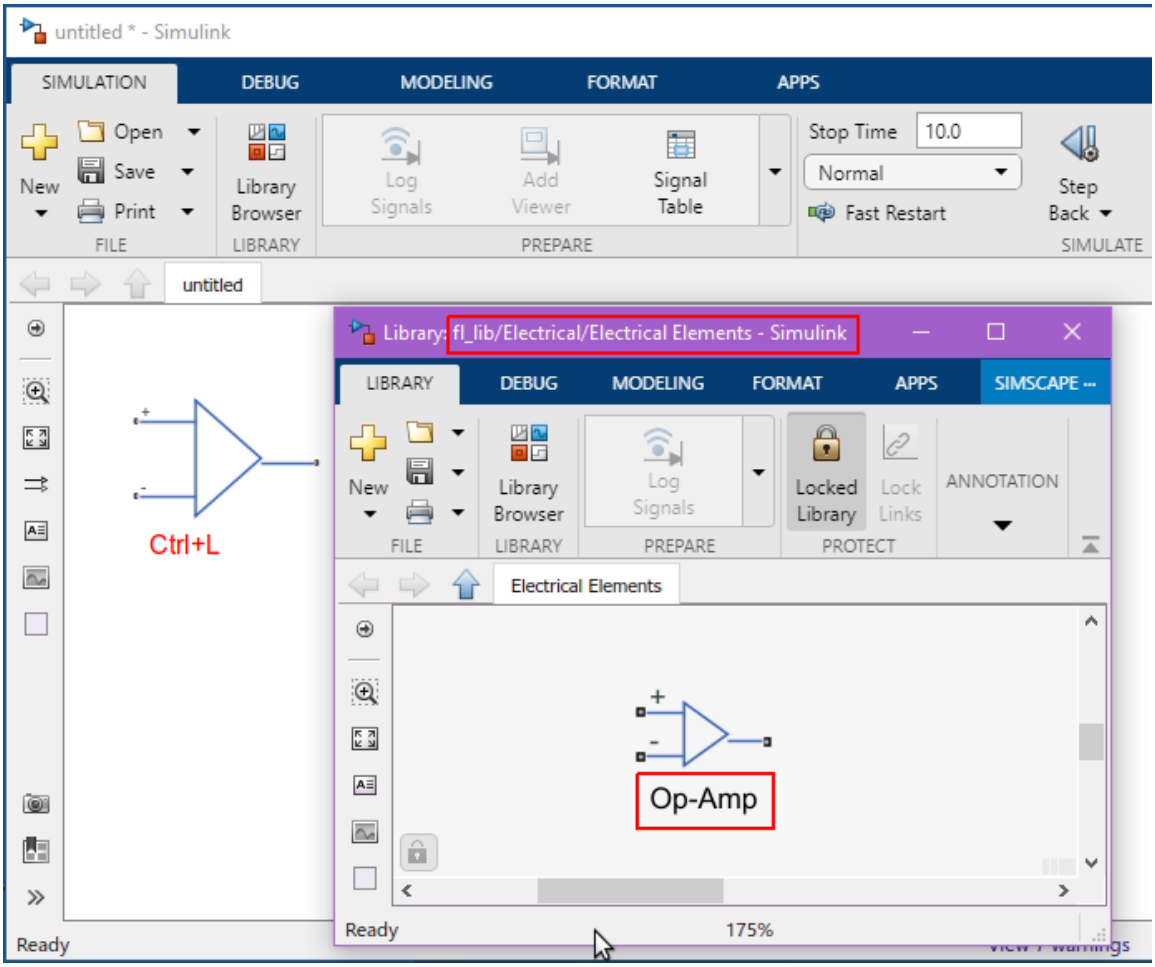
有关设置细节类路径的更多详细信息，请参阅声明成员组件 帮助"帮助主题。

对块使用 Simscape类路径

在Enterprise Architect SysPhS 组件中引用 Simscape 组件的过程是：

- 在包含 Simscape 组件的 Simulink 图中，单击该组件并按 Ctrl+L 以访问库中的该组件类型
- 在库窗口中，注记窗口标题中的路径，以及窗口正文中部件元素下的部件名称
- 在Enterprise Architect中，在**SimulinkBlock**下的 名称"字段（来自属性）的属性窗口中键入部件路径和名称：

以下是运算放大器的 Simscape 组件示例。



注记fl_lib 被 'Foundation' 引用，所以 SysPhS '名称' 是：

- Foundation.electrical.elements.op_amp

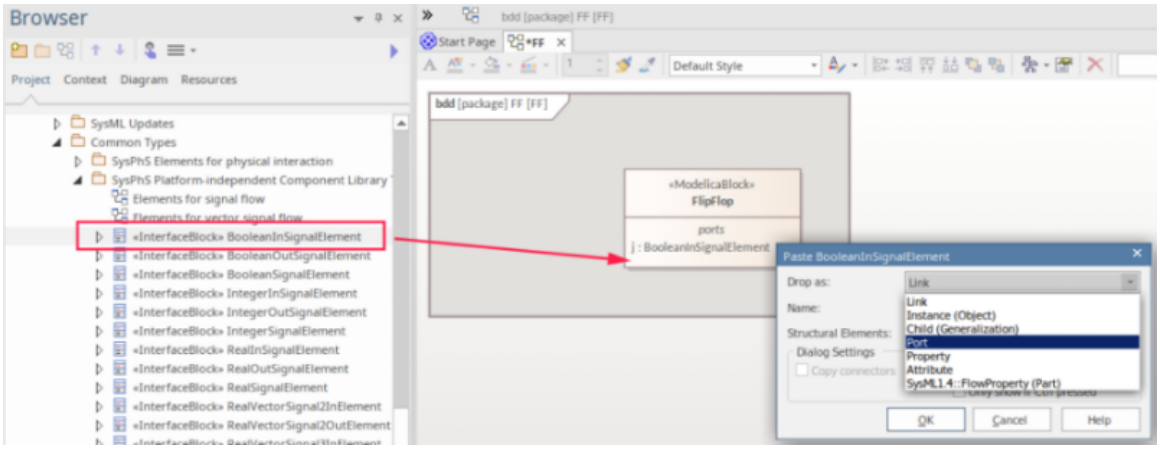
设置端口

端口上的块可以是非类型的端口或预定义的端口类型。

未定义端口是通过将 Simulink端口从工具箱拖到图表上来创建的。

要引用 SysPhS 预定义的端口类型，例如布尔信号输入端口或模拟信号输出端口，请在浏览器窗口中打开部件平台无关部件库。 A类型可以从库中拖到块上，并设置为块上的端口。

例如，这是在触发器上创建 j 和 k端口的过程的快照，显示进程中的端口设置和 k端口被定义为端口。



Simulink端口排序

Simulink端口是使用数组定义的（与 Modelica 中的名称引用相反），因此端口的创建顺序至关重要。 IN端口的排序与 OUT端口的排序是分开的。端口排序可以在 Simulink 中查看；第一个端口显示在块的顶部。

使用 Flip Flop端口示例，IN端口的 Simulink 排序将是：

j，clk，k，（参见本主题第一张图片中的 JK Flip Flop"元素）

因此，这些端口必须按Enterprise Architect的顺序创建。注记指出，端口在块定义图中按字母顺序显示，而不是创建顺序。

未应用排序A常见情况是 OUT端口在 SysML 图中正确显示，但在 Simulink 中模拟时未正确连接到 IN端口。如果发生这种情况，请确保正确应用端口的创建顺序。

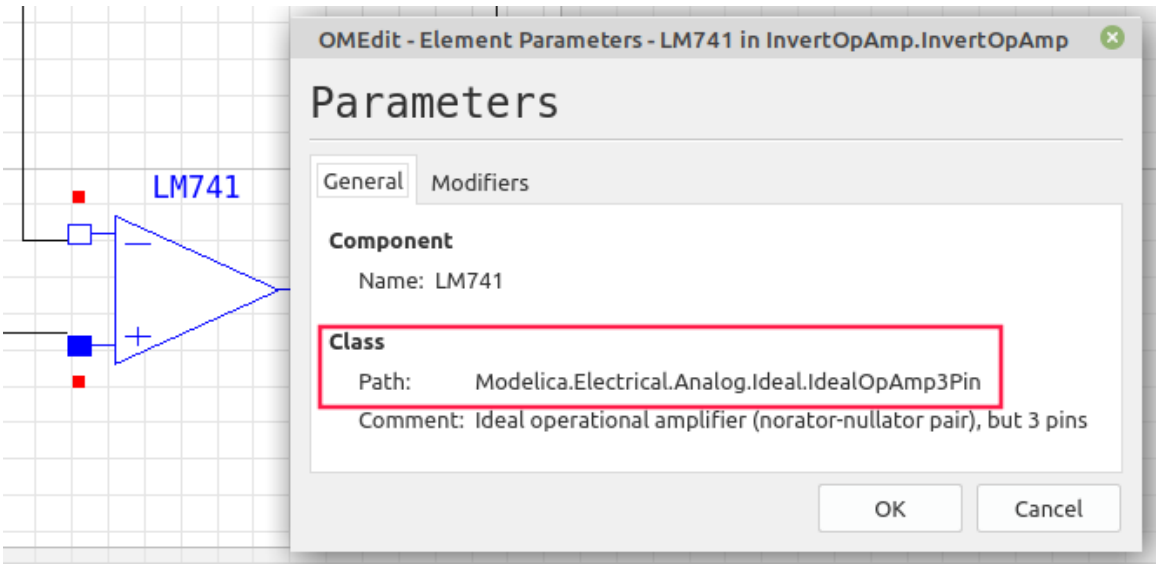
将模块设置为 Modelica 和 Simulink

对于同时使用 Modelica 和 Matlab 的情况，或设置通用模板以涵盖两种产品，您可以在 SysPhS块中设置对两种产品的组件类型的引用。

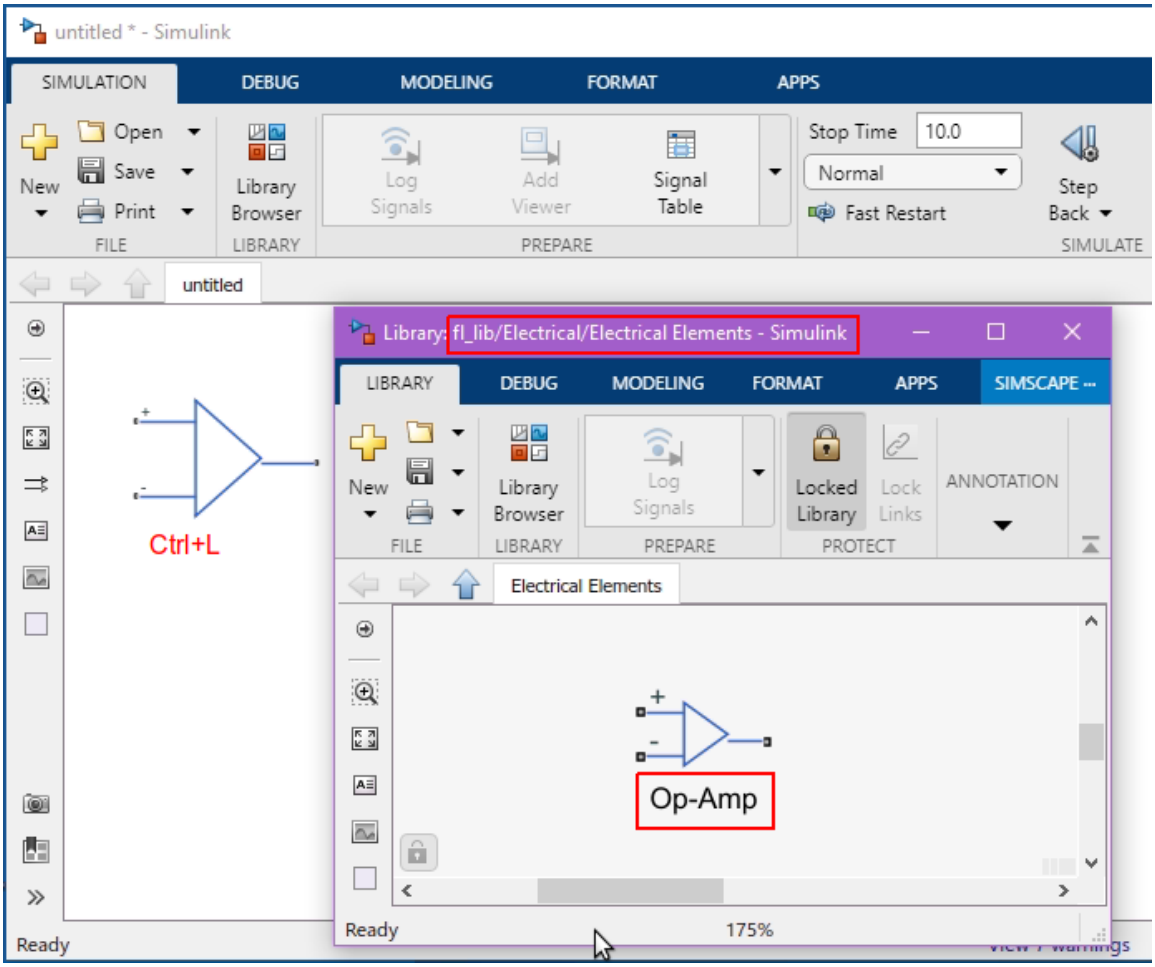
SysPhS模式附带的预定义模块已经包含 Modelica 和 Simulink属性。

示例

以下是运算放大器的 Modelica 组件示例。

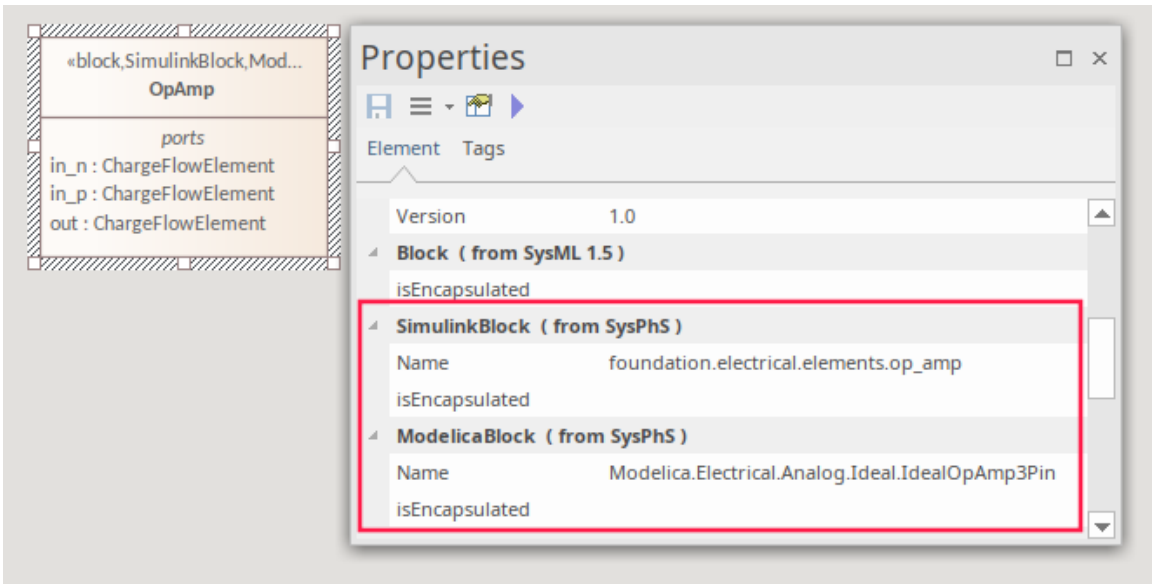


以下是运算放大器的 Simscape 组件示例。



注记由于 fl_lib 被 "foundation"引用，SysPhS 的 "名称"是*foundation.electrical.elements.op_amp*。

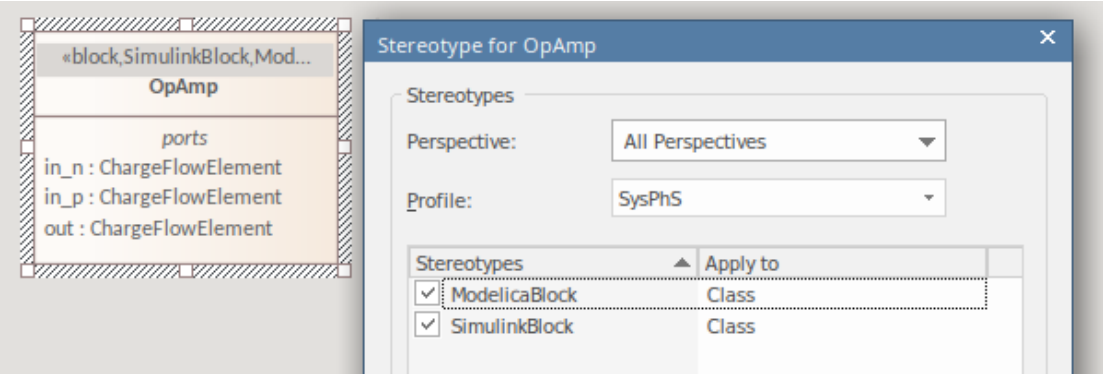
以下是 Modelica 和 Simulink 的 "名称"设置。



要为这两种产品手动设置属性，您必须为这两种产品设置构造型。要在新创建的 SysPhs块元素上执行此操作，请在该元素的属性窗口中：

- 选择 "构造型"
- 单击 [...] 按钮
- 在配置文件中选择SysPhS

- 勾选 SimulinkBlock 和 ModelicaBlock 原型



仿真

Enterprise Architect提供与 OpenModelica 和 MATLAB 的 Simulink 的集成，以支持对 SysML模型在不同情况下的行为方式进行快速而可靠的评估。

OpenModelica 库是提供许多有用类型、函数和模型的综合资源。在Enterprise Architect中创建 SysML 模型时，您可以参考这些库中可用的资源。

Enterprise Architect的 MATLAB 集成通过 MATLAB API 进行连接，允许您的Enterprise Architect仿真和其他脚本根据任何可用的 MATLAB 函数和表达式的值进行操作。您可以将模型导出到 MATLAB Simulink、Simscape 和/或状态流。

本节描述了定义内部块图表或参数模型、使用附加信息对模型进行注释以驱动模拟以及运行模拟以生成结果图的过程。

SysML内部块和参数模型介绍

SysPhS 模型支持关键系统参数的工程分析，包括评估关键指标，如性能、可靠性和其他物理特性。这些模型通过捕获基于复杂数学关系的可执行约束，将需求模型与系统设计模型相结合。参数图是专门的内部块图，可帮助建模者将行为和结构模型与工程分析模型（例如性能、可靠性和质量属性）结合起来。

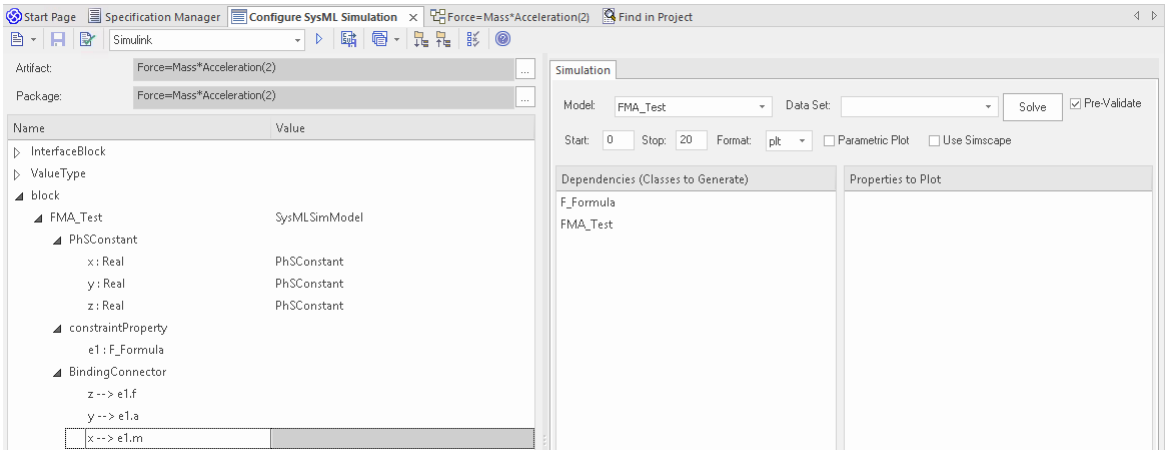
有关 SysML 参数模型概念的更多信息，请参阅 OMG SysML 官方网站及其链接资源。

这些主题描述了使用 SysML 和 SysPhS 定义和运行系统工程仿真的关键选项。

话题	描述
SysPhS 标准支持	*SysPhS* 标准是交互物理和辅助信号流仿真的 *SysML*扩展。它定义了在SysML模型与 Modelica模型或 Simulink/Simscape模型之间进行转换的标准方法，为共享仿真提供了一种更简单的基于模型的方法。
工件适合	Enterprise Architect帮助您扩展 SysML 参数模型的实用性，方法是使用允许模拟模型的额外信息对它们进行注释。然后将生成的模型生成为可以使用 MATLAB Simulink 或 OpenModelica 求解（模拟）的模型。 模拟属性针对您的仿真模型进行工件。这保留了您的原始模型并支持针对单个 SysML模型配置的多个模拟。仿真工件在工件的工具箱上找到
用户接口	SysML 仿真的用户界面在配置仿真主题中进行了描述。
查看生成的模型	为 Modelica 或 Simulink 生成模型后，您可以直接在外部应用程序中使用该模型。
SysPhS调试技巧	由于可能有许多因素会导致生成的脚本出错，因此我们提供了一些关于在模型中隔离问题源的提示。
模型分析使用数据集	在使用模拟时，可能会出现要测试多个变体的情况。为了支持这一点，可以针对 Blocks 定义多个数据集，从而允许使用相同的 SysML模型进行可重复的模拟变化。
SysPhS仿真实例	为了帮助您了解如何创建和模拟 SysPhS模型，我们提供了三个示例来说明三个不同的领域。所有三个示例都使用 OpenModelica 库。仿真仿真示例主题中描述了这些示例以及您能够从中学到的东西。

配置SysML仿真

配置SysML仿真窗口是一个界面，您可以通过该界面提供运行时参数以执行 SysML模型的模拟。工件中的模拟基于元素定义的模拟配置。

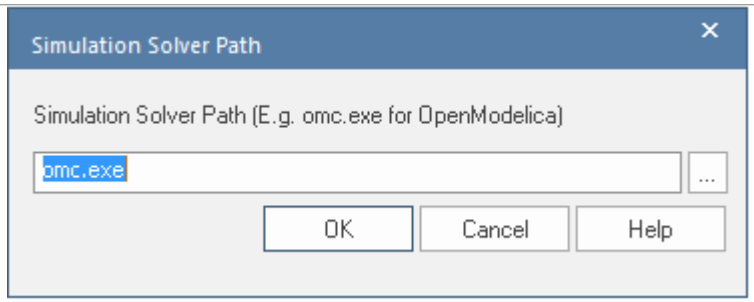


访问

功能区	仿真>系统行为> Modelica/Simulink >配置管理器
其它	双击具有工件构造型的属性。

工具栏选项

选项	描述
	单击下拉箭头并从以下选项中进行选择： • （如果尚未选择工件配置，则从一个工件配置中选择一个配置和负载） • 创建工件— 创建新的工件或选择并加载现有的配置 • 选择包- 选择一个包以扫描 SysML 元素以配置模拟 • Reload — 重新加载配置管理器并更改当前包 • 配置仿真求解器 - 显示 仿真求解器路径”对话框，您可以在其中键入或浏览要使用的求解器的路径。对于 MATLAB/Simulink，将自动检测路径，因此只有在自动检测存在问题或安装了多个版本的 MATLAB 时才需要更改此设置。

	
	单击此按钮将配置保存到当前工件。
	单击此图标现在专门针对现在配置验证模型。验证结果显示在系统输出窗口的 SysML仿真”选项卡中。您还可以选择一个选项以在执行每个模拟之前自动预验证模型。请参阅仿真标签表中的 预验证”选项。
	单击此图标可展开窗口 名称”列中层次结构中的每个项目。
	单击此图标可折叠窗口 名称”列中模型层次结构中所有展开的项目。
	单击此图标可显示可在模拟中抑制的object类型列表。单击要抑制的每个object的复选框，或单击 全部”按钮以选择要抑制的所有项目。 您还可以使用 选项”列顶部的过滤器栏来仅显示名称中具有指定字母或文本string的项目。
	单击下拉箭头并选择正在运行仿真的应用程序，例如运行或 Simulink。
	单击此按钮可生成、编译和运行当前配置，并显示结果。
	仿真后，以 plt、mat 或 csv 格式生成结果文件。也就是说，使用文件名： • ModelName_res.mat（OpenModelica 的默认值） • ModelName_res.plt 或 • 模型名称_res.csv 单击此按钮指定Enterprise Architect将复制结果文件的目录。
	单击此按钮可从以下选项中进行选择： • 运行Last Code - 执行最近生成的代码 • 生成代码——生成编译或运行代码 • Open仿真— 打开将生成 OpenModelica 或 Simulink 代码的目录 • 编辑模板——使用代码模板编辑器自定义为 OpenModelica 或 Simulink 生成的代码

仿真工件模型

字段	行动

工件	单击…并选择现有的工件图标，然后创建一个新的工件。
包	如果您有一个现有的工件配置模型包，该字段指定了与该属性相关联的默认工件。 否则，单击…图标并浏览并选择包含 SysML模型的包以进行仿真配置。您必须在选择包之前指定（或创建）工件。

包对象

本表讨论了将在配置SysML仿真窗口的 名称”列下列出的 SysML模型中的object类型，这些对象将在模拟中进行处理。每个object类型展开以列出该类型的命名对象，以及需要在 值”列中配置的每个object的属性。

很多级别的object类型、名称和属性都不需要配置，因此对应的 Value”字段不接受输入。如果输入合适且被接受，则在字段的右端会显示一个下拉箭头；当您单击此箭头时，将显示一个可能值的简短列表以供选择。某些值（例如部件的 部件”）添加更多的参数层和属性，您可以单击…按钮再次选择和设置参数的值。对于数据集，输入对话框允许您输入或导入值，例如初始值或默认值；请参阅模型分析使用数据集帮助主题。

元素类型	行为
值类型	ValueType 元素要么从原始类型泛化，要么被 SysMLSimReal 替代以进行仿真。
块	映射到 SysMLSimClass 或 SysMLSimModel 元素的块元素支持数据集的创建。如果您在上下文中定义了多个数据集（可以泛化），则必须将其中一个标识为默认值（使用时间菜单选项 设置为默认数据集”）。 由于 SysMLSimModel 可能是模拟的顶级元素，并且不会被概括，如果您定义了多个数据集，则在模拟期间选择要使用的数据集。
属性	指定常量或变量及其设置的首选方法是在属性本身上使用 SysPhS 构造型 PhSConstant 和 PhSVariable。 PhSVariable 构造型具有内置属性*isContinuous*、*isConserved*和*changeCycle*。 该属性将列在 PhSConstant 或 PhSVariable 下，并且该值不能更改。 也可以在配置SysML仿真窗口中定义设置。在这种情况下，它们将列在 属性”下。 块内的属性可以配置为 SimConstants 或 SimVariables。对于 SimVariable，您可以配置以下属性： • isContinuous — 确定属性值是连续变化（'true'，默认值）还是离散变化（'false'） • 属性— 确定属性的值是否保留（'true'）（'false'，默认值）；在为物理相互作用建模时，相互作用包括保守物理物质的交换，例如电流、力或流体流动 • changeCycle — 指定离散属性值更改的时间间隔；默认值为 0” - changeCycle 只能设置为 0 以外的值 isContinuous = 'false' - changeCycle 的值必须为正或等于 0
端口	无需配置。
模拟函数	函数被创建为块或约束块中的操作，原型为 SimFunction”。 配置仿真窗口无需配置。

概括	无需配置。
捆绑连接器	将属性绑定到约束属性的参数。 无需配置；但是，如果属性不同，系统会提供同步它们的选项。
连接器	连接两个端口。 配置仿真窗口无需配置。但是，您可能必须通过确定属性属性是否应该设置为 "False"（对于潜在的，以便建立相等耦合）或 "True"（对于流/保守的属性）来配置端口类型的属性, 从而建立和到零的耦合)。
约束块	无需配置。

仿真标签

此表描述了配置SysML仿真窗口上的 仿真"选项卡的字段。

字段	行动
模型	单击下拉箭头并选择模拟的顶级节点（SysMLSimModel元素）。该列表填充有定义为顶级模型节点的块的名称。
数据集	单击下拉箭头并选择所选模型的数据集。
预验证	选中此复选框可在执行模型的每次模拟之前自动验证模型。
开始	类型在开始模拟之前的初始等待时间，以秒为单位（默认值为 0）。
停止	类型模拟将执行的秒数。
格式	单击下拉箭头并选择 "plt"、 "csv"或 "mat"作为结果文件的格 ，其他工具可能会使用这些格式。
参数图	• 选中此复选框以在 y 轴上绘制图例A ，在 x 轴上图例B • 取消选中复选框以在 y 轴上绘制图例，在 x 轴上绘制时间 注记：选中复选框后，您必须选择两个要绘制的属性。
使用Simscape	（如果选定的数学工具是 Simulink）如果您还想在 Simscape 中处理仿真，请选中该复选框。
依赖项	列出模拟该模型必须生成的类型。
属性的属性	提供与模拟有关的变量属性列表。选中要绘制的每个属性对应的复选框。

查看生成的模型

一旦一个模型生成到 Modelica 或 Simulink 中，您可能希望在外部应用程序中直接使用该模型。这包括模型未正确生成并且您需要在外部应用程序中查找问题的任何情况。

使用运行或 Solve 按钮  模型生成，然后查看生成的代码。

查看生成的代码

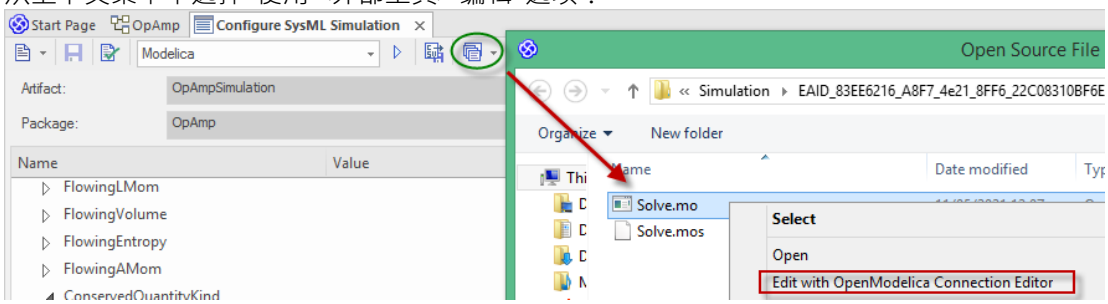
在Enterprise Architect代码编辑器中查看生成的 Modelica 或 Simulink 代码：

- 单击工具栏中的  图标
- 选择 打开仿真目录”选项
  这将打开生成 OpenModelica 或 Simulink 代码的目录
- 点击文件查看脚本

访问生成的模型

使用生成的代码访问 Modelica 或 Simulink模型：

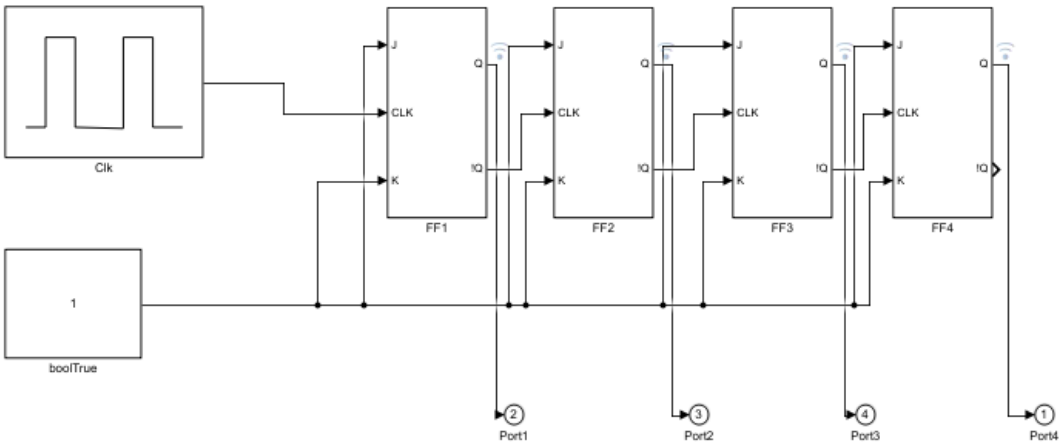
- 单击工具栏中的  图标
- 选择 打开仿真目录”选项
  这将打开生成 OpenModelica 或 Simulink 代码的目录
- 右键单击文件
- 从上下文菜单中选择 使用 <外部工具> 编辑”选项：



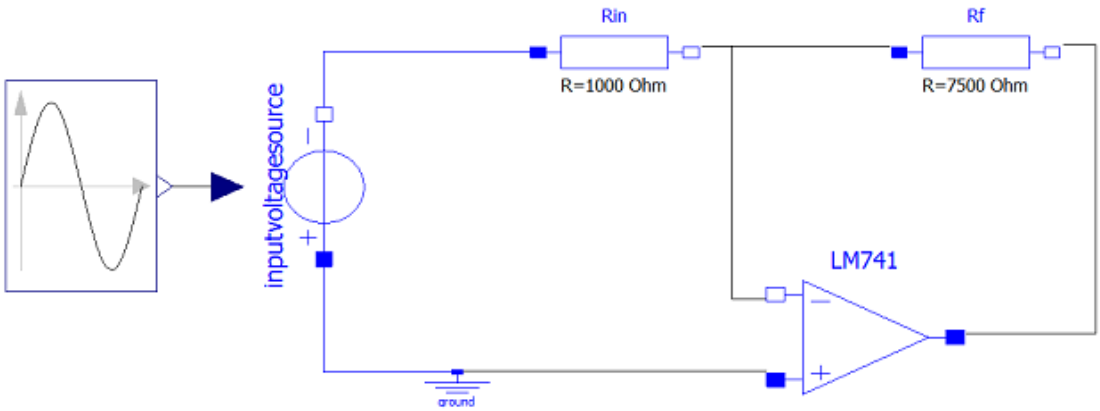
在这种情况下，会显示一个 Modelica .mo 文件。对于 Simulink 生成的文件，文件名为 Solve.m。

双击文件打开外部应用程序：

- 对于 Simulink，要查看生成的模型，必须打开以 SysMLSimModel 设置的块命名的 .slx 文件；这是在 Simulink 中生成并打开的 Flip-flop 示例的视图：



- 对于 Modelica，在 Libraries 浏览器中，图表按照 SysMLSimModel块命名 - 双击打开它；这是 OpAmp 示例的视图，在 Modelica 中生成并打开：



有关详细信息，请参阅配置*SysML*仿真帮助主题。

SysPhS调试技巧

与从模型生成的任何代码一样，在外部应用程序中使用的生成脚本中可能有许多因素会导致错误。因此，需要找到问题原因的选项。本主题提供了一些关于在模型中隔离问题源的提示。

在检查生成的代码之前，必须使用  或 Solve 按钮生成外部脚本。

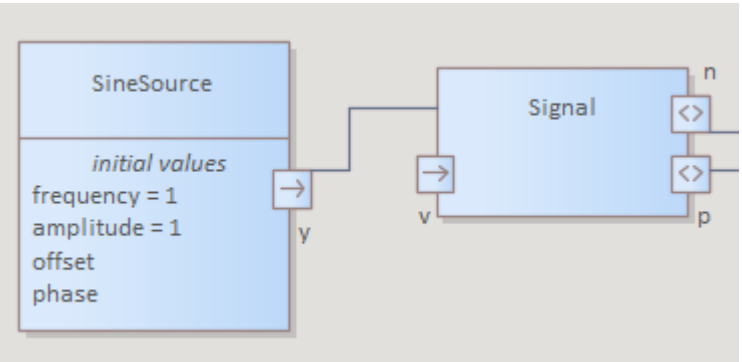
发现问题

要查找生成的脚本的任何问题，要使用的关键选项包括：

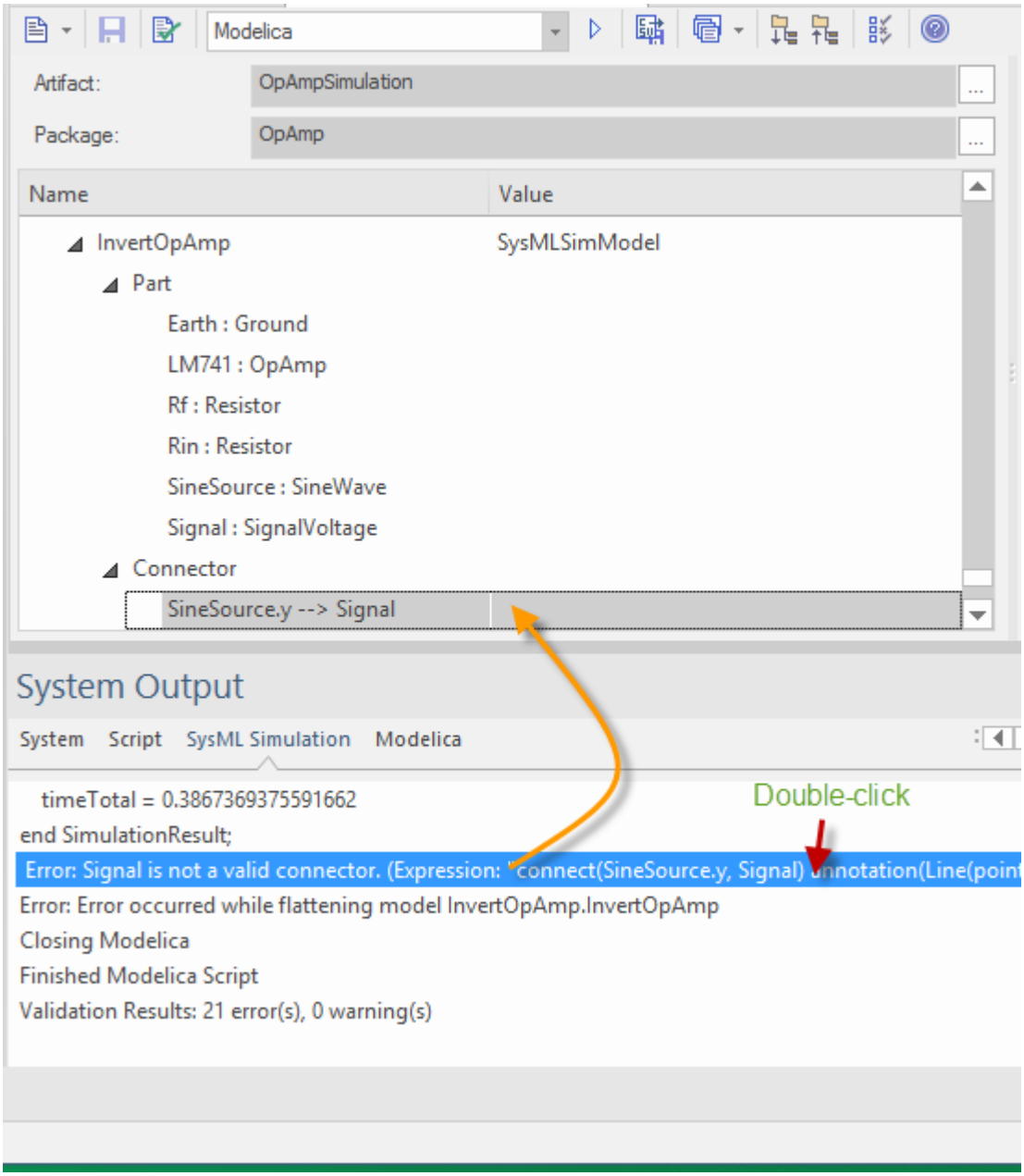
- 对于生成过程，运行的步骤运行和生成的任何错误都会记录在系统输出视图中（Ctrl+Shift+8）
- 对于系统输出窗口中列出的任何错误，如果它与特定元素相关，则双击该错误将在配置仿真窗口的左窗格中找到关联的object
- 如果需要访问生成的脚本，请使用工具栏上  图标下的 打开仿真目录”选项；请参阅查看生成的模型帮助主题

示例

在本例中，我们在 IBD 中有一个部件错误地链接到另一个部件。即直接链接，而不是通过端口的链接。这会在同时生成到 Modelica 和 Simulink 时导致错误。



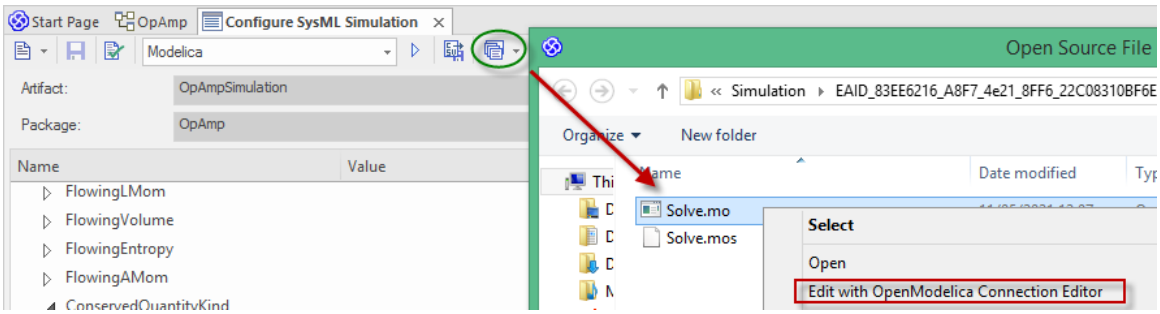
系统输出中产生的是错误。双击该错误会突出显示仿真列表中的相关object 。



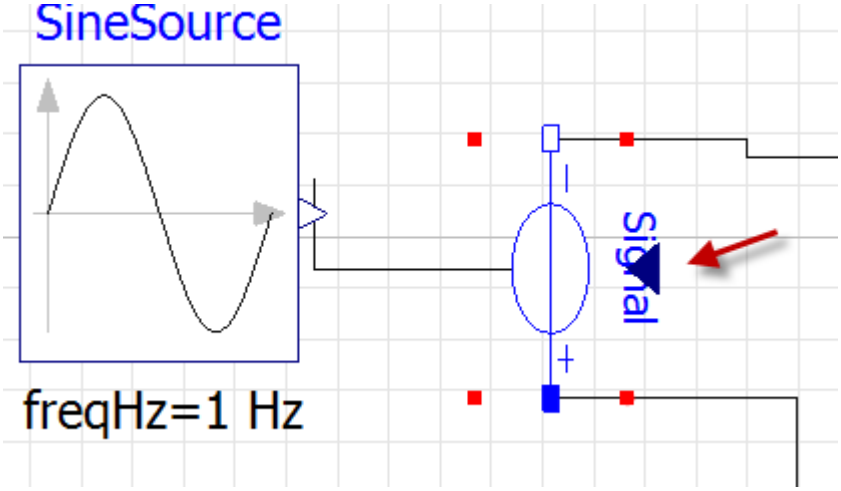
这可以通过在外部应用程序中打开生成的脚本并查看生成的图表来进一步验证，方法是：

- 单击工具栏中的  图标
- 选择打开仿真目录选项
- 使用上下文菜单在外部应用程序中打开文件

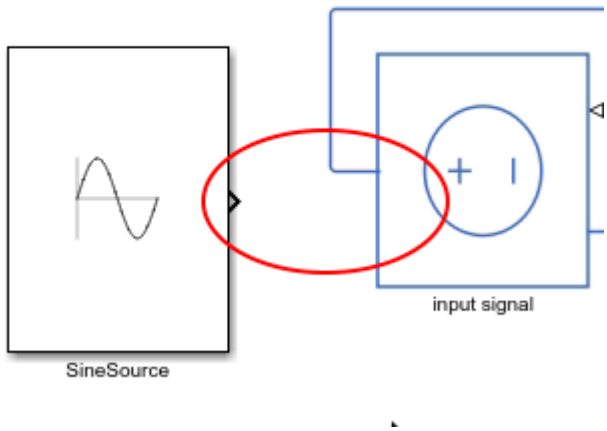
例如，在 Modelica 中的这种情况下：



在 Modelica 中打开生成的模型时，使用工具栏上的  图标，此视图显示连接器未链接到导入信号object的三角形：



与 Simulink 类似，这是生成的模型，显示 SineOutput 缺少连接器：



使用数据集进行模型分析

参数模型中使用的每个 SysML块都可以在仿真配置中定义多个数据集。这允许使用相同的 SysML模型进行可重复的模拟变化。

块A键入为 SysMLSimModel（不能被概括或构成组合的一部分的顶级节点）或 SysMLSimClass（可以被概括或构成组合的一部分的较低级别的元素）。在 SysMLSimModel元素上运行模拟时，如果您定义了多个数据集，则可以指定要使用的数据集。但是，如果模拟中的类有多个数据集，则您无法选择在模拟期间使用哪一个，因此必须将一个数据集标识为该类的默认值。

访问

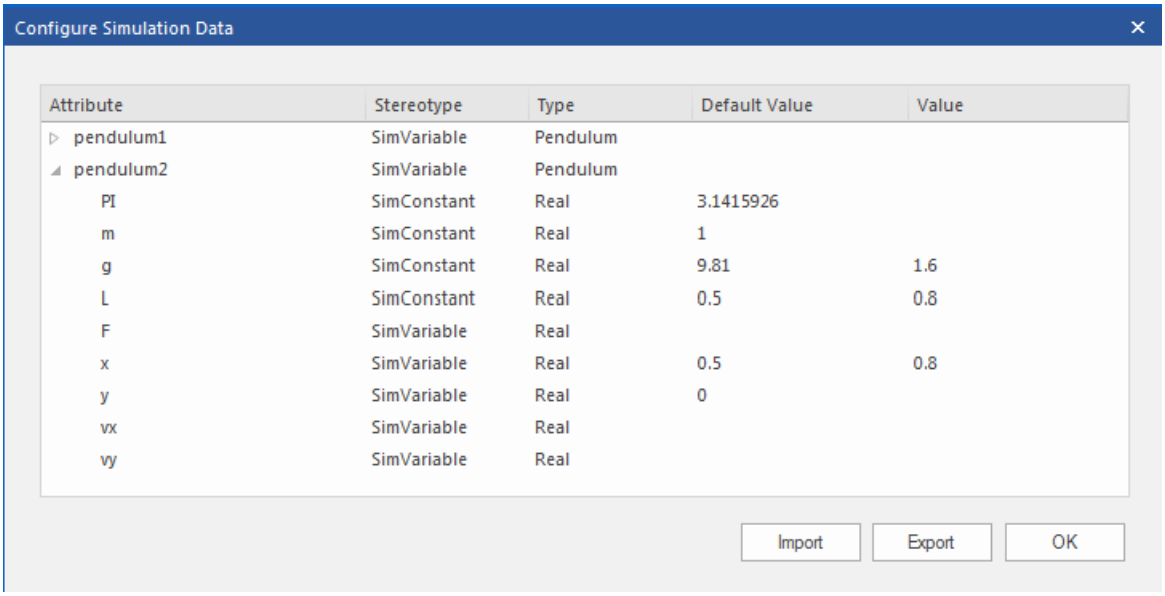
功能区	仿真>系统行为>Modelica/Simulink>配置管理器>在 块"组>名称列>块元素上下文菜单>创建仿真数据集

数据集管理

任务	行动
创造	要创建新数据集，请右键单击块名称并选择 创建仿真数据集"选项。数据集被添加到块名称下方的组件列表的末尾。单击 ... 按钮在 配置仿真数据"对话框中设置数据集（参见配置仿真数据表）。
复制	要复制现有数据集作为创建新数据集的基础，请右键单击数据集名称并选择 复制"选项。重复的数据集将添加到块名称下方的组件列表的末尾。单击 ... 按钮以编辑 配置仿真数据"对话框中的数据集（请参阅配置仿真数据表）。
删除	要删除不再需要的数据集，请右键单击数据集并选择 删除数据集"选项。
默认设置	要设置 SysMLSimClass 在用作属性类型或继承（以及当有多个数据集时）使用的默认数据集，请右键单击数据集并选择 设置为默认值"选项。默认数据集的名称以粗体突出显示。模型使用的属性将使用此默认配置，除非模型明确覆盖它们。

配置仿真数据

此对话框主要用于提供信息。您可以直接添加或更改数据的唯一列是 值"列。



柱子	描述
属性	属性”列提供了正在编辑的块中所有属性的树视图。
构造型	对于每个属性， 构造型”列标识它是否已配置为模拟期间的常量，或者其值预计会随时间变化的变量。
类型	类型”列描述了用 模拟此属性的类型。它可以是原始类型（例如 Real”）或对模型中包含的块的引用。属性引用块将显示由它们下面的引用块指定的子属性。
默认值	如果未提供覆盖， 默认值”列显示将在模拟中使用的值。这可以来自SysML模型中的 初始值”字段或来自父类型的默认数据集。
价值	值”列允许您覆盖每个原始值的默认值。
导出/导入	单击这些按钮可使用电子表格等外部应用程序修改当前数据集中的值，然后将它们重新导入列表。

SysPhS仿真实例

本节为每个阶段提供了一个工作示例：为域创建 SysML模型，对其进行仿真，并评估仿真结果。这些示例应用了前面主题中讨论的信息。

例子

模型	描述
模拟电路运算放大器	第一个例子是模拟一个简单的电子电路。该示例从定义电路组件的块的块定义图开始，以及包含概述电路的内部块定义图的块。然后对模型进行仿真，评估运算放大器的源端和目标端的sine电压，并将其与预期值进行比较。
数字电子仿真示例	第二个示例显示了在 Modelica 和 Simulink 中预定义的标准触发器组件，用于创建基于触发器的简单数字信号分频器。这表明如何使用 SysPhS 表示法来引用 Modelica 和 Simulink 中可用但未在 SysPhS 模式中定义的组件。
加湿器仿真示例	该示例讨论了使用 SysML状态机图来定义运行中的加湿器的开关状态。这是生成到 Modelica 和 MATLAB 的 StateFlow 图进行仿真。

了解更多（视频和WebEA ）

- Simulink 中的 SysML仿真
- SysML Tank示例
- MATLAB状态流状态for状态机
- SysPhS 运算放大器示例
- SysPhS 启动器
- 使用仿真模拟数字电子学
- WebEA ： EA Sim 示例

运算放大器电路仿真示例

在本例中，我们为一个简单的电子电路创建了一个 SysPhS模型，然后使用模拟来预测和绘制该电路的行为。

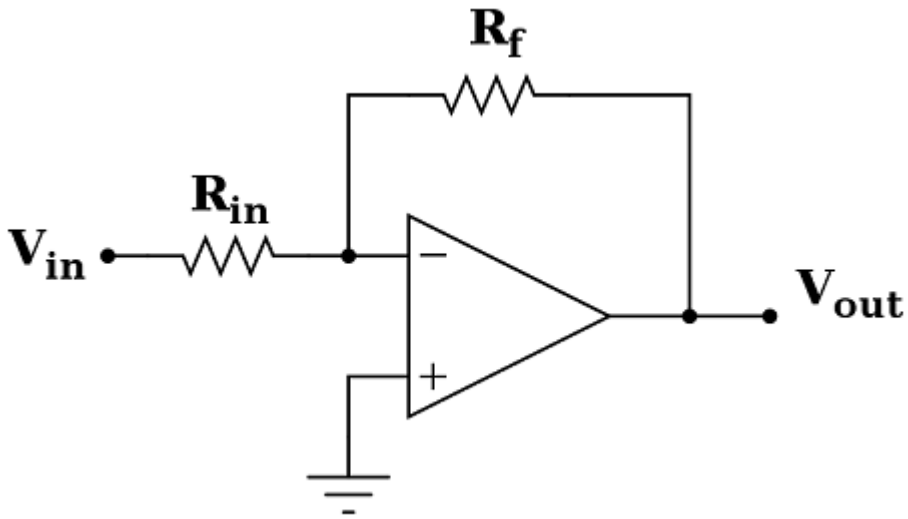
先决条件

运行此模拟需要：

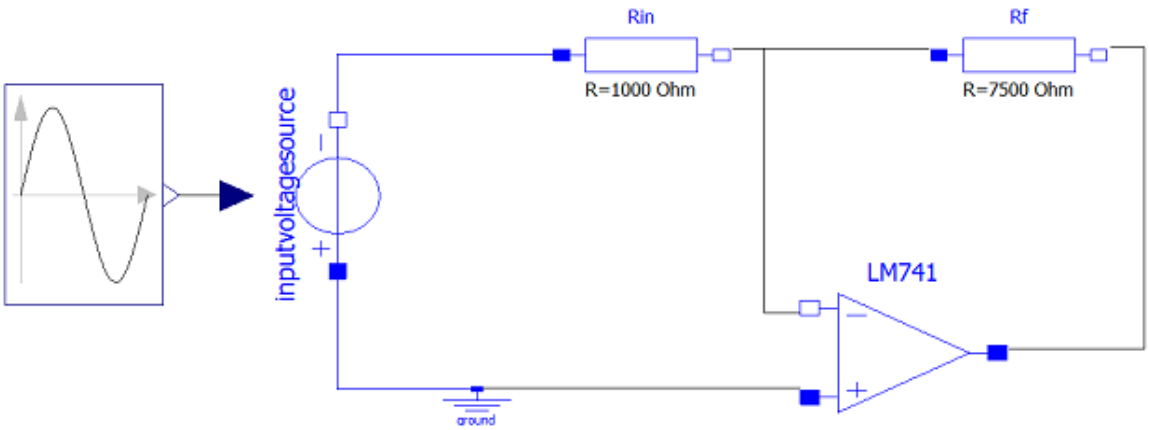
- OpenModelica 或
- MATLAB 的 Simulink 和 Simscape

图表

我们将要学习的电子电路模型如图所示，使用标准电子电路符号。

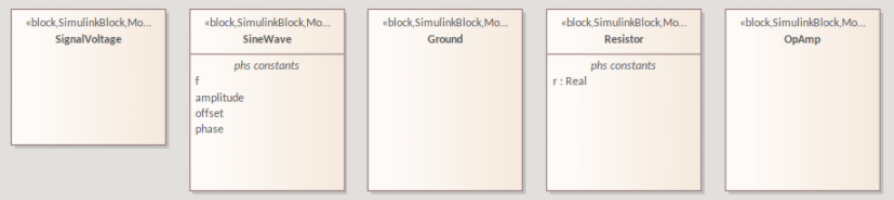
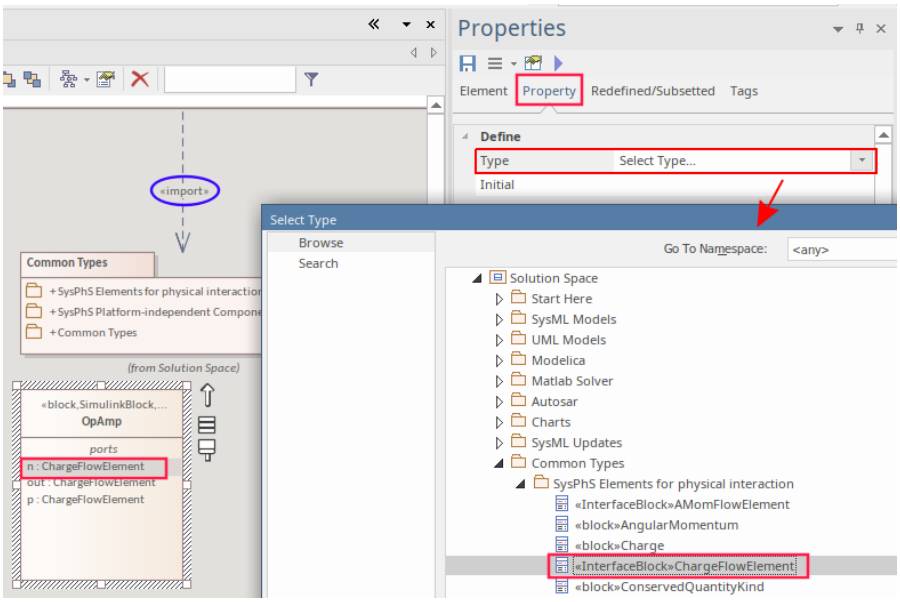


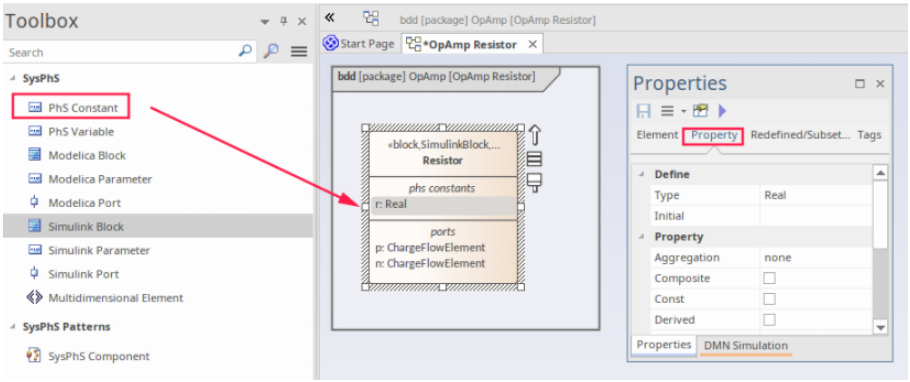
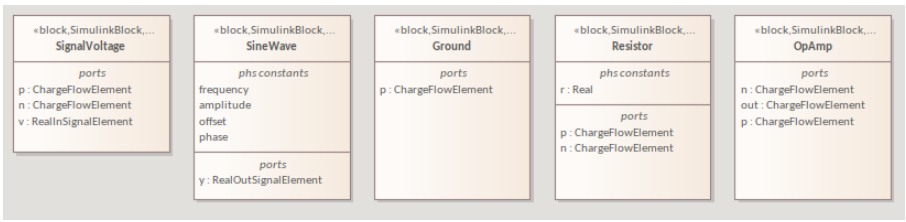
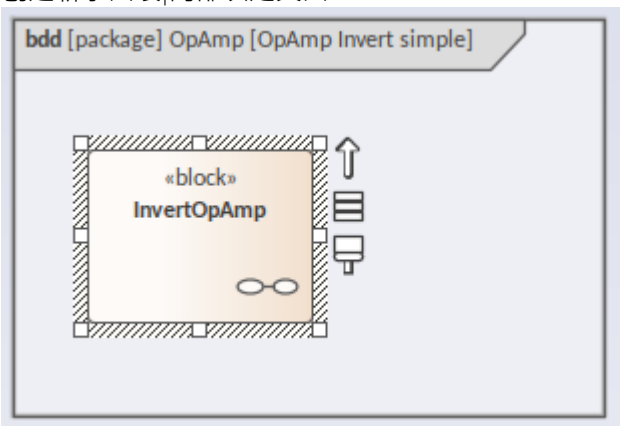
本例中的电路将包括一个sine信号源、一个接地、两个电阻和一个运算放大器集成电路。下图显示了使用 SysPhS 从 SysML 图生成到外部应用程序（在本例中为 Modelica）的结果图。

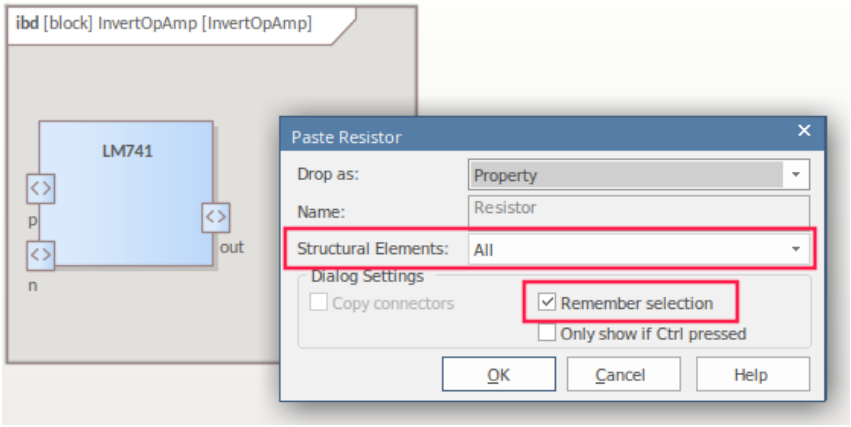
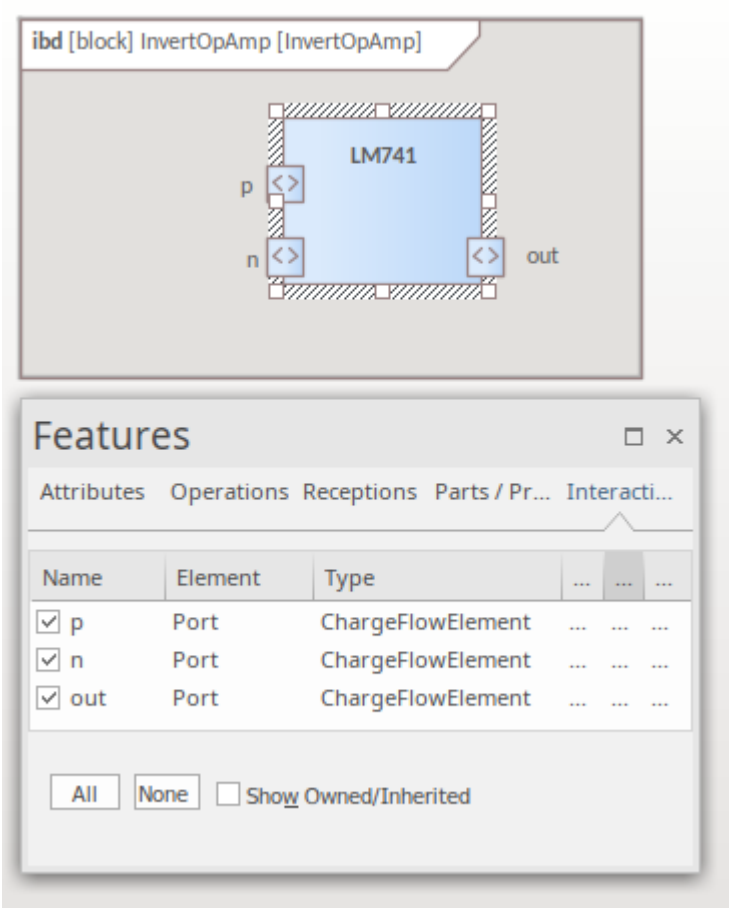


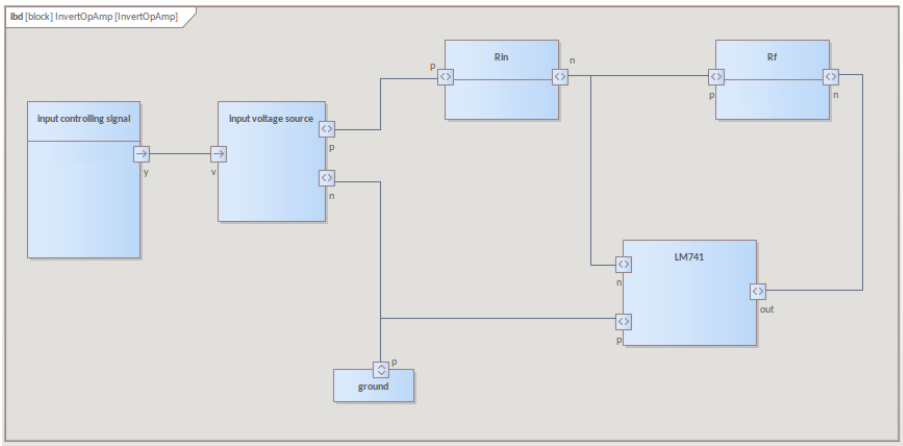
创建 SysML模型

这张表显示了我们如何构建一个完整的完成模型来表示电路，从最低级别的类型开始，一步一步地构建模型。

部件	行动
块	在 SysML 中，使用 SysPhS，电路和每个组件类型都表示为块。 在块定义图 (BDD) 中，创建电路组件块。该电路有五个部分：运算放大器、信号源、信号电压转换器、地和电阻。这些部分属于不同的类型，具有不同的行为。 关键模块，包括运算放大器、接地和电阻器，可作为预定义模块使用部件元件模式进行访问。但是，为了清楚起见，我们将运行开始运行它们的创建。 为每个部件类型创建一个部件块。电路内部块定义 (IBD) 的各个部分将通过端口连接，端口代表电气引脚。这些在 BDD 中定义。电阻器有一个正极和一个负极引脚。地只有一个引脚，它是正极。电流（电荷）通过引脚传输。这两个端口分别命名为 "p"（正）和 "n"（负），它们的类型为 ChargeFlowElement。 此图显示了 BDD，其中块定义了所用组件的类型。这包括信号源sine 、信号电压转换器、接地、电阻型和运算放大器型。  注记：这些模块是使用 Modelica Blocks 或 Simulink Blocks 从工具箱创建的。您可以在两个工具中键入每个块。有关详细信息，请参阅将模块设置为 *Modelica* 和 *Simulink*帮助主题。
端口	用于端口的值类型在端口仿真库中预先定义。使用的两个关键量是 ChargeFlowElement 和 RealSignalInElement ValueTypes， 此图显示了块定义图中的 OpAmp块，OpAmp块上的端口设置为 ChargeFlowElement ValueType，并且在浏览器窗口中引用了它。  公共类型包的导入连接器显示了如何设置这些。更多关于包导入（圈出）的信息，请参见引用*SysPhS*仿真库 帮助主题。
Phs 常数	电阻器和Sine模块都在 MATLAB 和 Modelica 中各自的组件中定义了属性。

	让我们以电阻为例。对于这些组件，在它们各自的工具中，我们需要设置一个以欧姆为单位的电阻值。对于大多数电路，可能有多个电阻器部件源自此电阻器块，在 IBD 或参数图中建模为部件。因此，它们的值（以欧姆为单位）将在 IBD、参数图或可能在模拟数据集中设置。 要设置定义电阻的属性，我们需要创建一个 Phs 变量 r'''。初始值未设置。  **注记**：SysPhS模式包括一个已构建的电阻类型块。 要在每个单独的组件 (元件) 中设置电阻值，请参阅此表中的*Initial Values*行，其中显示了使用端口和部件常量设置的块。 
内部结构	对于内部结构，我们创建一个带有子 IBD 图的块。 • 为反相运算放大器电路创建一个块 • 在此下使用上下文菜单选项创建一个内部块图表（IBD）： 创建新子图表|内部块定义图  • 双击打开 IBD • 将图表设置为仅显示端口的名称，使用： - F5 |元素|元素外观 • 取消设置这两个选项： - 显示构造型 - 显示属性类型 这将只显示端口和类型 在 IBD 上，创建零件并连接它们： • 从浏览器中，将 Blocks 作为 Parts（属性）拖到 IBD 上

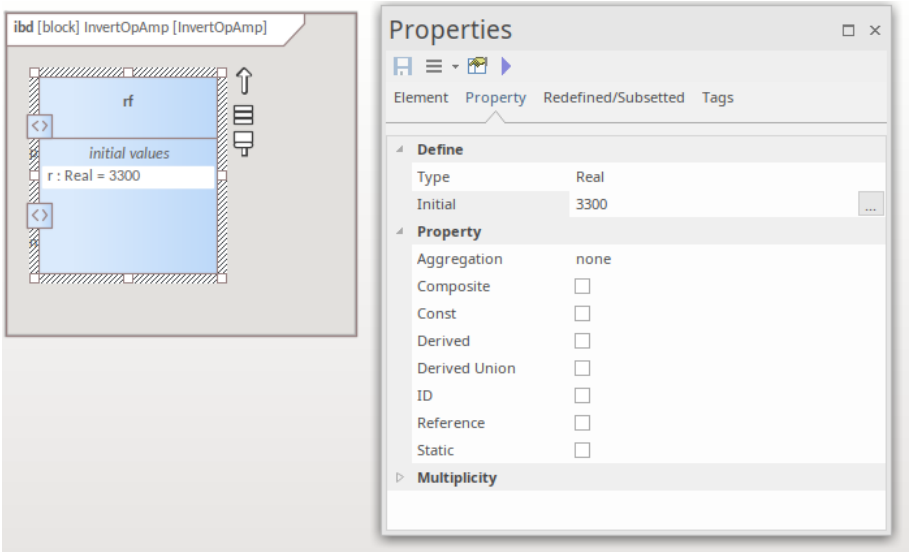
	• 要查看隔间中的部件，请将它们从图表中删除 注记: 将粘贴选项 '结构元素' 设置为 'ALL'  • 否则，在 属性”按钮的 特征视图”和 属性”选项卡的 交互”选项卡中单击 全部”选项卡  • 为源信号、转换器、2 个电阻、运算放大器和接地添加属性，由相应的 Blocks 键入
绑定	到模型这些组件的接线： • 使用类型连接器的连接器设置端口之间的连接。



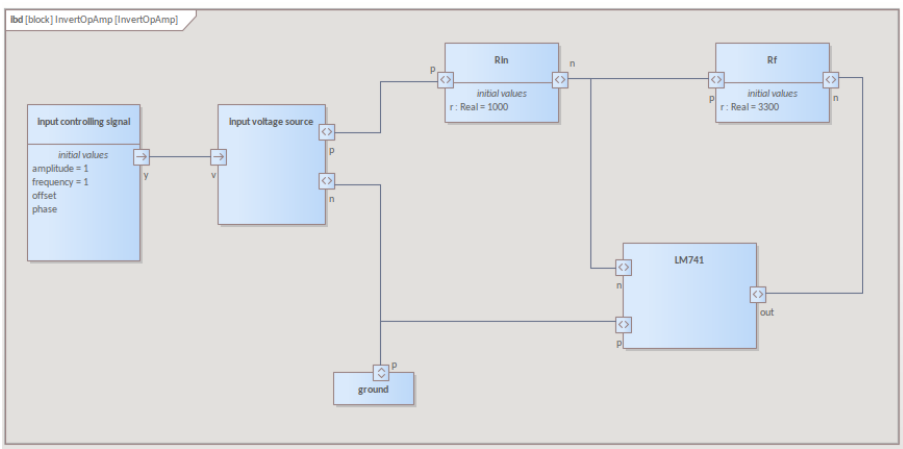
请注意，这遵循与原始电路图相同的结构，但每个组件的符号已替换为由我们定义的块键入的属性。

初始值

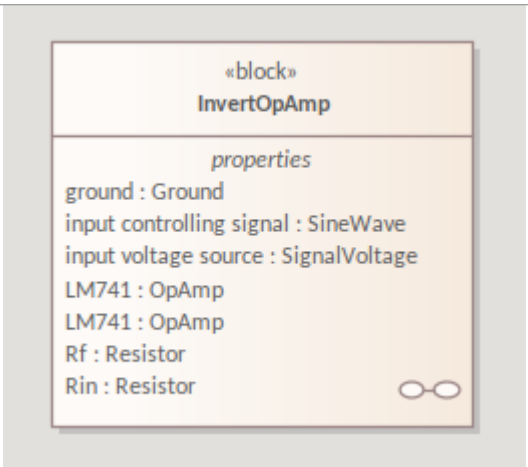
由于有两个电阻器，因此输入电阻器 (RIn) 和反馈电阻器 (Rf) 必须在 IBD 中创建为两个不同的（部分）属性，源自电阻器块，因为它们中的每一个都有不同的值。这些值必须在属性窗口、 属性”选项卡、 初始”字段中设置。



对于 InputControllingSignal，幅度和频率需要初始值。

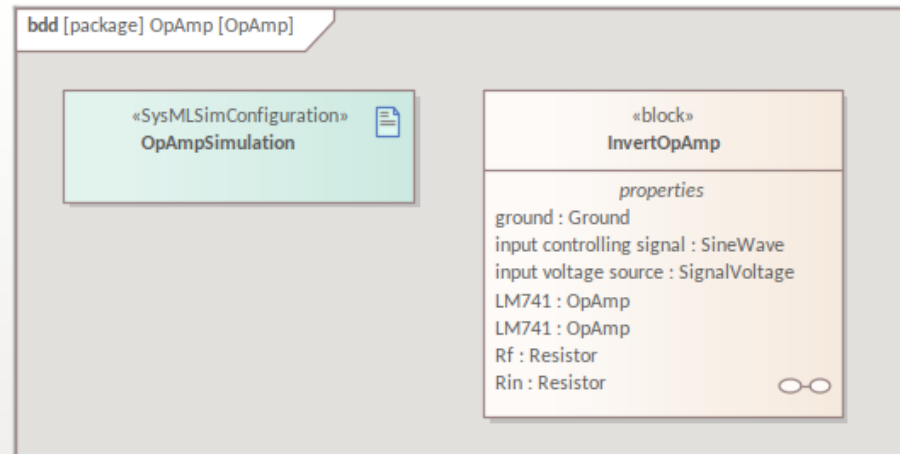
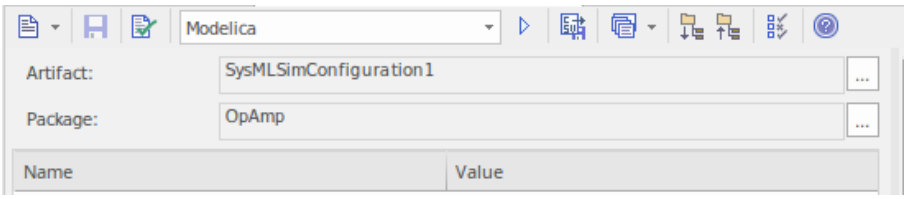


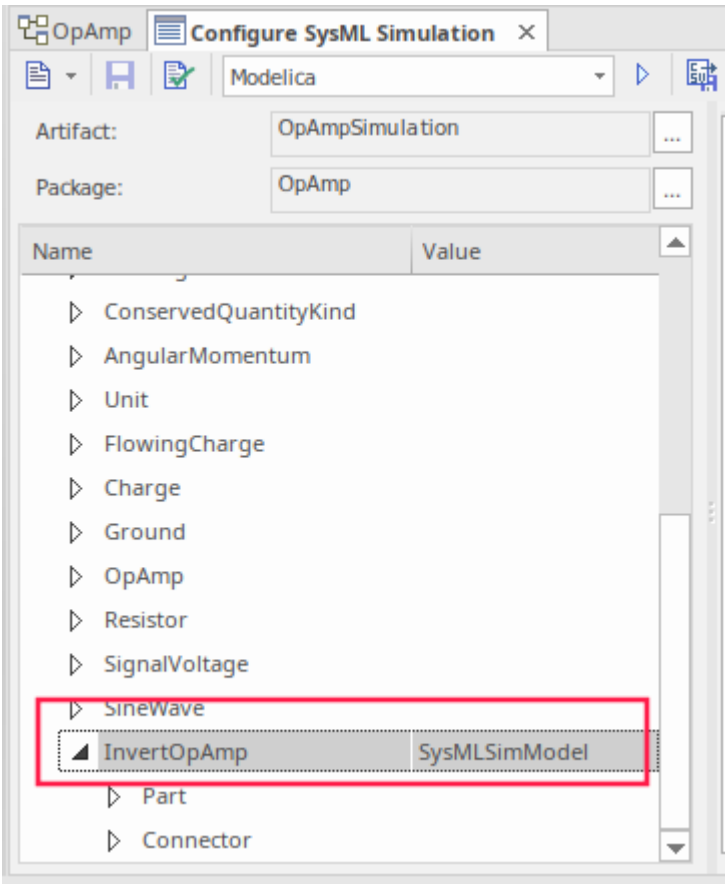
返回 BDD 后，您现在应该将现在块显示为：

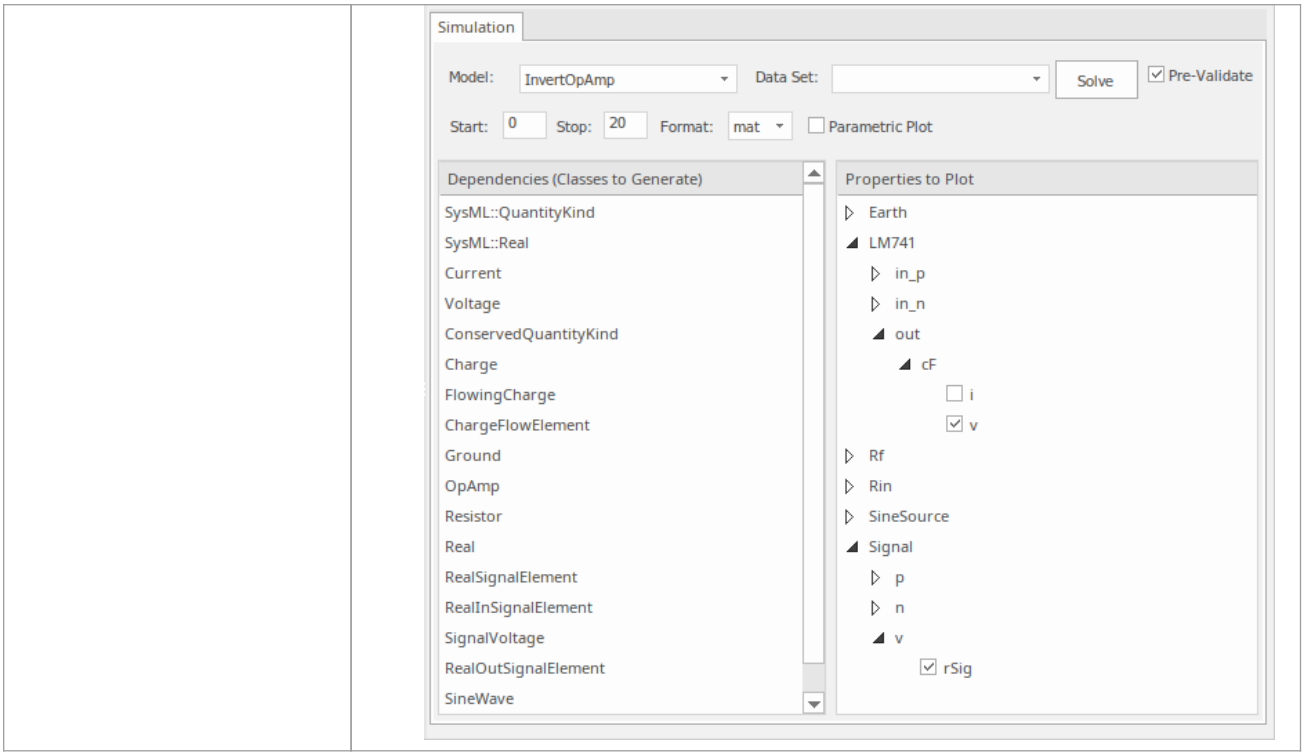
	

配置仿真行为

此表显示了 SysMLSim 配置的详细步骤。

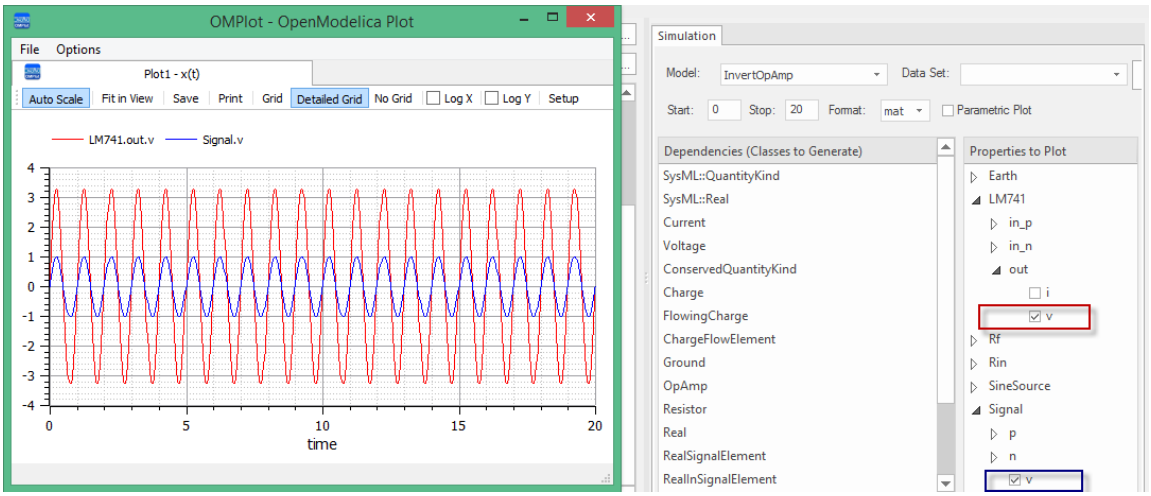
节	行动
创建一个工件	• 打开块定义图 • 点击图中空白处 • 按空格键 • 从 配置”中选择 工件”子菜单 这创建了一个新的配置工件 
设置包	• 在配置上工件 这将打开配置SysML配置窗口 • 在 包”字段中单击 [...] 按钮并选择包含 SysML 图的包： 

设置 Modelica 或 Simulink	在顶部下拉菜单中，您可以选择要使用的模拟工具： • 模型 • Simulink 有关这些设置的更多详细信息，请参阅配置*SysML*仿真帮助主题。
设置块为仿真	• 在左侧列表的 "名称"下，找到 "InvertOpAmp" • 在 "值"列中，单击下拉菜单并选择 "SysMLSimModel" 
选择要绘制的属性	您现在可以选择要绘制的属性： • 在右侧窗格中，选择要绘制的端口



运行仿真

在 仿真"选项卡上，单击求解按钮。此示例显示了在 Modelica 中生成的图：



在图例中，您可以看到绘制的属性和 '信号.v 属性，并将它们与在属性属性下选择的两个匹配。

在 Modelica 或 Simulink 中视图模型

要在外部应用程序、Modelica 或 Simulink 中查看生成的模型，请参阅查看帮助或 Simulink视图的模型帮助主题，以及调试生成代码中任何问题的提示。

数字电子仿真示例

在本例中，我们为一个简单的数字电子电路创建 SysPhS模型，然后使用模拟来预测和绘制该电路的行为。

此示例适用于 SysPhS 通用组件中未包含的组件，因此它贯穿了创建块以从头开始匹配外部组件的过程。有关演示这一点的网络研讨会，请参阅主题末尾的了解更多信息中的链接。

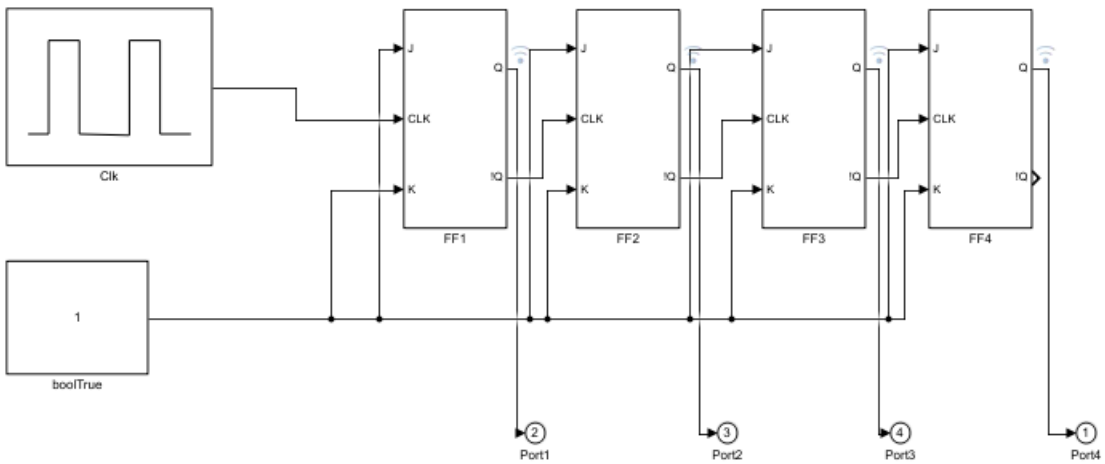
先决条件

运行此模拟需要：

- OpenModelica 或
- MATLAB 的 Simulink

图表- 数字分频器

我们要学习的数字电子电路模型如图所示，它使用标准电子电路符号。

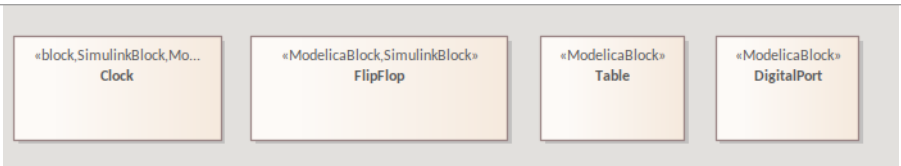


本例中的电路包括一个脉冲数字信号源、四个触发器和一个逻辑布尔真状态，构成一个简单的分频器电路。

创建 SysML模型

这张表显示了我们如何构建一个完整的完成模型来表示电路，从最低级别的类型开始，一步一步地构建模型。

部件	行动
块	在 SysML 中，使用 SysPhS，可以使用块来表示电路和每个组件类型。 首先在一个名为 数字模型”的包下创建一个块定义图（BDD）。 在 BDD 中，您将为电路创建一组组件，作为 SysML 块。该电路包含四种 Part-端口一个脉冲数字信号源、一个Flip-Flop、一个布尔真端口和一个逻辑布尔状态表示。这些部件属于不同的类型，具有不同的行为。 为每个部件类型创建一个部件块。电路内部块定义 (IBD) 的部件将通过端口连接，端口代表电气引脚。这些必须在 BDD 中定义。 此图显示了 BDD，其中块定义了所用组件的类型。



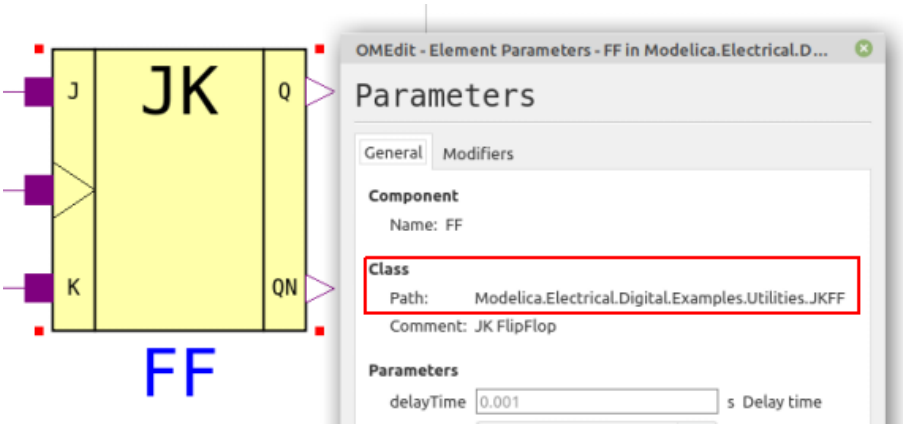
注记这些模块是使用 Modelica Blocks 或 Simulink Blocks 从工具箱创建的。您可以在这两个工具中键入这些块。

有关详细信息，请参阅将模块设置为 *Modelica* 和 *Simulink*帮助主题。

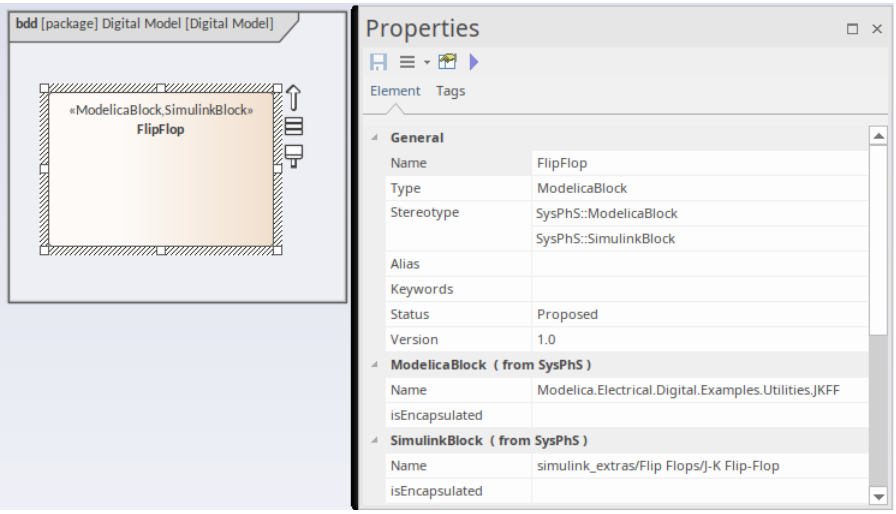
设置 Modelica 和 Simulink 路径

为了定义 Modelica 或 Simulink 特定的块，您需要在相应的应用程序中访问部件的路径，然后在块的属性中设置它。

例如，我们可以在 Modelica 中找到 Flip-Flop 组件。



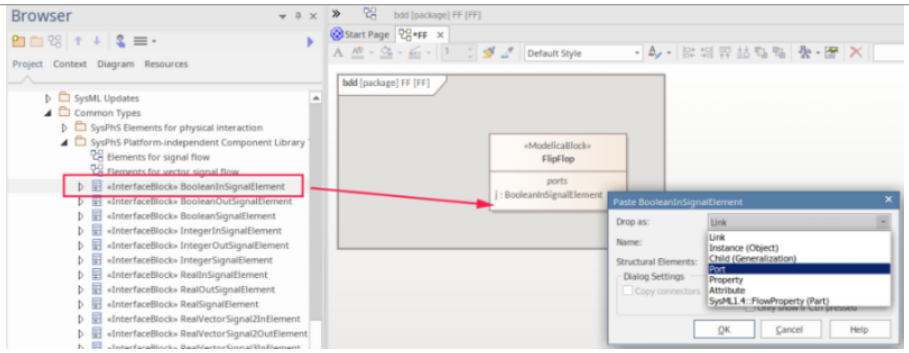
然后我们将其复制到块的属性中。



有关更多详细信息，请参阅创建 *Modelica* 特定模块和创建 *Simulink* 特定模块帮助主题。

端口

要在块上设置（在这种情况下为 Flip-端口时钟端口），您可以将 Modelica 或 Simulink PhS端口拖到块上。然后必须将该端口输入为端口。



注记：

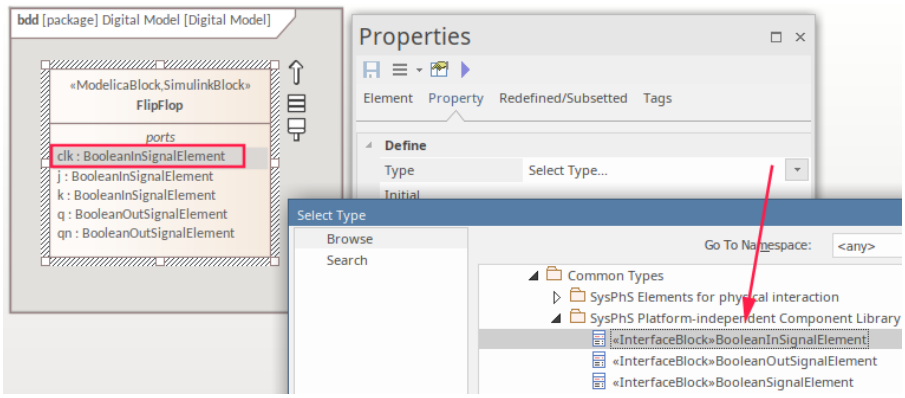
- 对于 Simulink，创建端口的端口很关键 - 请参阅*Simulink*端口排序 在创建 *Simulink* 和 *Simscape* 特定模块帮助主题中
- 这些端口可以通过为其他端口类型添加构造型来设置为 Modelica 和 Simulink

公共类型

作为所有 SysPhS 模型的启动器，您必须确保使用包导入连接器将 SysPhS 常用类型加载到存储库中并在新模型中引用。有关更多信息，请参阅引用 *SysPhS*仿真库 帮助主题。

用于端口的值类型在端口仿真库中预先定义。使用的两种键类型是 BooleanInSignal 和 BooleanOutSignal ValueTypes。

此图显示块定义图中的触发器块，时钟设置为端口ValueType，并且在浏览器窗口中引用。



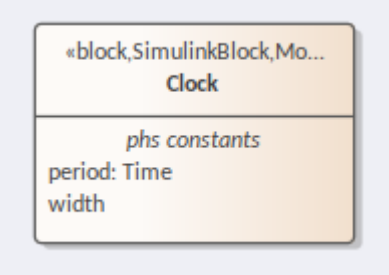
Phs 常数

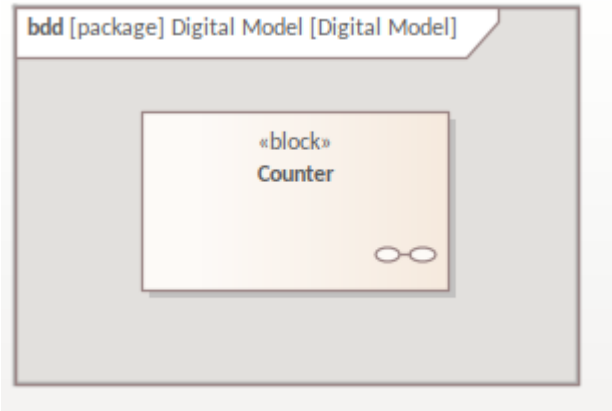
Clock 和 Boolean-true 模块都在 MATLAB 和 Modelica 的各自组件中定义了属性。

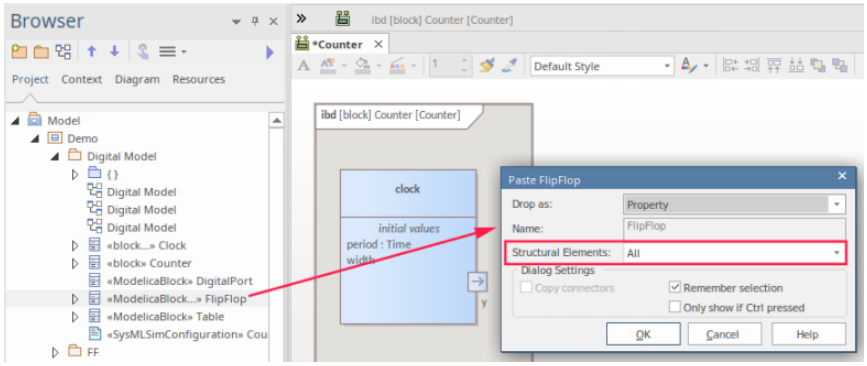
让我们以时钟为例。对于这种组件类型，在 Simulink 和 Modelica 中，我们都需要为每个脉冲的周期和每个脉冲的宽度设置一个值。必须设置和键入这些属性。属性的值将在 IBD、参数图或可能在模拟数据集中设置。

设置定义周期的属性：

- 将 PhsConstant 从 SysPhS 工具箱拖到时钟上
- 从图中删除元素以在其隔间中显示
- 在属性窗口的属性选项卡中，选择 类型"字段
- 单击 [...] 并选择 时间"作为类型

<table>
<tr><td></td><td> 

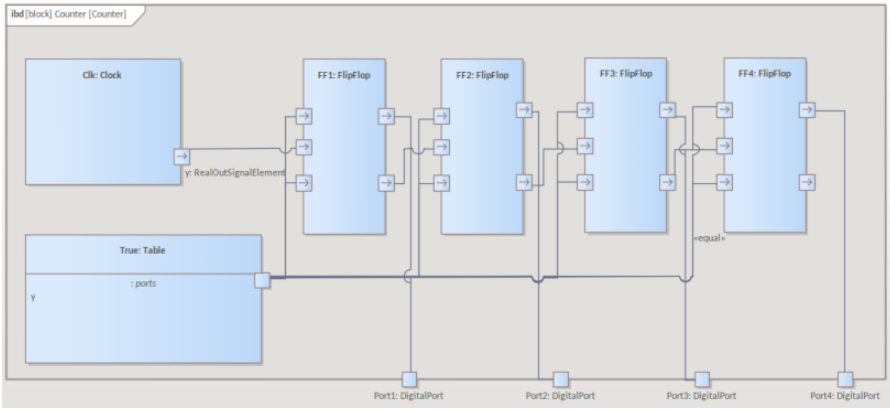
要设置单个组件（部件）中的值，请参阅此表中的Initial Values行。</td></tr>
<tr><td>内部结构——电路</td><td>对于内部结构，我们创建一个带有子 IBD 图的块。
• 为触发器电路创建块
• 在此块下使用上下文菜单选项创建一个内部块定义 (IBD) 图
创建新子图表|内部块定义图


• 双击打开 IBD
• 将图表设置为仅显示端口的名称，使用：
- F5 |元素|元素外观
• 然后取消设置这两个选项：
- 显示构造型
- 显示属性类型
这将只显示端口和类型
在 IBD 上，您创建零件并连接它们：
• 从浏览器窗口中，将 Blocks 作为 Parts（属性）拖到 IBD 上
• 要查看隔间中的这些部件，请将它们从图表中删除
注记: 将 '结构元素' 的粘贴选项设置为 'ALL'


• 添加一个时钟、四个触发器和一个表
• 在 Counter IBD 的边界添加四个数字端口</td></tr>
<tr><td>绑定</td><td>到模型这些组件的接线：</td></tr>
</table>

<table>
<tr><td></td><td>

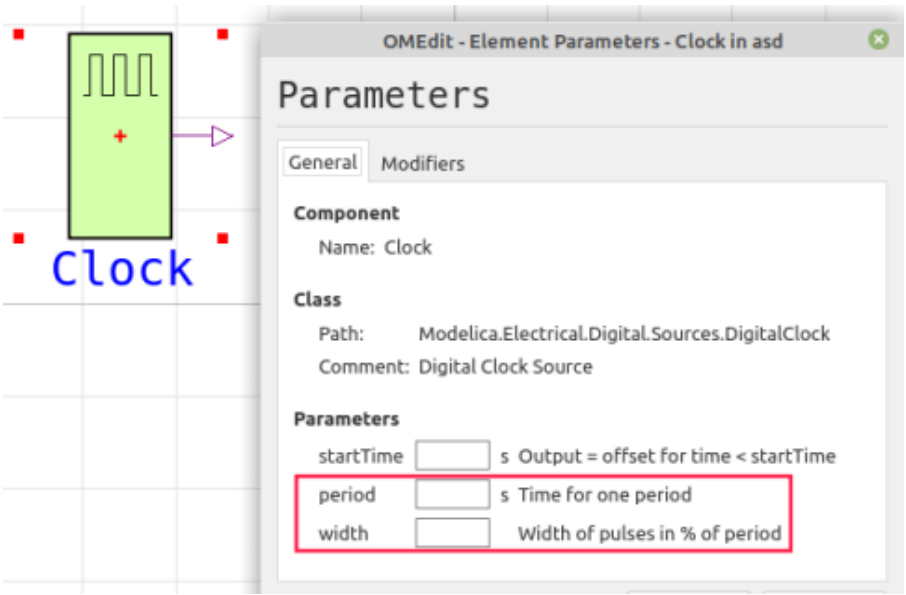
- 使用 连接器”类型的连接器在端口之间创建连接



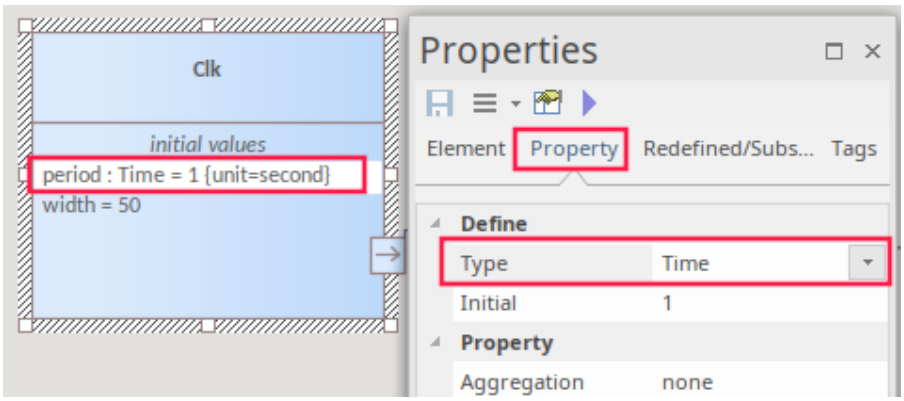
请注意，这遵循与原始电路图相同的结构，但每个组件的符号已替换为由我们定义的块键入的属性。

</td></tr>
<tr><td>初始值</td><td>

数字脉冲源是 Modelica 和 Simulink 中的 DigitalClock 组件。这需要两个参数 - 'Period' 和 'Width'，如 Modelica 编辑器中所示。

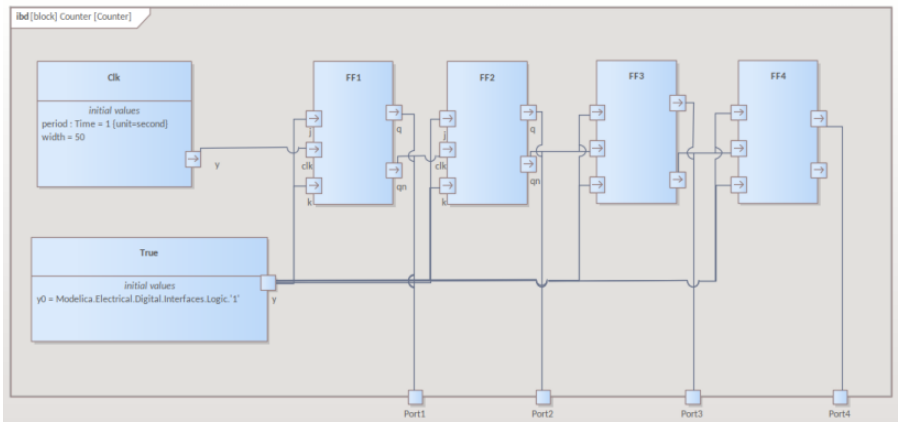


这些参数的值必须在 IBD部件的 Clk”中设置，在属性窗口 属性”选项卡的 初始”字段中

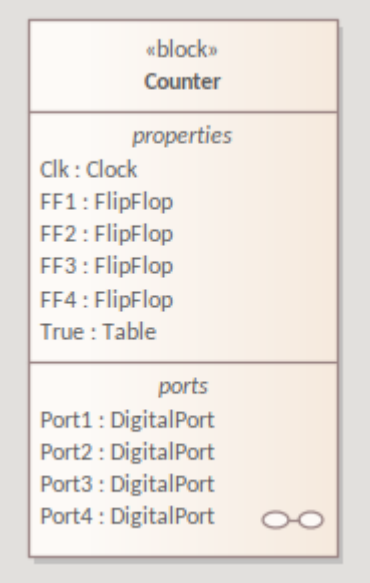


J & K端口需要一个固定的逻辑 True ”状态。这是通过使用初始值设置为 “true”的库表定义的，如此处针对 Modelica 所示。

</td></tr>
</table>



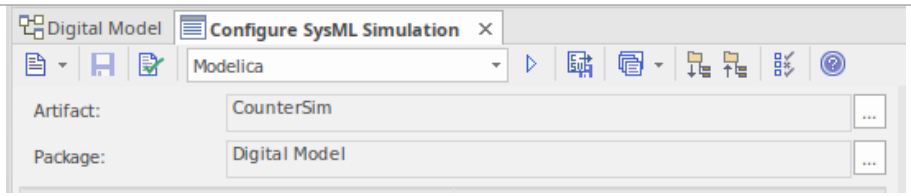
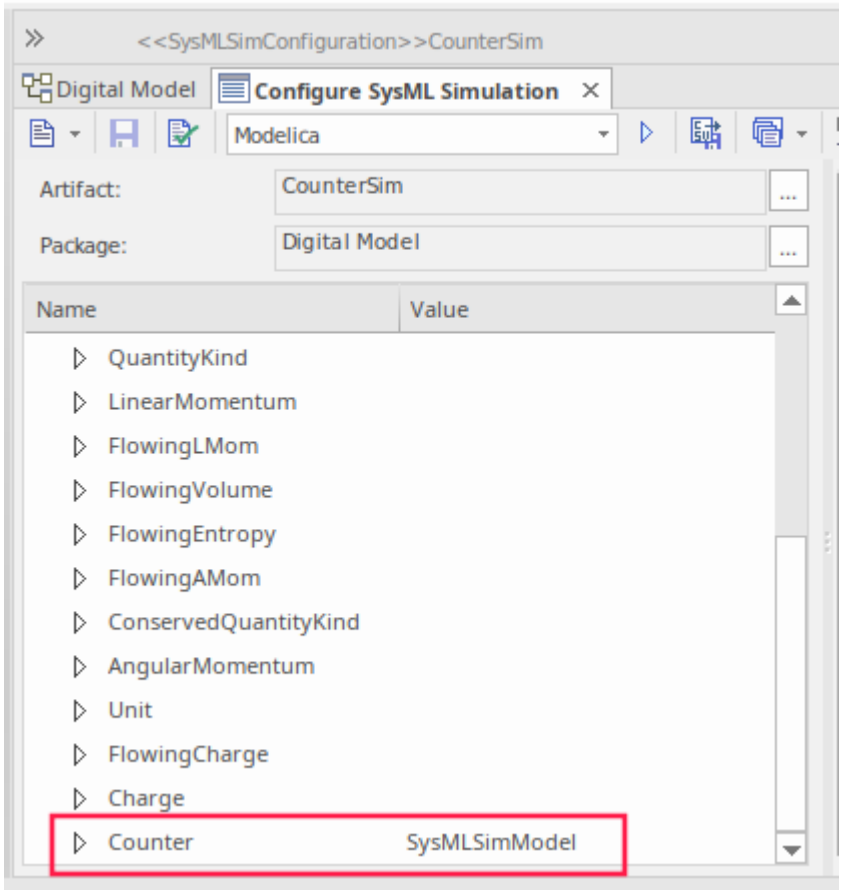
返回 BDD 后，您现在应该将 Counter块显示为：

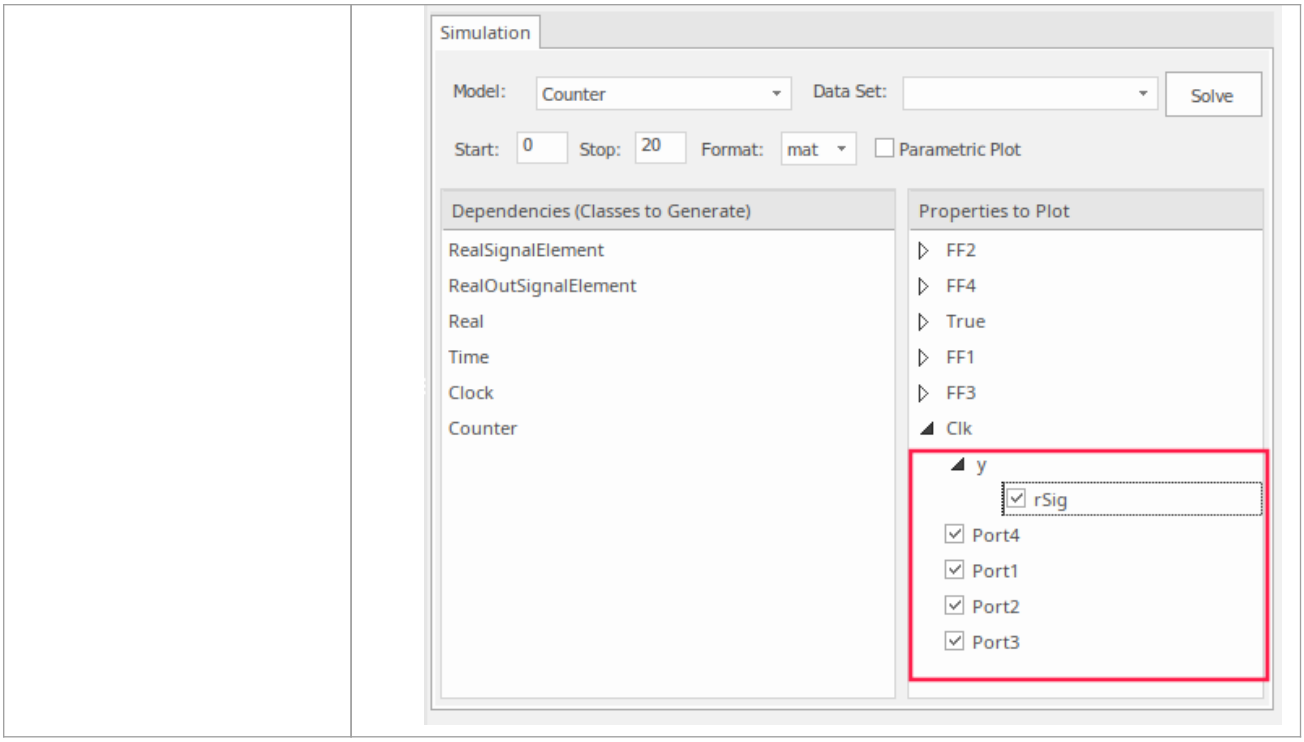


配置仿真行为

此表显示了 SysMLSim 配置的详细步骤。

节	行动
创建一个工件	• 打开块定义图 • 点击图中空白处 • 按空格键 • 从 配置”中选择 工件”子菜单 这创建了一个新的配置工件
设置包	• 在配置上工件 这将打开配置SysML配置窗口 • 在 包”字段中，单击 [...] 按钮并选择包含 SysML 图的包

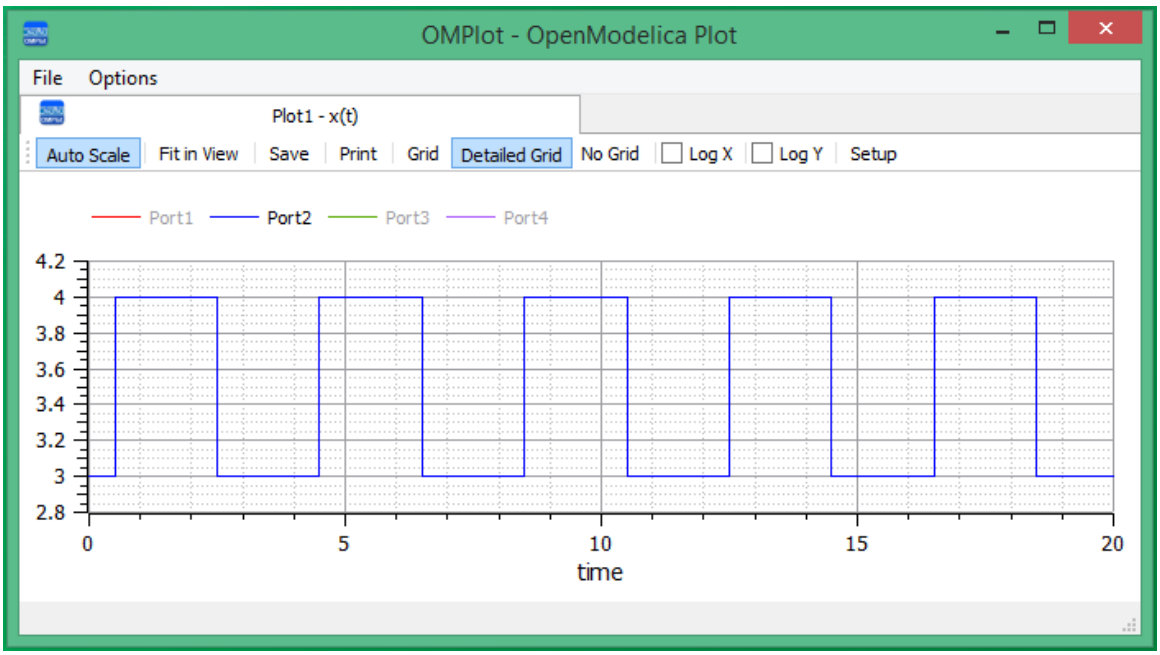
	
设置 Modelica 或 Simulink	在顶部下拉列表中选择要使用的模拟工具： • 模型 • Simulink 有关这些设置的更多详细信息，请参阅配置*SysML*仿真帮助主题。
设置块为仿真	• 在左侧列表中的 块”下，找到 InvertOpAmp” • 在 值”列中，单击下拉菜单并选择 SysMLSimModel” 
选择要绘制的属性	您现在可以选择要绘制的属性： • 在右侧窗格中，选择要绘制的端口



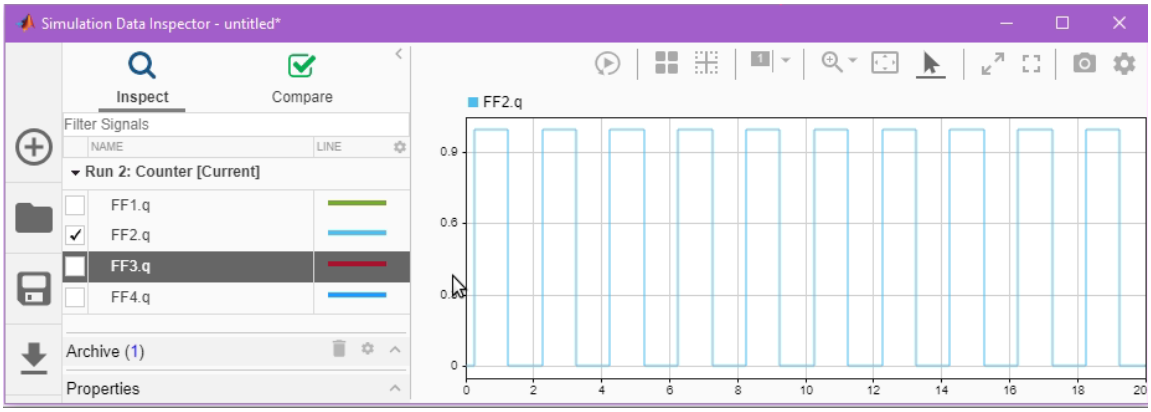
运行仿真

在 仿真"页面上，单击求解按钮。这显示了在以下位置生成的图的示例：

模型



Simulink



在图例中，您可以看到端口2 已被选中，而另一个端口已被取消选中，以显示一个简单的*Plot*

在 Modelica 或 Simulink 中视图模型

要在外部应用程序、Modelica 或 Simulink 中查看生成的模型，请参阅查看生成的模型帮助主题。另请参阅*SysPhS*调试提示帮助帮助中有关调试生成代码中任何问题的提示。

加湿器示例

在本例中，我们通过创建室内加湿器的 SysPhS模型来说明信号流和状态机的使用。本例中的信号反映了物理量，但不同于物理物质意义上的物理相互作用与流速和电位。

完成加湿器示例是一组复杂的 SysML 模型，但它包含需要使用 MATLAB 的状态流进行仿真的部分。我们将使用示例的这一部分来完成 SysML状态机的使用，并查看它是如何在 MATLAB 中使用状态流以及在运行中配置和运行的。

先决条件

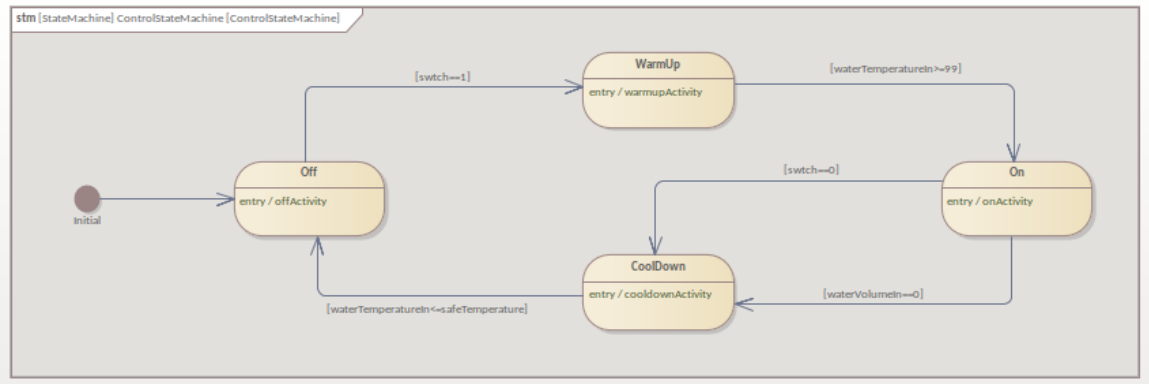
运行此模拟需要：

- OpenModelica 或
- MATLAB 的 Simulink 和 StateFlow

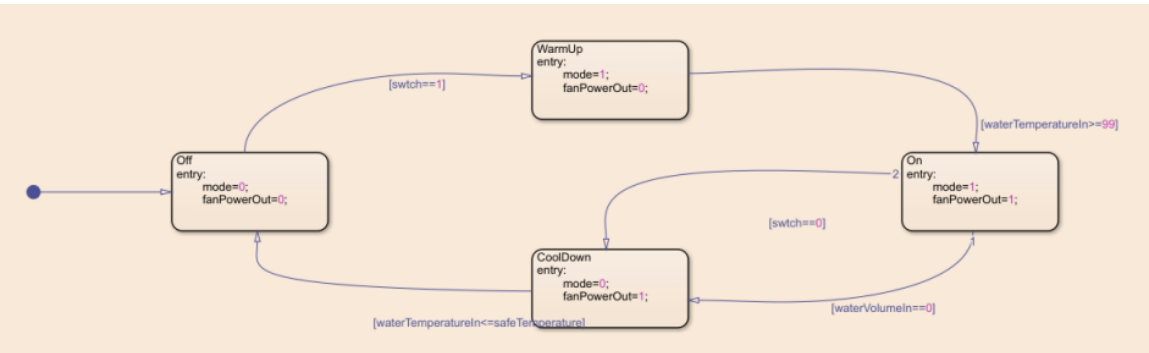
概述

完成模型中提供了完成模拟WebEA 。请参阅本主题末尾了解更多信息下的链接。

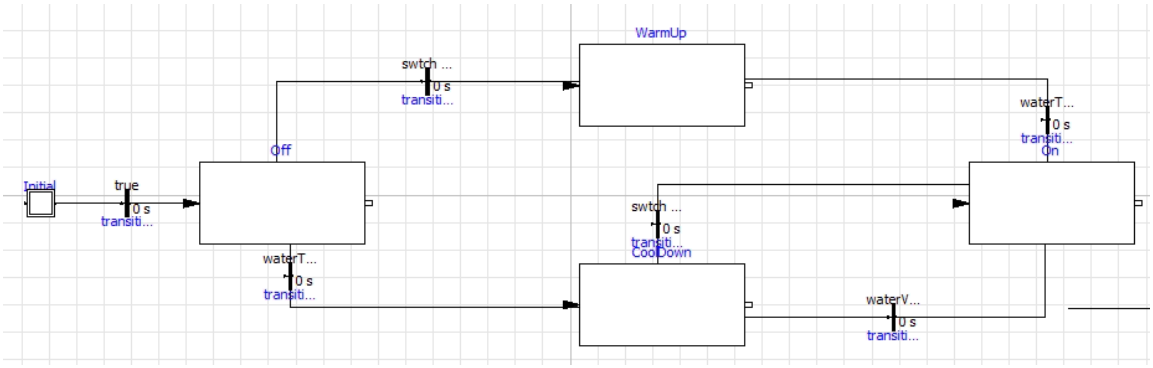
SysML状态机图为：



对于 MATLAB，生成的状态流图包括：



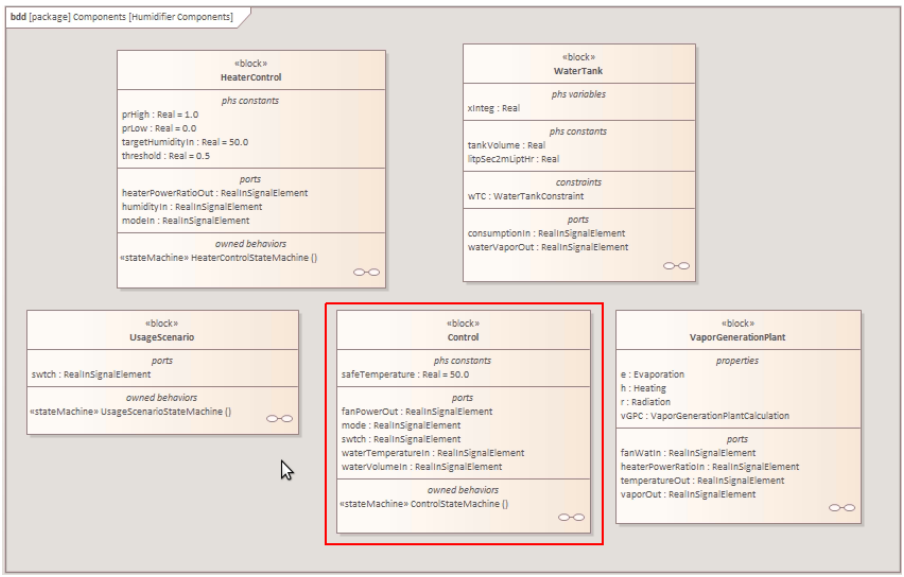
对于 Modelica，生成的图表如下所示：

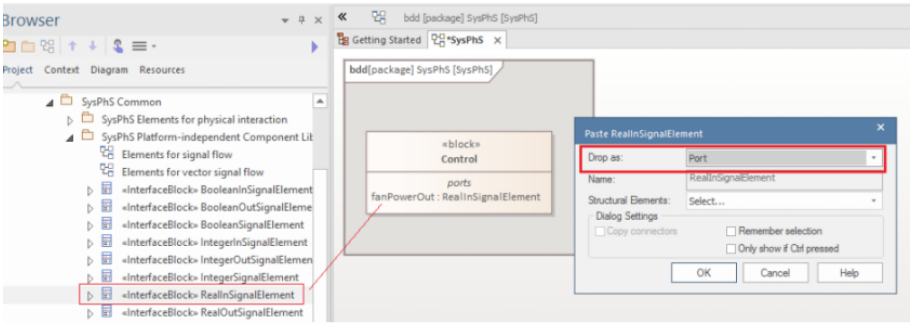
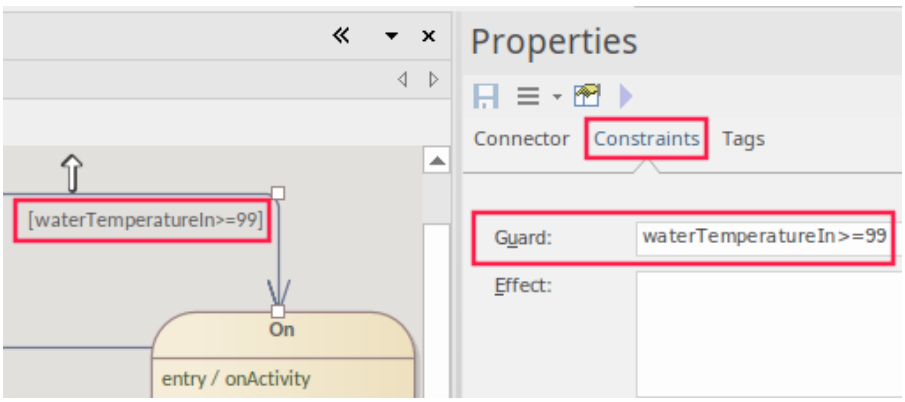


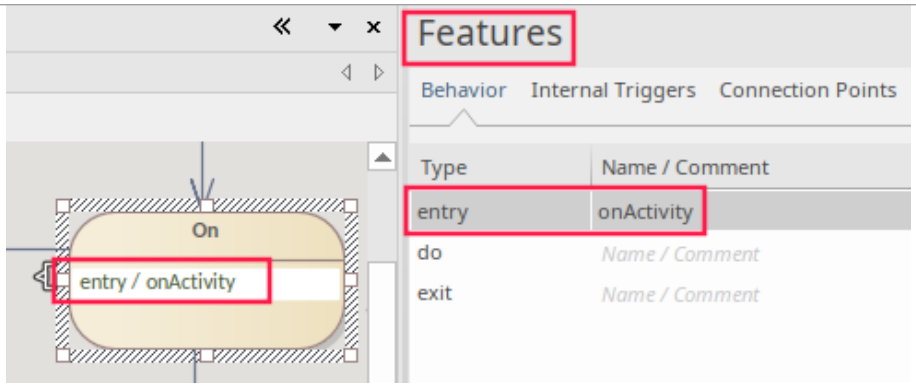
创建 SysML状态机模型

此表显示了我们如何构建 SysML状态机模型来表示开关状态。

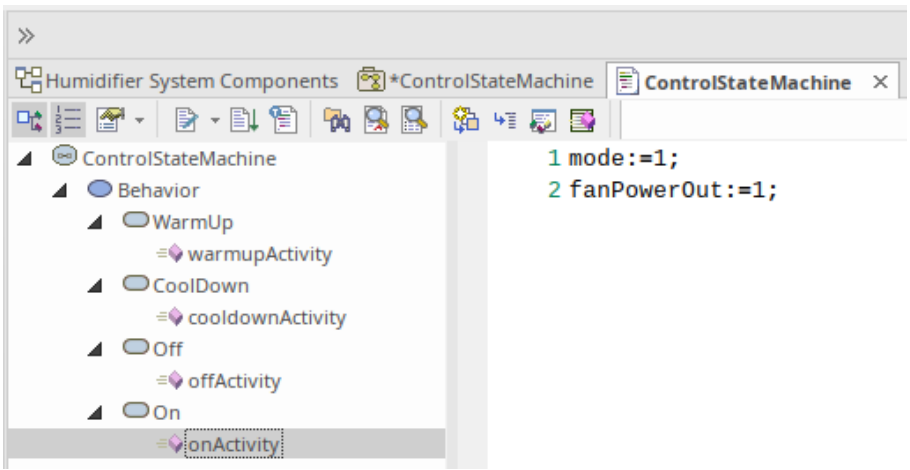
注记：完整的 SysML状态机特征仅在 MATLAB 的状态流和 Modelica 中得到部分支持。因此，为了在这些产品中创建用于仿真的状态图， Enterprise Architect的状态机模型中使用的连接器类型和对象类型需要仅包含状态流和 Modelica 中支持的相应特征。

节	描述
块	在 SysML 中，使用 SysPhS，核心加湿器组件由块表示。对于这个例子，我们正在创建状态机模拟，聚焦在 控件”块上。块定义图 (BDD) 称为 加湿器组件”，其中包括 控件”块，它只是定义加湿器的 BDD 列表中的一个。  '控件'块下有状态机'ControlStateMachine'作为复合图；这在本主题开头的概述部分的第一个图表中进行了说明。 控件”块是定义加湿器控制的端口和端口常数的位置。端口的类型为端口，即信号流。
公共类型	作为所有 SysPhS 模型的启动器，您需要确保 SysPhS 常用类型已加载到存储库中，并使用包'导入'连接器在新模型中引用。有关更多信息，请参阅参考 *SysPhS* 库 帮助话题。
端口	用于端口的值类型在端口仿真库中预先定义。使用的密钥类型是 RealInSignal ValueType (RealInSignalElement)。这可以作为端口拖到块上，然后重命名。 作为在端口上设置的示例，在本例中为*fanPowerOut*端口块，您将一个

	RealInSignalElement 拖动到一个 RealInSignalElement 上并将其定义为一个端口块。  **注记：** • 这些端口可以通过为其他端口类型添加构造型来设置为 Modelica 和 Simulink
创建状态机	将状态机定义为块的复合子图： • 从块上的上下文菜单中选择 新子图表>状态机”，然后 • 从上下文上的块中选择 新子图表>选择复合”，然后选择刚刚创建的图表
创建状态和转换	在子状态机图中，添加初始元素、四个状态和一个终点元素。然后适当地命名它们。 注记：选中图表时按空格键，可以从上下文菜单中选择一个初始object ，然后使用快速链接器添加剩余的对象。
转移 警卫	状态之间的转换包含它们的守卫中的条件。这些条件在属性窗口的 约束”选项卡中定义。 使用的变量是 控件”块的控件常数和端口的。请参阅前面的*Blocks*行中的图像。 
行为	每个状态都包含一个定义风扇模式和状态的脚本。这些脚本在 Entry”操作中设置。



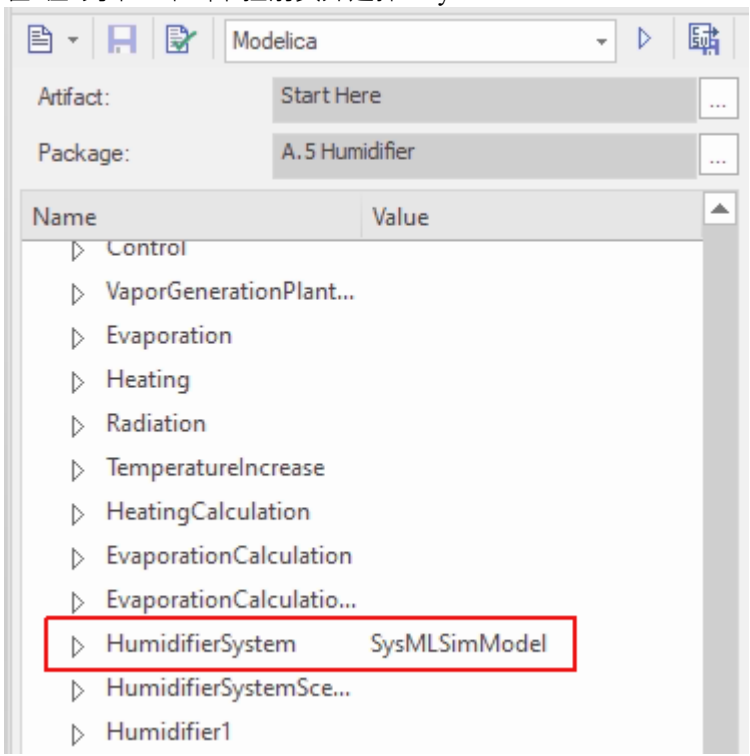
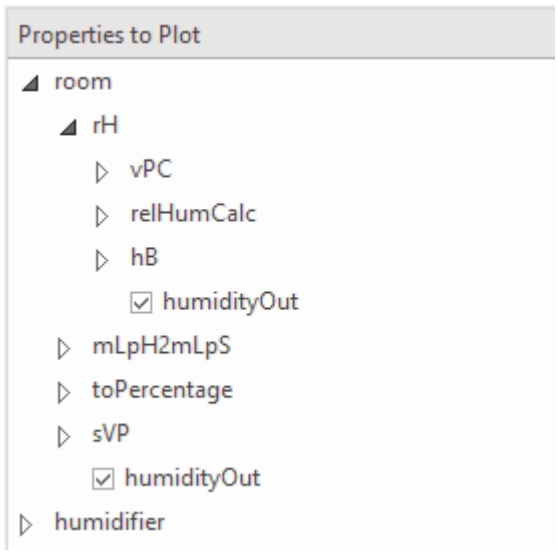
要为此编辑脚本，请选择根状态机并按 Alt+7。



配置仿真行为

此表显示了 SysMLSim 配置的详细步骤。

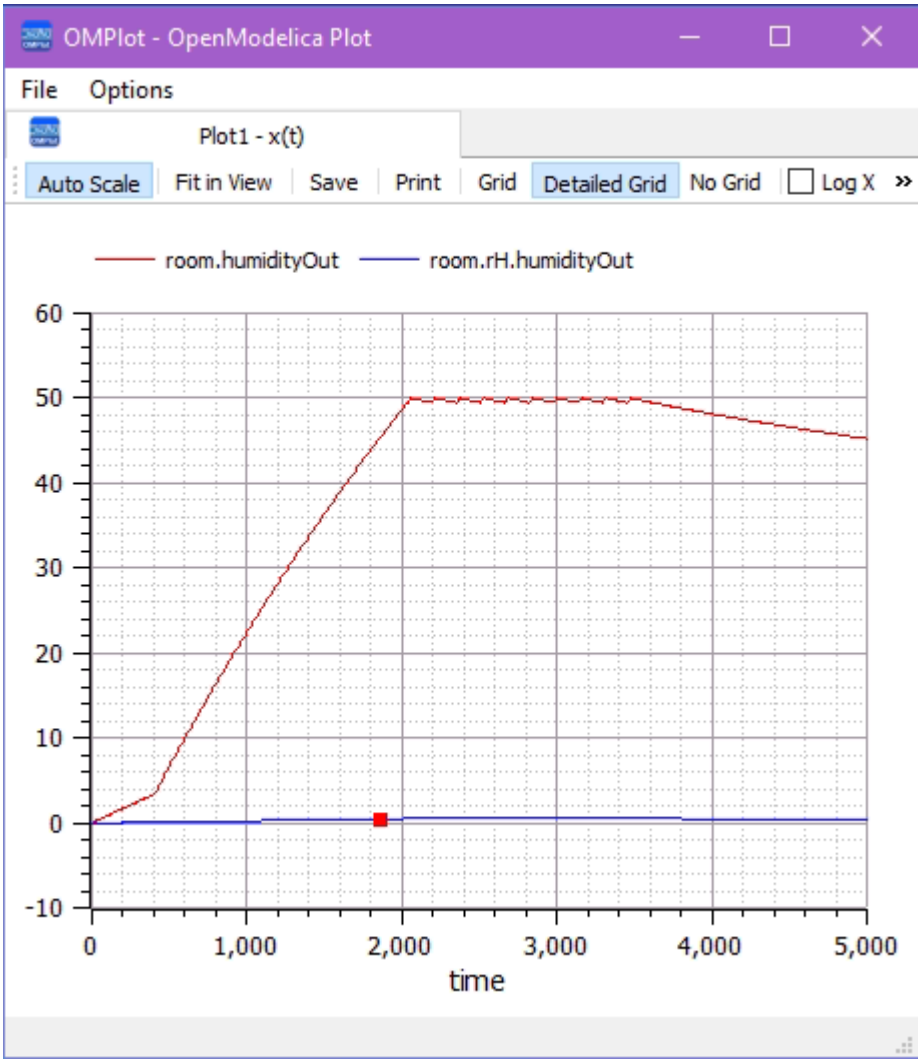
节	行动
创建一个工件	• 打开块定义图 • 点击图中空白处 • 按空格键 • 从 配置"中选择 工件"子菜单 这创建了一个新的配置工件
设置包	• 在配置上工件 这将打开配置SysML仿真窗口 • 在 包"字段中，单击 [...] 按钮并选择包含 SysML 图的包
设置 Modelica 或 Simulink	在顶部下拉列表中，选择要使用的模拟工具： • 模型 • Simulink 有关这些设置的更多详细信息，请参阅配置*SysML*仿真帮助主题。
设置块为仿真	• 在左侧列表中，在 名称"列的 块"下，找到 HumidifierSystem"

	• 在 值”列中，单击下拉箭头并选择“ SysMLSimModel” 
选择要绘制的属性	您现在可以选择要绘制的属性： • 在右侧窗格中，选择要绘制的端口 

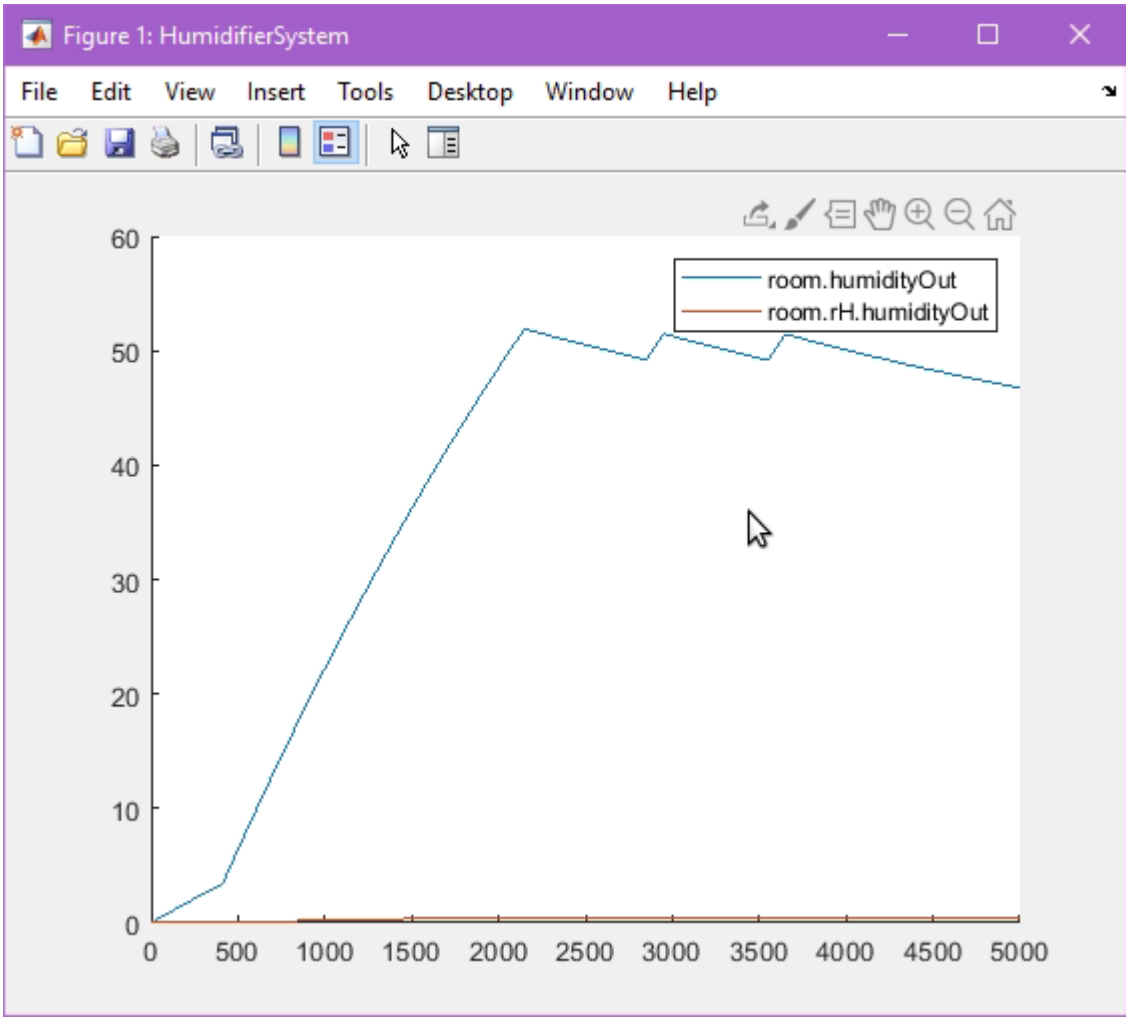
运行仿真

在 仿真”页面中，单击求解按钮。这显示了在以下位置生成的图的示例：

模型


OMPlot - OpenModelica Plot
File
Options
Plot1 - x(t)
Auto Scale
Fit in View
Save
Print
Grid
Detailed Grid
No Grid
Log X
room.humidityOut
room.rH.humidityOut
60
50
40
30
20
10
0
-10
0
1,000
2,000
3,000
4,000
5,000
time

Simulink



注记：由于 Simulink 模型的默认精度对于此特定示例来说不够，因此 Simulink 中的输出看起来不同。可以通过直接在 Simulink 中打开生成的文件来修改 Simulink 设置。

在 Modelica 或 Simulink 中视图模型

要在外部应用程序、Modelica 或 Simulink 中查看生成的模型，请参阅查看帮助或 Simulink视图的模型帮助主题，其中包含调试生成代码中任何问题的提示。

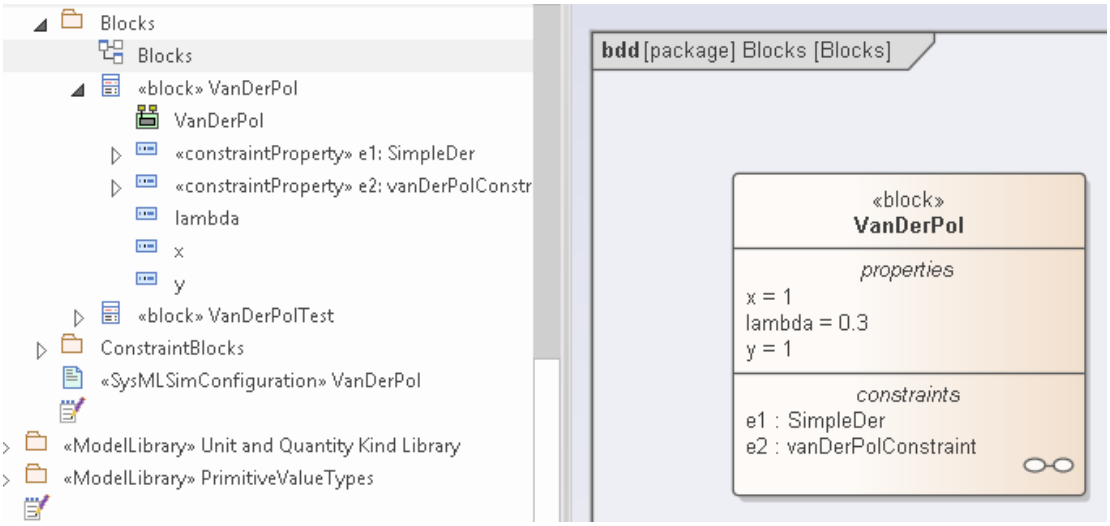
更新配置

在Enterprise Architect 15.2 之前的版本中，对于每个部件/属性，类型和选项是在配置 SysML仿真窗口中定义的，从配置配置窗口打开到工件。这种方法仍然有效且可用；但是，使用现在标准的最新特征，您现在可以在模型本身中定义仿真参数，这意味着您可以从相同的模型生成不同的仿真工具（Simulink 和 Modelica）并在元素级别进行调整。

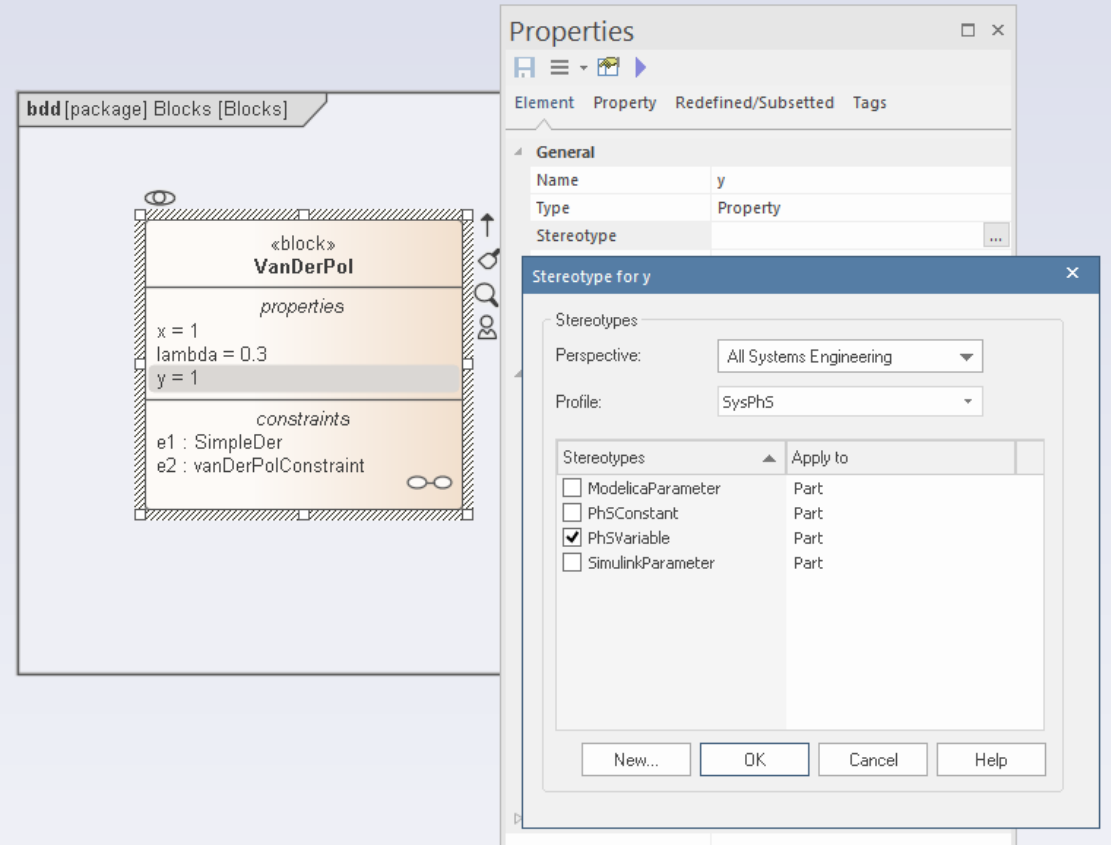
如果您更喜欢使用 SysPhS 选项，您可以更新最近的仿真配置以反映 SysPhS 标准的使用，并通过不同的工具支持仿真。

要更新现有配置：

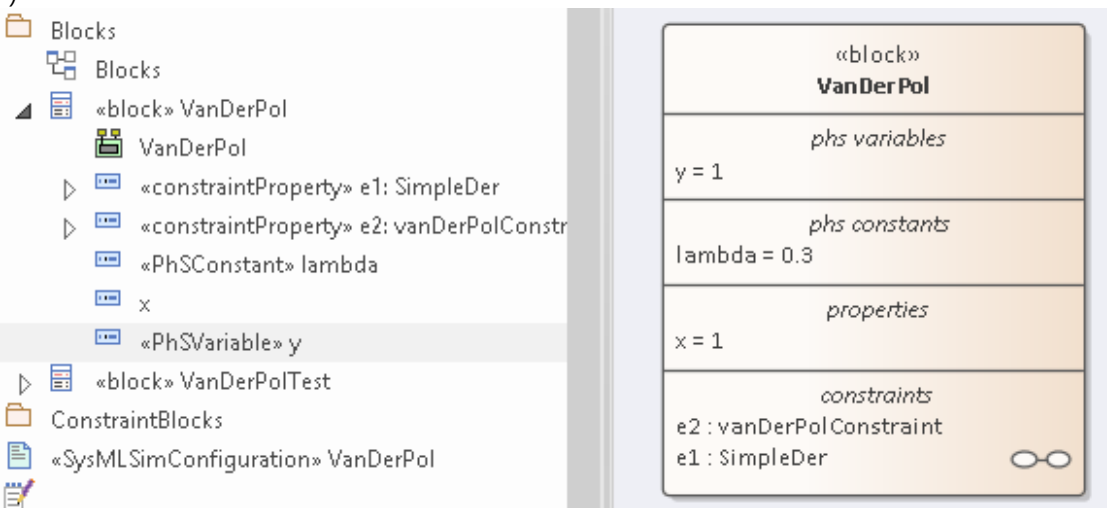
1. 确保您有对导入仿真库的包导入引用。有关更多详细信息，请参阅参考帮助仿真库帮助主题。
2. 打开包含模拟的块元素的图表。例如：



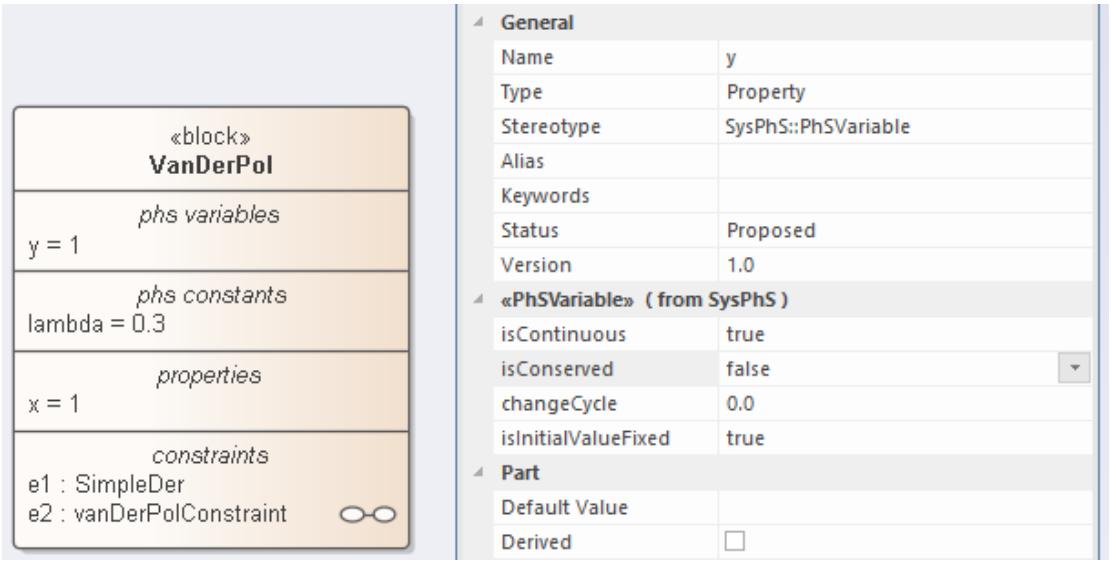
3. 注意浏览器窗口中的属性元素（在我们的插图中，*x* 、 *y*和*lambda* ）；无论是在此处、在图表中还是在属性窗口中，都无法查看每个属性的类型。
4. 在图中的元素中，单击一个属性，然后在属性窗口的 构造型"字段中，单击 ... 图标以显示 "<propertyname>的构造型"对话框。确保 蓝图"字段设置为 所有系统工程"，配置文件字段设置为 "SysPhS"。



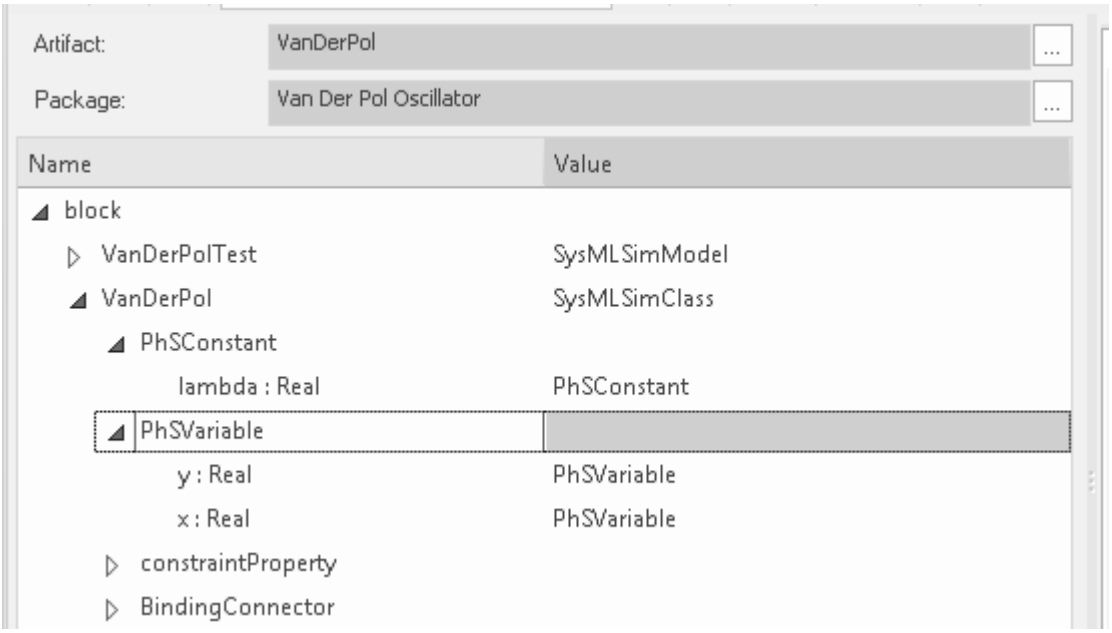
5. 单击相应属性类型（PhSVariable 或 PhSConstant）的复选框，然后单击确定按钮。
6. 注记如何将常量和变量分配给图表上自己的元素，以及如何在浏览器窗口中查看属性类型（作为构造型）。



7. 另请注意，在属性窗口中，您可以看到定义属性类型的属性类型，以及构造型标准中定义的该类型的选项（作为标记值）。



8. 最后，如果您仿真打开配置模拟配置工件查看该模拟配置，您将按类型识别属性。



SysML参数仿真

Enterprise Architect提供与 OpenModelica 和 MATLAB Simulink 的集成，以支持对 SysML模型在不同情况下的行为方式进行快速而可靠的评估。

OpenModelica 库是提供许多有用类型、函数和模型的综合资源。在Enterprise Architect中创建 SysML 模型时，您可以参考这些库中可用的资源。

Enterprise Architect的 MATLAB 集成通过 MATLAB API 进行连接，允许您的Enterprise Architect仿真和其他脚本根据任何可用的 MATLAB 函数和表达式的值进行操作。您可以通过 Solver类调用 MATLAB，或将您的模型导出到 MATLAB Simulink、Simscape 和/或状态流。

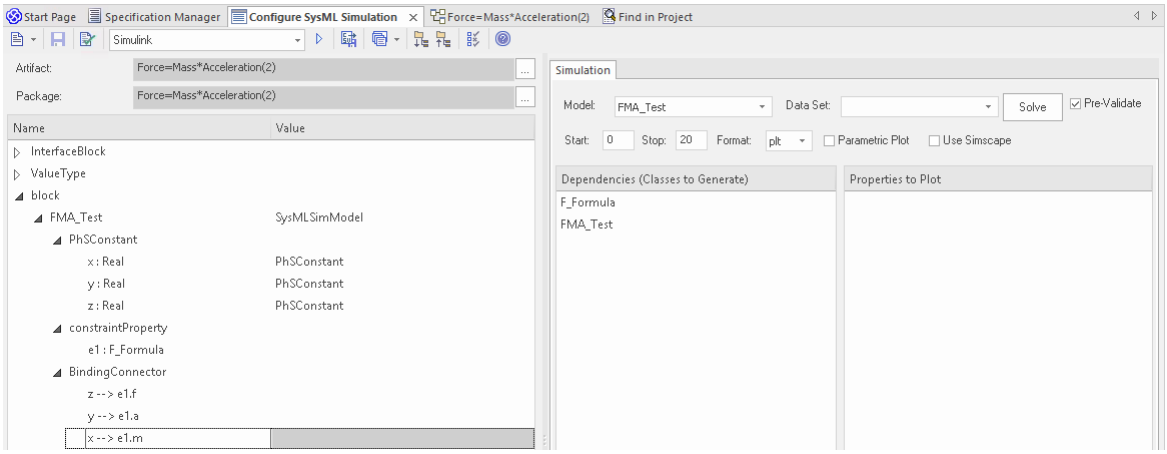
SysML仿真特征

这些部分描述了定义参数模型、使用附加信息对模型进行注释以驱动模拟以及运行模拟以生成结果图的过程。

部分	描述
SysML 参数模型简介	SysML 参数模型支持关键系统参数的工程分析，包括评估关键指标，如性能、可靠性和其他物理特性。这些模型通过捕获基于复杂数学关系的可执行约束，将需求模型与系统设计模型相结合。参数图是专门的内部块图，可帮助建模者将行为和结构模型与工程分析模型（例如性能、可靠性和质量属性）结合起来。 有关 SysML 参数模型概念的更多信息，请参阅 OMG SysML 官方网站及其链接资源。
创建参数模型	概述开发用于仿真的 SysML模型元素、在配置SysML仿真窗口中配置这些元素以及观察仿真结果。
工件适合	Enterprise Architect帮助您扩展 SysML 参数模型的实用性，方法是使用允许模拟模型的额外信息对它们进行注释。然后将生成的模型生成为可以使用 MATLAB Simulink 或 OpenModelica 求解（模拟）的模型。 模拟属性针对您的仿真模型进行工件。这保留了您的原始模型并支持针对单个 SysML模型配置的多个模拟。仿真工件在工件的工具箱上找到
用户接口	SysML 模拟的用户界面在配置模拟仿真主题中进行了描述。
使用数据集进行模型分析	使用仿真配置，SysML块可以有多个针对它定义的数据集。这允许在SysML模型的模拟上运行可重复的变化。
SysPhS 标准支持	*SysPhS* 标准是交互物理和辅助信号流仿真的 *SysML*扩展。它定义了在SysML模型和 Modelica模型或 Simulink/Simscape模型之间转换的标准方法，为共享仿真提供了一种更简单的基于模型的方法。请参阅*SysPhS* 标准支持帮助帮助。
例子	为了帮助您了解如何创建和模拟 SysML 参数模型，我们提供了三个示例来说明三个不同的领域。所有三个示例都使用 OpenModelica 库。 *SysML*仿真示例主题中描述了这些示例以及您可以从中学到的东西。

配置SysML仿真

配置SysML仿真窗口是一个界面，您可以通过该界面提供运行时参数以执行 SysML模型的模拟。工件中的模拟基于元素定义的模拟配置。

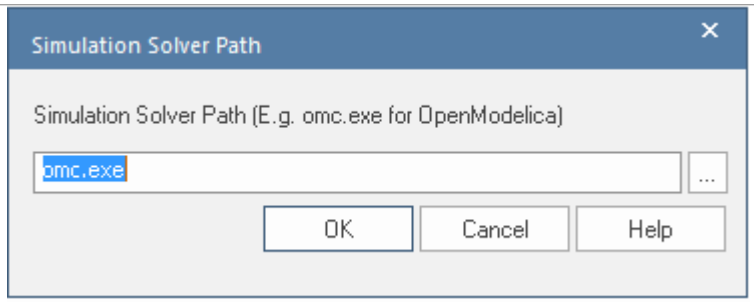


访问

功能区	仿真>系统行为> Modelica/Simulink >配置管理器
其它	双击具有工件构造型的属性。

工具栏选项

选项	描述
	单击下拉箭头并从以下选项中进行选择： • （如果尚未选择工件配置，则从一个工件配置中选择一个配置和负载） • 创建工件— 创建新的工件或选择并加载现有的配置 • 选择包- 选择一个包以扫描 SysML 元素以配置模拟 • Reload — 重新加载配置管理器并更改当前包 • 配置仿真求解器 - 显示 仿真求解器路径”对话框，您可以在其中键入或浏览要使用的求解器的路径。对于 MATLAB/Simulink，将自动检测路径，因此只有在自动检测存在问题或安装了多个版本的 MATLAB 时才需要更改此设置。

	
	单击此按钮将配置保存到当前工件。
	单击此图标现在专门针对现在配置验证模型。验证结果显示在系统输出窗口的 SysML仿真”选项卡中。您还可以选择一个选项以在执行每个模拟之前自动预验证模型。请参阅仿真标签表中的 预验证”选项。
	单击此图标可展开窗口 名称”列中层次结构中的每个项目。
	单击此图标可折叠窗口 名称”列中模型层次结构中所有展开的项目。
	单击此图标可显示可在模拟中抑制的object类型列表。单击要抑制的每个object的复选框，或单击 全部”按钮以选择要抑制的所有项目。 您还可以使用 选项”列顶部的过滤器栏来仅显示名称中具有指定字母或文本string的项目。
	单击下拉箭头并选择正在运行仿真的应用程序，例如运行或 Simulink。
	单击此按钮可生成、编译和运行当前配置，并显示结果。
	仿真后，以 plt、mat 或 csv 格式生成结果文件。也就是说，使用文件名： • ModelName_res.mat（OpenModelica 的默认值） • ModelName_res.plt 或 • 模型名称_res.csv 单击此按钮指定Enterprise Architect将复制结果文件的目录。
	单击此按钮可从以下选项中进行选择： • 运行Last Code - 执行最近生成的代码 • 生成代码——生成编译或运行代码 • Open仿真— 打开将生成 OpenModelica 或 Simulink 代码的目录 • 编辑模板——使用代码模板编辑器自定义为 OpenModelica 或 Simulink 生成的代码

仿真工件模型

字段	行动

工件	单击…并选择现有的工件图标，然后创建一个新的工件。
包	如果您有一个现有的工件配置模型包，该字段指定了与该属性相关联的默认工件。 否则，单击…图标并浏览并选择包含 SysML模型的包以进行仿真配置。您必须在选择包之前指定（或创建）工件。

包对象

本表讨论了将在配置SysML仿真窗口的 名称”列下列出的 SysML模型中的object类型，这些对象将在模拟中进行处理。每个object类型展开以列出该类型的命名对象，以及需要在 值”列中配置的每个object的属性。

很多级别的object类型、名称和属性都不需要配置，因此对应的 Value”字段不接受输入。如果输入合适且被接受，则在字段的右端会显示一个下拉箭头；当您单击此箭头时，将显示一个可能值的简短列表以供选择。某些值（例如部件的 部件”）添加更多的参数层和属性，您可以单击…按钮再次选择和设置参数的值。对于数据集，输入对话框允许您输入或导入值，例如初始值或默认值；请参阅模型分析使用数据集帮助主题。

元素类型	行为
值类型	ValueType 元素要么从原始类型泛化，要么被 SysMLSimReal 替代以进行仿真。
块	映射到 SysMLSimClass 或 SysMLSimModel 元素的块元素支持数据集的创建。如果您在上下文中定义了多个数据集（可以泛化），则必须将其中一个标识为默认值（使用时间菜单选项 设置为默认数据集”）。 由于 SysMLSimModel 可能是模拟的顶级元素，并且不会被概括，如果您定义了多个数据集，则在模拟期间选择要使用的数据集。
属性	指定常量或变量及其设置的首选方法是在属性本身上使用 SysPhS 构造型 PhSConstant 和 PhSVariable。 PhSVariable 构造型具有内置属性*isContinuous*、*isConserved*和*changeCycle*。 该属性将列在 PhSConstant 或 PhSVariable 下，并且该值不能更改。 也可以在配置SysML仿真窗口中定义设置。在这种情况下，它们将列在 属性”下。 块内的属性可以配置为 SimConstants 或 SimVariables。对于 SimVariable，您可以配置以下属性： • isContinuous — 确定属性值是连续变化（'true'，默认值）还是离散变化（'false'） • 属性— 确定属性的值是否保留（'true'）（'false'，默认值）；在为物理相互作用建模时，相互作用包括保守物理物质的交换，例如电流、力或流体流动 • changeCycle — 指定离散属性值更改的时间间隔；默认值为 0” - changeCycle 只能设置为 0 以外的值 isContinuous = 'false' - changeCycle 的值必须为正或等于 0
端口	无需配置。
模拟函数	函数被创建为块或约束块中的操作，原型为 SimFunction”。 配置仿真窗口无需配置。

概括	无需配置。
捆绑连接器	将属性绑定到约束属性的参数。 无需配置；但是，如果属性不同，系统会提供同步它们的选项。
连接器	连接两个端口。 配置仿真窗口无需配置。但是，您可能必须通过确定属性属性是否应该设置为 "False"（对于潜在的，以便建立相等耦合）或 "True"（对于流/保守的属性）来配置端口类型的属性, 从而建立和到零的耦合)。
约束块	无需配置。

仿真标签

此表描述了配置SysML仿真窗口上的 仿真"选项卡的字段。

字段	行动
模型	单击下拉箭头并选择模拟的顶级节点（SysMLSimModel元素）。该列表填充有定义为顶级模型节点的块的名称。
数据集	单击下拉箭头并选择所选模型的数据集。
预验证	选中此复选框可在执行模型的每次模拟之前自动验证模型。
开始	类型在开始模拟之前的初始等待时间，以秒为单位（默认值为 0）。
停止	类型模拟将执行的秒数。
格式	单击下拉箭头并选择 "plt"、 "csv"或 "mat"作为结果文件的格 ，其他工具可能会使用这些格式。
参数图	• 选中此复选框以在 y 轴上绘制图例A ，在 x 轴上图例B • 取消选中复选框以在 y 轴上绘制图例，在 x 轴上绘制时间 注记：选中复选框后，您必须选择两个要绘制的属性。
使用Simscape	（如果选定的数学工具是 Simulink）如果您还想在 Simscape 中处理仿真，请选中该复选框。
依赖项	列出模拟该模型必须生成的类型。
属性的属性	提供与模拟有关的变量属性列表。选中要绘制的每个属性对应的复选框。

创建参数模型

在本主题中，我们将讨论如何开发用于仿真的 SysML模型元素（假设已有 SysML 建模知识），在配置SysML仿真窗口中配置这些元素，并观察在一些不同定义和建模方法下的仿真结果。本章提供的 SysML仿真示例中的图表和屏幕快照说明了这些要点。

创建参数模型时，您可以应用以下三种方法之一来定义约束方程：

- 在块元素上定义内联约束方程
- 创建可重用的约束块，以及
- 使用连接约束属性

您还将考虑：

- 物理交互中的流动
- 默认值和初始值
- 仿真函数
- 价值分配，以及
- 包导入

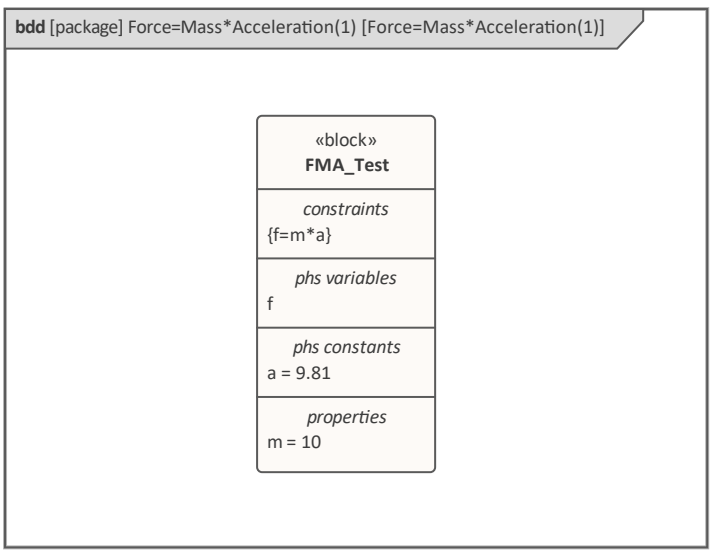
访问

功能区	仿真>系统行为> Modelica/Simulink >配置管理器

在块上定义内联约束方程

直接在块中定义约束很简单，也是定义约束方程的最简单方法。

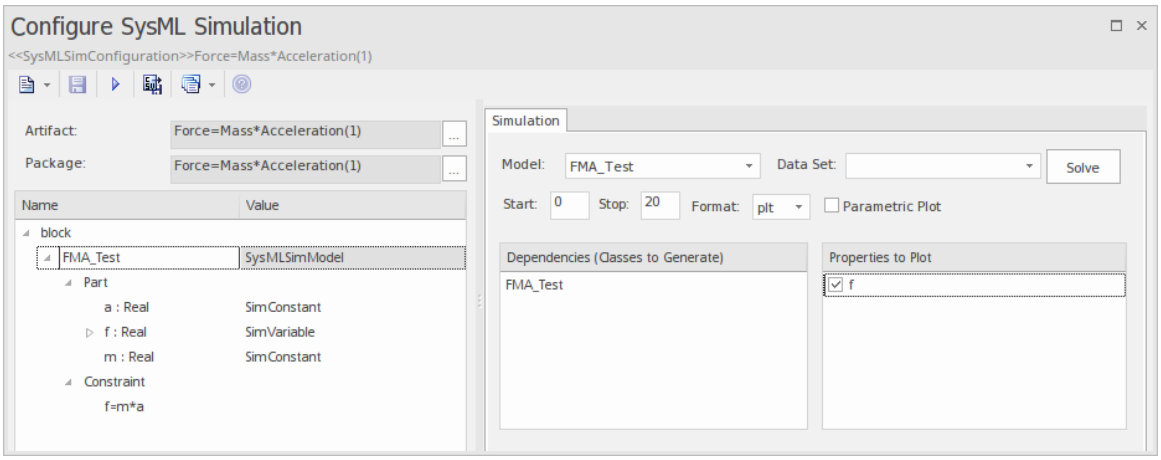
在这个图中，约束'f = m * a'被定义在一个块元素。



提示：您可以在一个块中定义多个约束。

1. 创建一个SysMLSim配置工具工件=Mass*Acceleration(1)'并将其指向包'FMA_Test'
2. 对于 "FMA_Test"，在 "值"列中设置 "SysMLSimModel"。

3. 对于 á"、 ḿ"和 f"部分，在 值"列中：将 á"和 ḿ"设置为 ƤhSConstant"，（可选）将 f"设置为 "PhSVariable"。
4. 在 仿真"选项卡的 要绘制的属性"面板中，选中 f"复选框。
5. 单击求解按钮运行模拟。

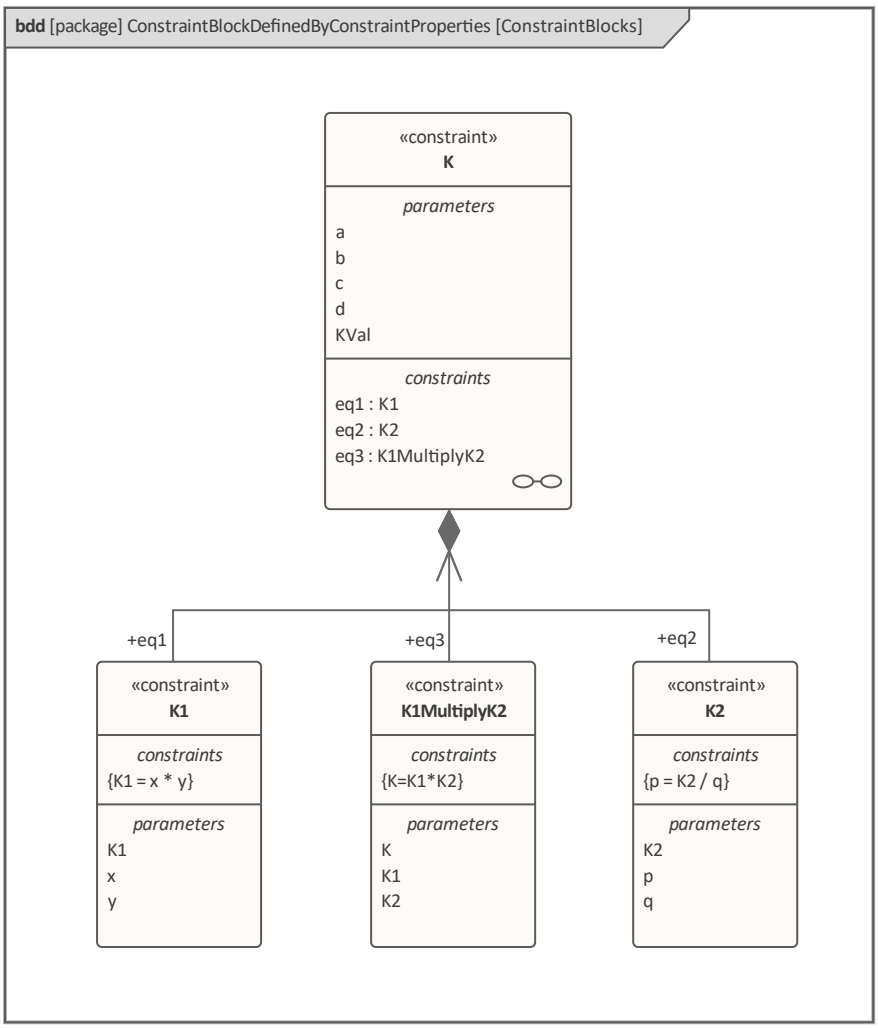


应A f = 98. 1 （来自 10 * 9.81）绘制图表。

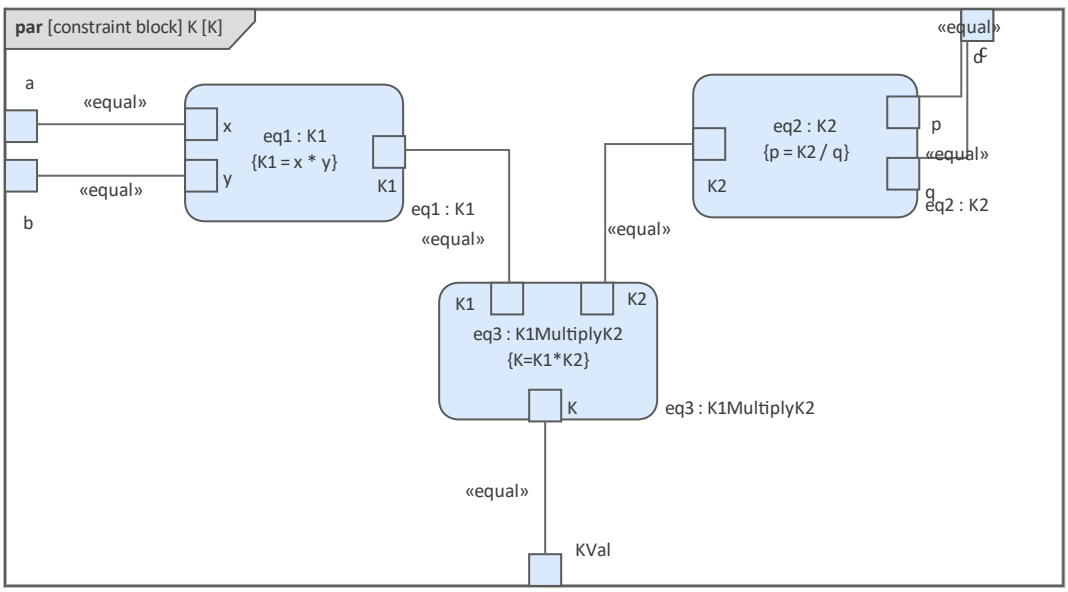
连接约束属性

在 SysML 中，ConstraintBlocks 中存在的约束属性可用于在定义约束时提供更大的灵活性。

在这个图中，ConstraintBlock ' K ' 定义了参数'a'、'b'、'c'、'd'和'属性'，以及三个约束'eq1'、'eq2'和'eq3'，类型为 'K1 '、'K2' 和 'K1MultiplyK2' 分别。

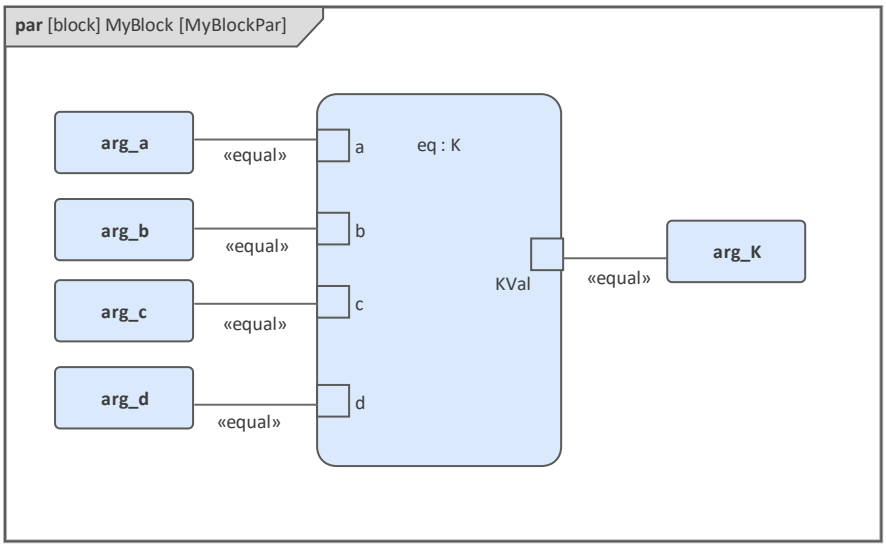


在 ConstraintBlock ' K ' 中创建一个参数图，并使用捆绑连接器将参数连接到约束属性，如图所示：

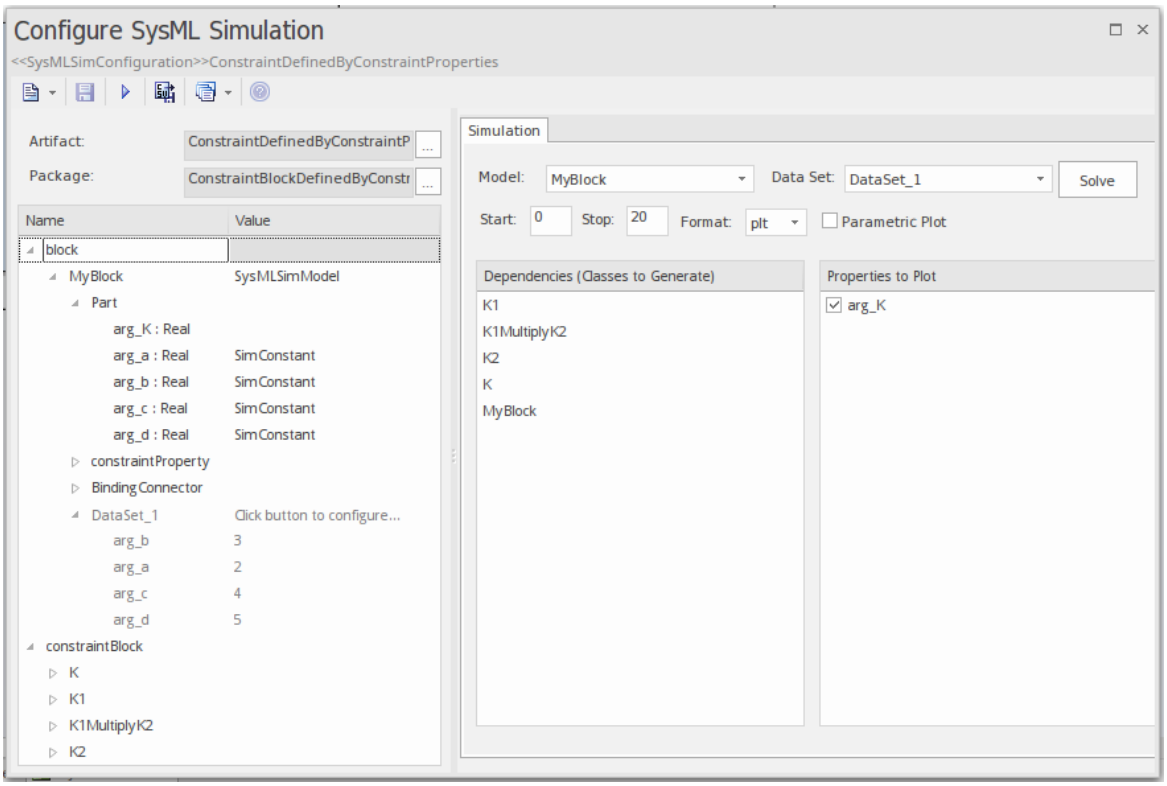


- 创建具有五个属性（零件）的模型属性
- 为属性创建一个约束属性 'eq' 并显示参数

- 将属性绑定到参数



- 在数据集中提供值 (arg_a = 2, arg_b = 3, arg_c = 4, arg_d = 5)
- 在 "配置SysML仿真"对话框中，将 "模型"设置为 "MyBlock"，将 "数据集"设置为 "DataSet_1"
- 在 "propertyto Plot"面板中，选中 "属性"复选框
- 点击 Solve 按钮运行模拟



将计算并绘制结果 120（计算为 2 * 3 * 4 * 5）。这与我们用笔和纸进行扩展时相同： K = K1 * K2 = (x*y) * (p*q)，然后绑定值 (2 * 3) * (4 * 5)；我们得到 120。

有趣的是，我们有意将 K2 的方程定义为 "p = K2 / q"，这个例子仍然有效。

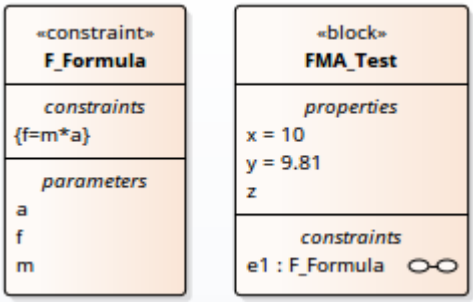
在这个例子中，我们可以很容易地将 K2 求解为 p * q，但在一些复杂的例子中，从方程中求解变量是非常困难的；但是， Enterprise Architect SysMLSim 仍然可以做到这一点。

总之，该示例向您展示了如何通过构造约束属性来定义具有更大灵活性的 ConstraintBlock。虽然我们只在 ConstraintBlock 中演示了一层，但这种机制将适用于任意级别的复杂模型。

创建可重用的约束块

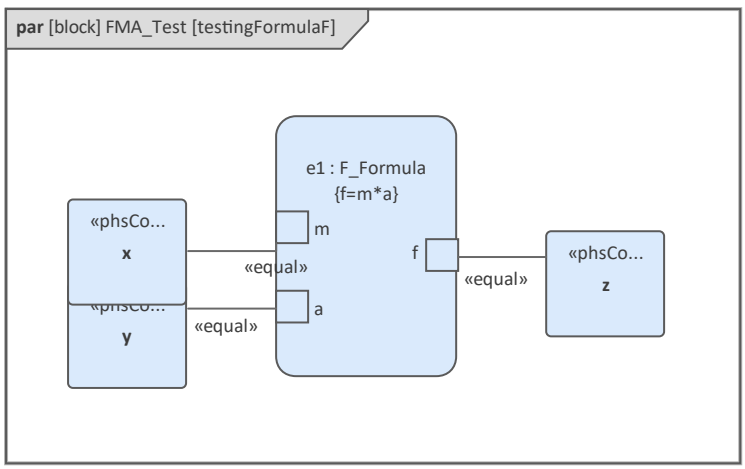
如果一个方程在许多块中通用，则可以创建一个 ConstraintBlock 用作每个块中的约束属性。这些是我们根据前面的示例所做的更改：

- 创建一个 ConstraintBlock元素'F_Formula' 具有三个参数 'a'、'm' 和 'f'，以及一个约束 'f = m * a'

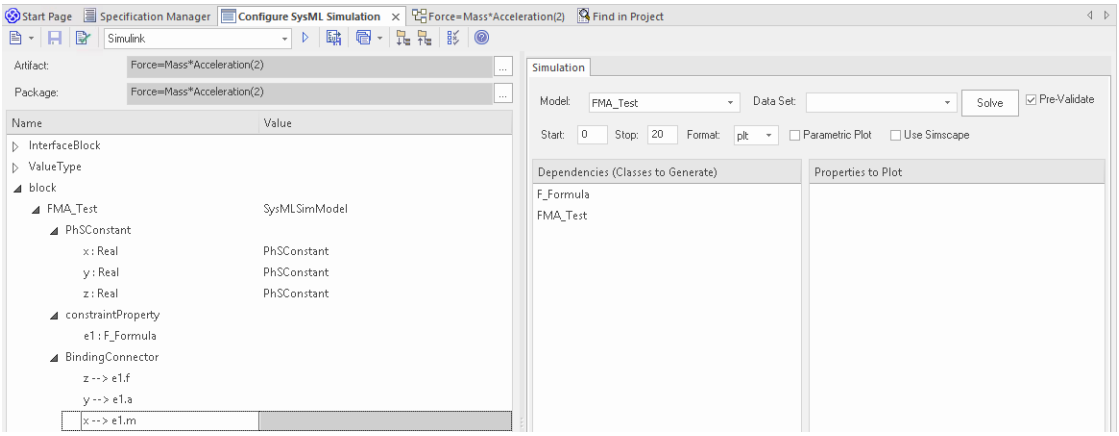


提示：如果属性类型为空，将应用原始类型 *'Real'*

- 用'x'、'y'和'z'三个属性创建一个块'属性'，并分别赋予'x'和'y'默认值'10'和'9.81'
- 在 "FMA_Test" 中创建一个参数图，显示属性 "x"、"y" 和 "z"
- 创建一个 ConstraintProperty 'e1' 类型为 'F_Formula' 并显示参数
- 在'x-m'、'y-a'和'f-z'之间划捆绑连接器，如图所示：



- 在元素中创建一个工件并对其进行配置，如图所示：
  - 在 "值" 列中，将 "FMA_Test" 设置为 "SysMLSimModel"
  - 在 "值" 列中，将 "x" 和 "y" 设置为 "PhSConstant"
  - 在 "要绘制的属性" 面板中，选中 "Z" 复选框
  - 单击求解按钮运行模拟



应A f = 98. 1 （来自 10 * 9.81）绘制图表。

物理交互中的流动

在对物理相互作用进行建模时，应将电流、力、扭矩和流速等守恒物理物质的交换建模为流，并且应将流变量设置为属性 "isConserved"。

连接建立两种不同类型的耦合，取决于流属性是潜在的（默认）还是流（守恒）：

- 等式耦合，用于潜在（也称为努力）属性
- 和零耦合，用于流（守恒）属性；例如，根据电学领域的基尔霍夫电流定律，电荷守恒使所有电荷流入一个和为零的点

在 "ElectricalCircuit"示例的生成 OpenModelica 代码中：

连接器充电端口

流动电流 i； // 如果 'isConserved' = true 将生成流关键字

电压v；

结束充电端口；

电路模型

源；

电阻电阻器；

接地；

方程

连接（源.p，电阻器.n）；

连接（接地.p，源.n）；

连接（电阻器.p，源.n）；

结束电路；

每个连接方程实际上扩展为两个方程（ChargePort 中定义了两个属性），一个用于等式耦合，另一个用于和零耦合：

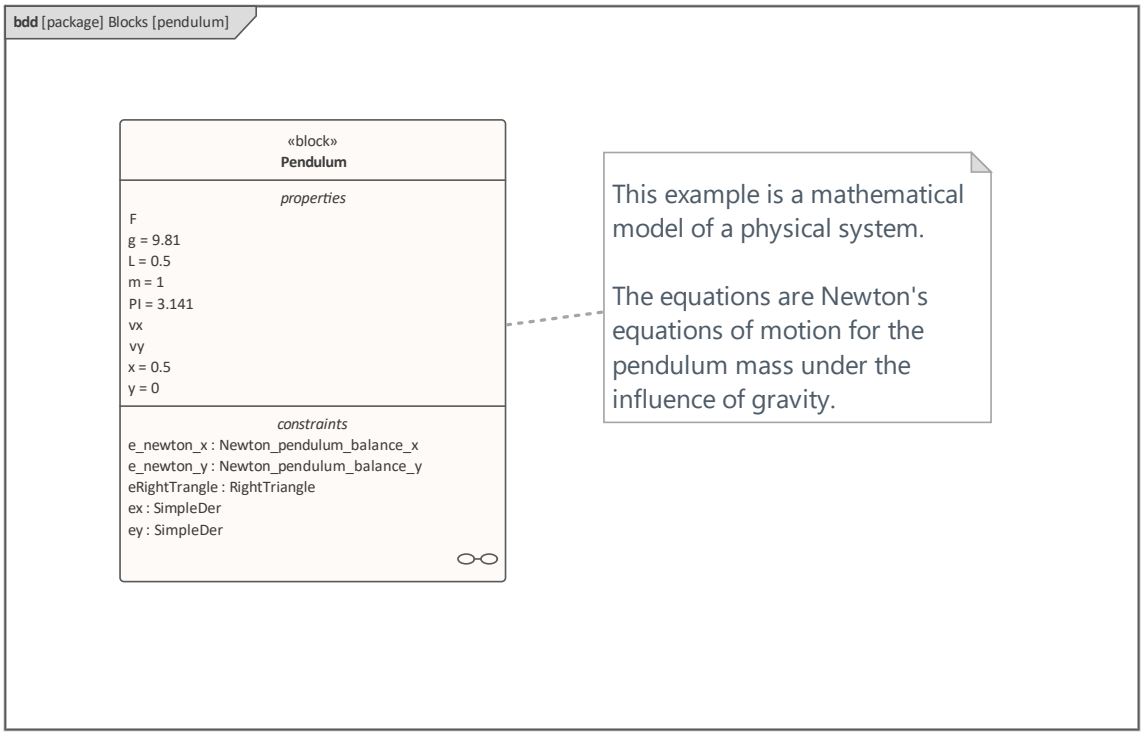
源.pv = 电阻器.nv;

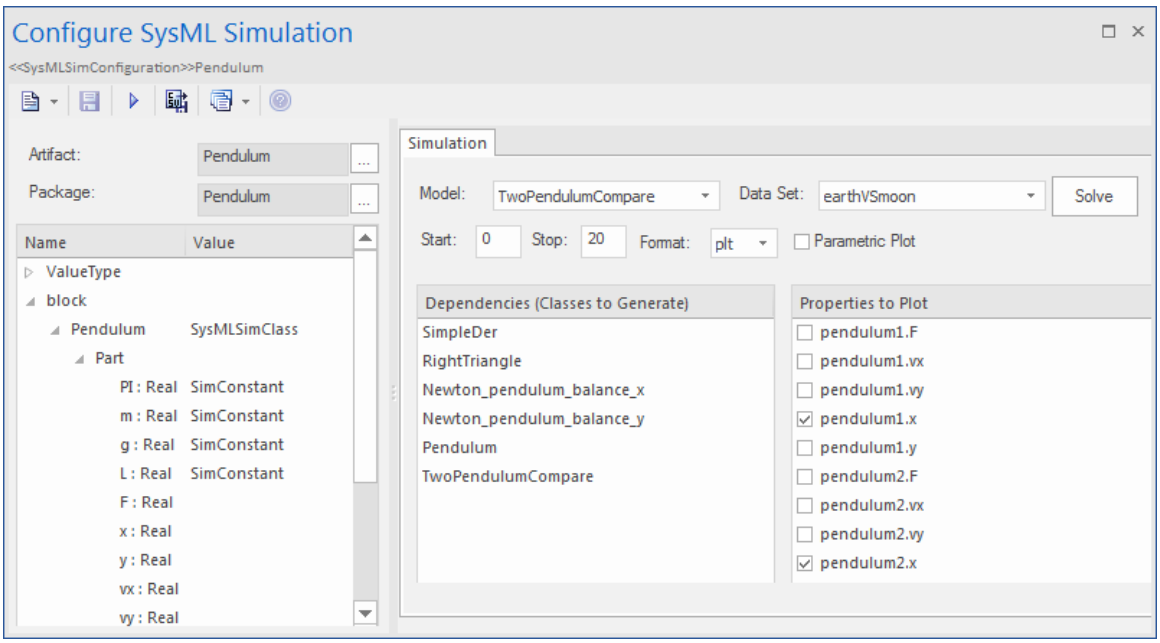
源.pi + 电阻.ni = 0;

默认值和初始值

如果在 SysML属性元素（ 属性”对话框 > 属性”页面 > 初始”字段）中定义了初始值，则它们可以作为 PhSConstant 的默认值或 PhSVariable 的初始值加载。

在这个 Pendulum 示例中，我们为属性 ǵ”、 Ł ”、 ḿ”、 ₱I”、 x́”和 ý”提供了初始值，如图左侧所示。由于 “PI”（数学常数）、 ḿ”（摆锤的质量）、 ǵ”（重力因子）和 Ł ”（摆锤的长度）在模拟过程中不会发生变化，因此将它们设置为 ₱hSConstant”。





生成的 Modelica 代码如下所示：

类摆
参数 Real PI = **3.141；**
参数 Real m = **1** ;
参数 Real g = **9.81** ;
参数 Real L = **0.5** ;
实F；
实数 x **(start=0.5)** ;
实 y **(start=0)** ;
真正的vx；
真正的 vy;
……
方程
……
结束摆；

- 属性'PI'、'm'、'g'和' L '是常数，并作为声明方程生成
- 属性 x‴和 y‴是可变的；它们的起始值分别为 0.5 和 0，初始值作为修改生成

仿真函数

仿真函数是编写复杂逻辑A有用工具，并且易于用于约束。本节介绍 TankPI 示例中的一个函数。
在 ConstraintBlock 'Q_OutFlow' 中，函数'LimitValue' 被定义并在约束中使用。

- 在块或约束块上，创建一个操作（本例中为 L‴imitValue”）并打开特征窗口的 操作”选项卡

- 给操作定型 "SimFunction"
- 定义参数并将方向设置为 "输入/输出"

提示：多个参数可以定义为'out'，调用者取值的格式为：

*(out1, out2, out3) = function_name(in1, in2, in3, in4, ...); //*方程形式

*(out1, out2, out3) := function_name(in1, in2, in3, in4, ...); //*报表形式

- 在属性窗口的 "代码"选项卡的文本字段中定义函数体，如图：

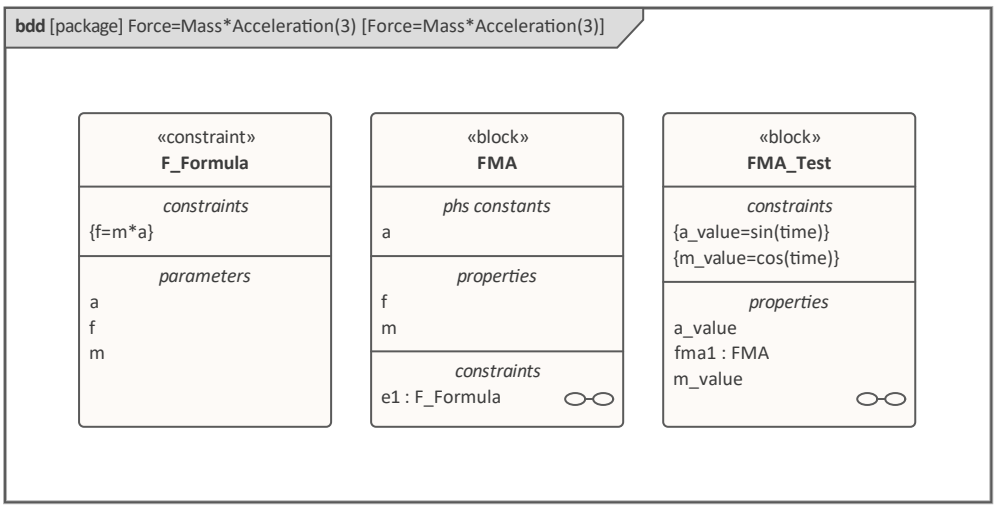
```
pLim :=
如果 p > pMax 那么
最大
else如果 p < pMin 那么
pMin
else
p;
```

生成代码时， Enterprise Architect将收集所有在 ConstraintBlocks 和 Blocks 中定义为 "SimFunction"的操作，然后生成类似于以下内容的代码：

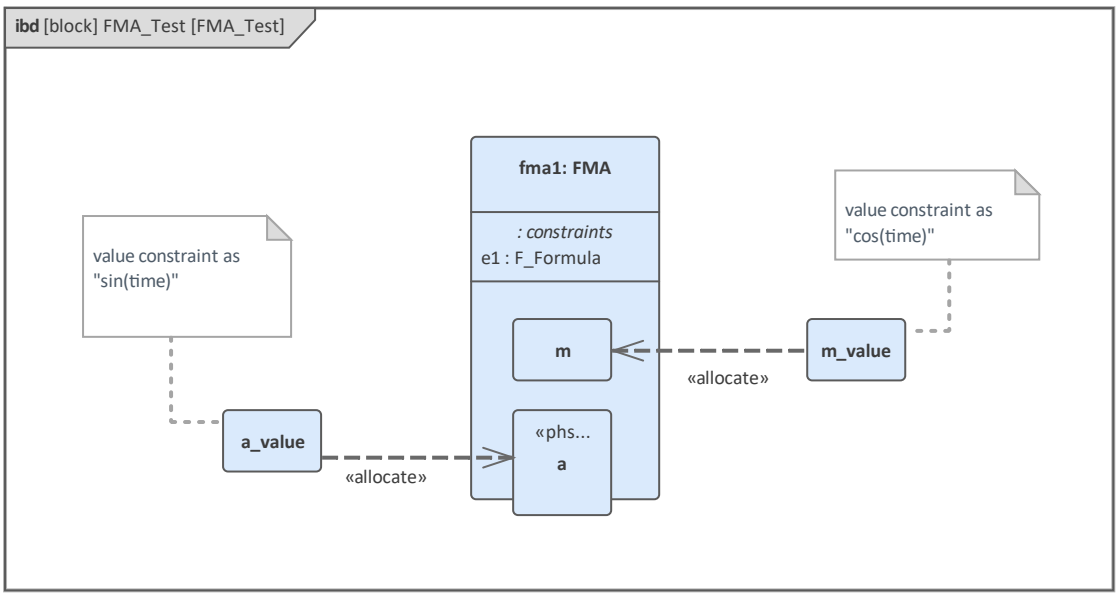
```
函数极限值
输入Real pMin；
输入真实 pMax；
输入实数 p;
输出实 pLim；
算法
pLim :=
如果 p > pMax 那么
最大
else如果 p < pMin 那么
pMin
else
p;
结束极限值；
```

价值分配

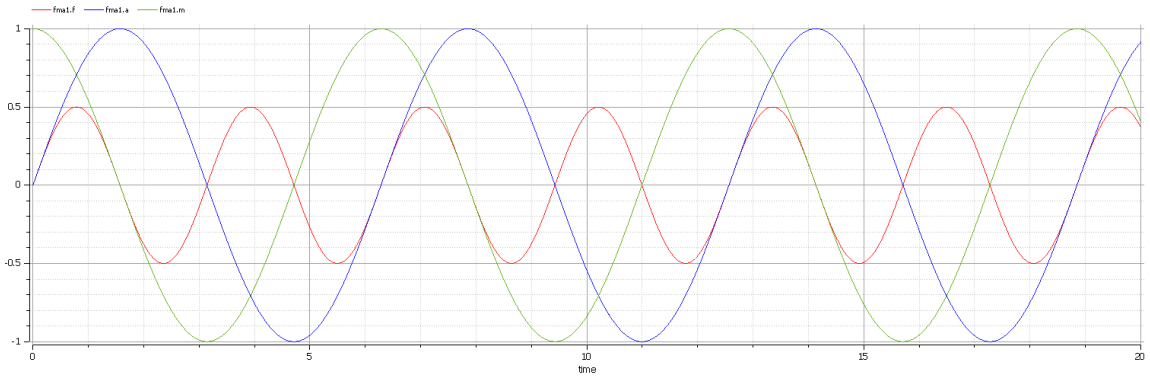
该图显示了一个名为 "力=质量*加速度"的简单模型。



- 块 f"FMA" A属性 á"、 f"和 ḿ"以及约束属性 é1"建模，输入约束块 f"_Formula"
- 块'FMA'的属性没有设置任何初始值，并且属性'a'、'f'和'm'都是可变的，所以它们的值变化取决于它们被模拟的环境
- 创建一个块'FMA_Test'作为SysMLSimModel并添加属性'fma1'来测试块'FMA'的行为
- 约束'a_value' 为' sin (time)'
- 约束'm_value'为' cos (time)'
- 划分配连接器将值从环境分配到模型'FMA'



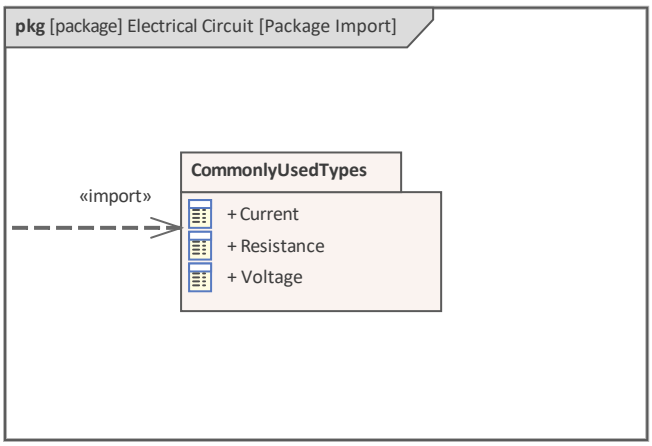
- 选中针对 f"ma1.a"、 f"ma1.m"和 f"ma1.f"的 要绘制的属性"复选框
- 单击求解按钮以模拟模型



包导入

工件选择一个包的元素（如Blocks、ConstrainBlocks和Value Types）如果模拟依赖于不属于这个包的元素，比如Reusable库， Enterprise Architect在包元素之间提供了一个导入连接器来满足这个需求。

在电气电路示例中，工件被配置为 "ElectricalCircuit"包，其中包含模拟所需的几乎所有元素但是，一些属性被键入为值类型，例如 "电压"、 "电流"和 "电阻"，这些值类型在多个 SysML 模型中常用，因此放置在单个 SysML 模型之外的一个名为 "CommonlyUsedTypes"的包中。如果使用导入连接器导入此包，则导入包中的所有元素都将出现在导入配置管理器中。



使用数据集进行模型分析

参数模型中使用的每个 SysML块都可以在仿真配置中定义多个数据集。这允许使用相同的 SysML模型进行可重复的模拟变化。

块A键入为 SysMLSimModel（不能被概括或构成组合的一部分的顶级节点）或 SysMLSimClass（可以被概括或构成组合的一部分的较低级别的元素）。在 SysMLSimModel元素上运行模拟时，如果您定义了多个数据集，则可以指定要使用的数据集。但是，如果模拟中的类有多个数据集，则您无法选择在模拟期间使用哪一个，因此必须将一个数据集标识为该类的默认值。

访问

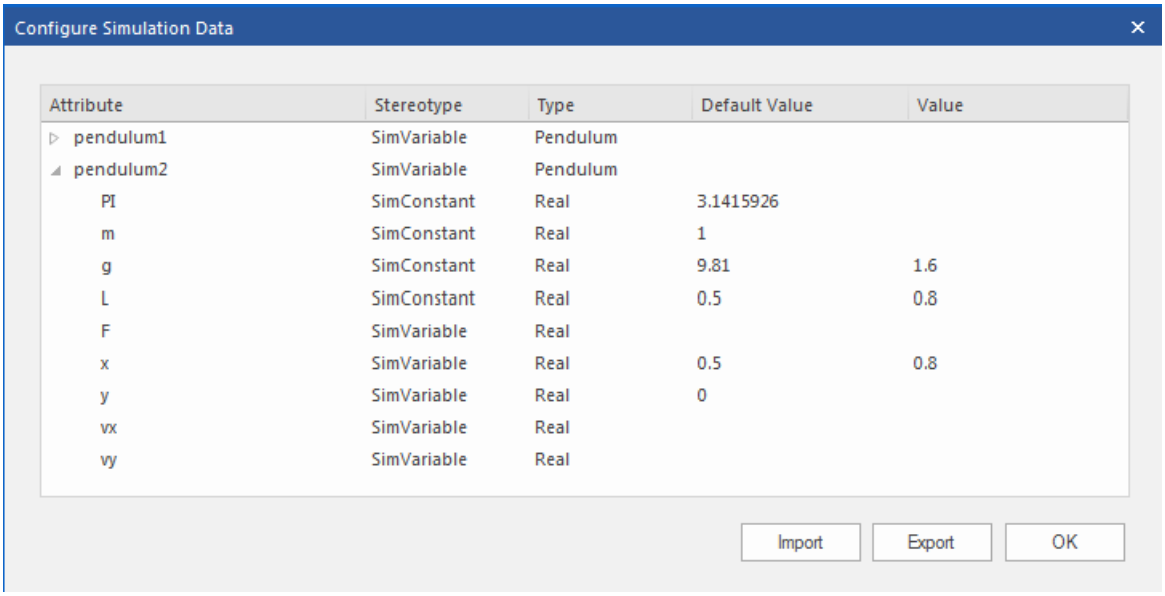
功能区	仿真>系统行为>Modelica/Simulink>配置管理器>在 块"组>名称列>块元素上下文菜单>创建仿真数据集

数据集管理

任务	行动
创造	要创建新数据集，请右键单击块名称并选择 创建仿真数据集"选项。数据集被添加到块名称下方的组件列表的末尾。单击 ... 按钮在 配置仿真数据"对话框中设置数据集（参见配置仿真数据表）。
复制	要复制现有数据集作为创建新数据集的基础，请右键单击数据集名称并选择 复制"选项。重复的数据集将添加到块名称下方的组件列表的末尾。单击 ... 按钮以编辑 配置仿真数据"对话框中的数据集（请参阅配置仿真数据表）。
删除	要删除不再需要的数据集，请右键单击数据集并选择 删除数据集"选项。
默认设置	要设置 SysMLSimClass 在用作属性类型或继承（以及当有多个数据集时）使用的默认数据集，请右键单击数据集并选择 设置为默认值"选项。默认数据集的名称以粗体突出显示。模型使用的属性将使用此默认配置，除非模型明确覆盖它们。

配置仿真数据

此对话框主要用于提供信息。您可以直接添加或更改数据的唯一列是 值"列。



柱子	描述
属性	属性”列提供了正在编辑的块中所有属性的树视图。
构造型	对于每个属性， 构造型”列标识它是否已配置为模拟期间的常量，或者其值预计会随时间变化的变量。
类型	类型”列描述了用 模拟此属性的类型。它可以是原始类型（例如 Real”）或对模型中包含的块的引用。属性引用块将显示由它们下面的引用块指定的子属性。
默认值	如果未提供覆盖， 默认值”列显示将在模拟中使用的值。这可以来自 SysML模型中的 初始值”字段或来自父类型的默认数据集。
价值	值”列允许您覆盖每个原始值的默认值。
导出/导入	单击这些按钮可使用电子表格等外部应用程序修改当前数据集中的值，然后将它们重新导入列表。

SysML仿真实例

本节为每个阶段提供了一个工作示例：为域创建 SysML模型，对其进行仿真，并评估仿真结果。这些示例应用了前面主题中讨论的信息。

例子

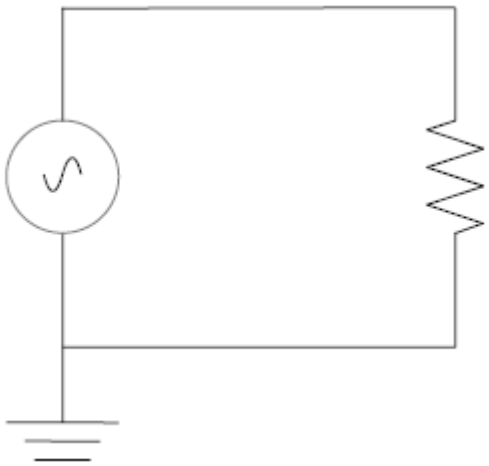
模型	描述
电路仿真示例	第一个例子是加载电路的模拟。该示例从电路图开始，并将其转换为参数模型。然后对模型进行仿真，评估电阻器源端和目标端的电压并将其与预期值进行比较。
质量弹簧阻尼振荡器仿真示例	第二个例子使用一个简单的物理模型来演示质量-弹簧-阻尼系统的振荡行为。
水箱压力调节器	最后一个示例显示了两个水箱的水位，其中水在它们之间分配。我们首先模拟一个平衡良好的系统，然后我们模拟一个水将从第二个水箱溢出的系统。

电路仿真示例

在本例中，我们为一个简单的电路创建 SysML 参数模型，然后使用参数仿真来预测和绘制该电路的行为。

图表

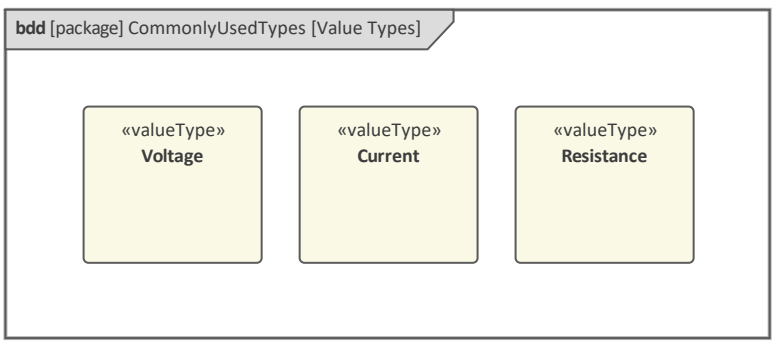
我们将要学习的电路模型（此处显示）使用标准电路符号。

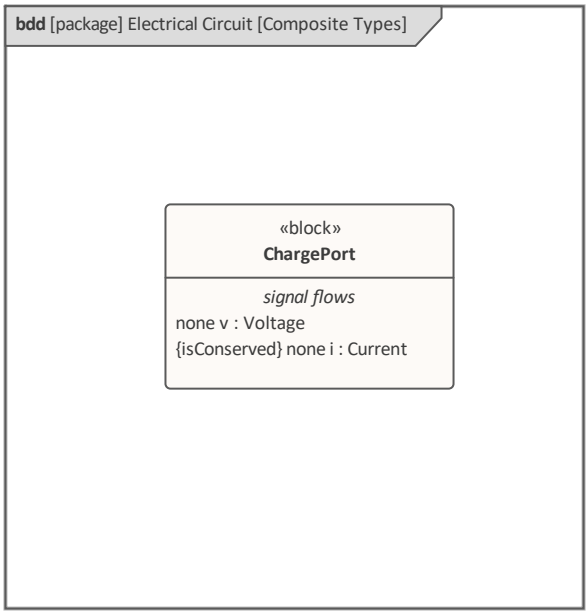


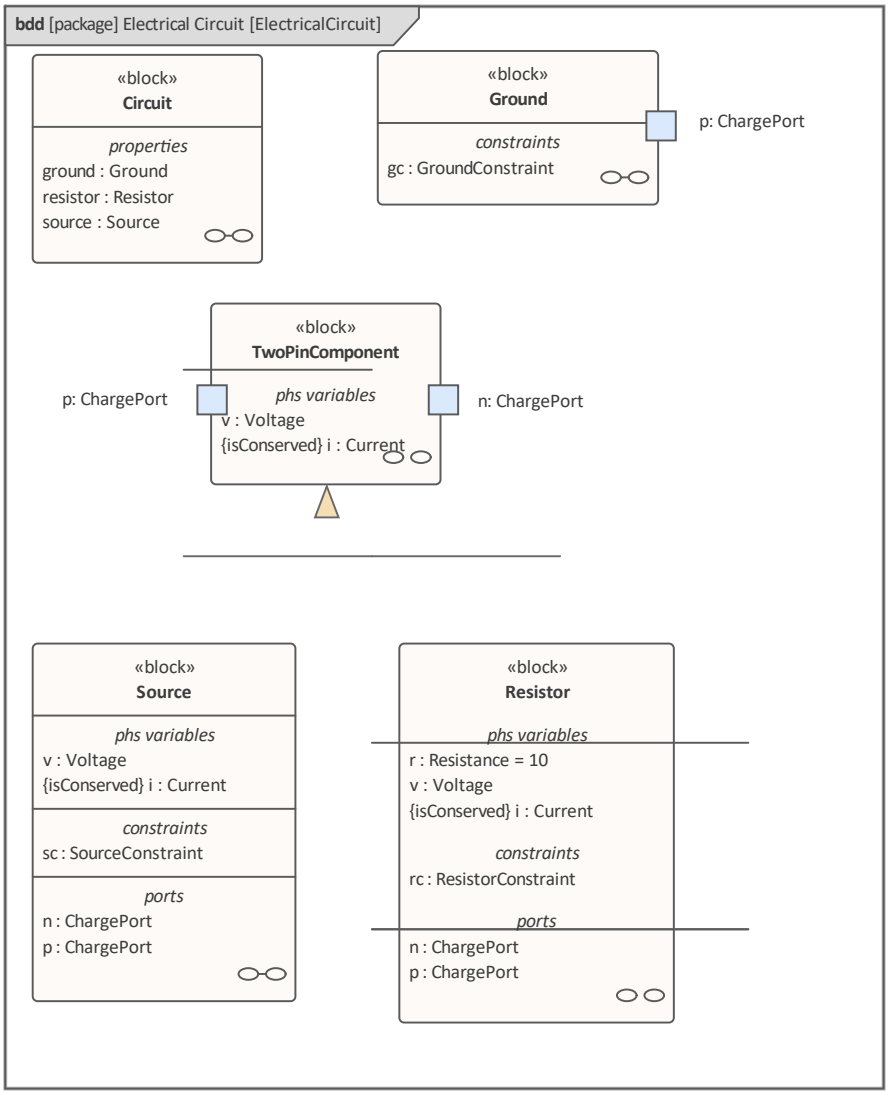
该电路包括交流电源、源和电阻器，通过电线相互连接。

创建 SysML模型

这张表显示了我们如何构建一个完整的完成模型来表示电路，从最低级别的类型开始，一步一步地构建模型。

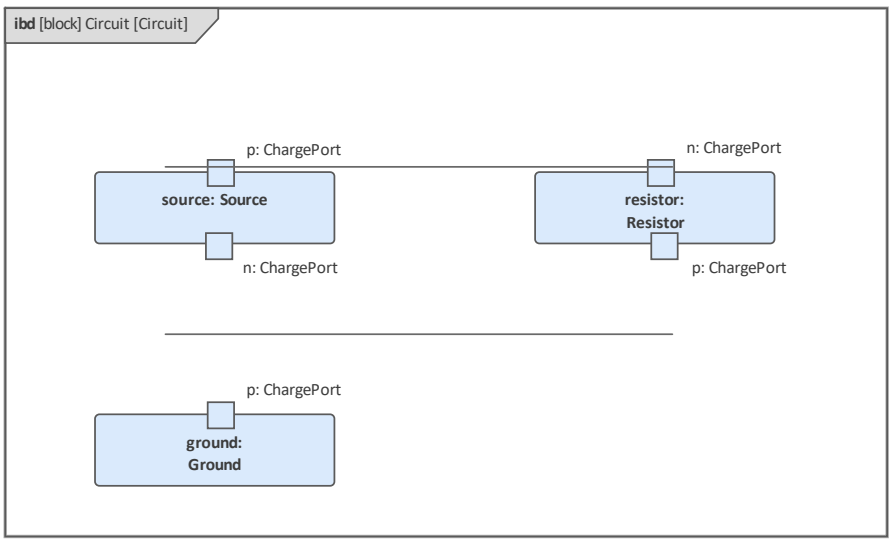
部件	行动
类型	定义电压、电流和电阻的值类型。单位和数量种类对于模拟而言并不重要，但如果定义一个完整的完成模型则需要设置。这些类型将从原始类型“Real”泛化。在其他模型中，您可以选择将值类型映射到与模型分开的相应仿真类型。  此外，定义一个名为 ChargePort 的复合类型 (块)，其中包括电流和电压属性。这种类型允许我们表示组件之间连接器处的电能。

	
块	在 SysML 中，电路和每个组件都将表示为块。 在块定义图 (图表) 中，创建一个电路块。该电路由三部分组成：源、地和电阻。这些部分属于不同的类型，具有不同的行为。 为每个零件类型创建一个块。电路块的三个部分通过端口连接，代表电气引脚。源和电阻有一个正极和一个负极引脚。地只有一个引脚，它是正极。电流（电荷）通过引脚传输。创建一个带有两个端口（引脚）的抽象块“TwoPinComponent”。这两个端口分别命名为 p”（正）和 n”（负），它们的类型为 ChargePort. 此图显示了 BDD，包括电路块、接地、TwoPinComponent、源和电阻器。

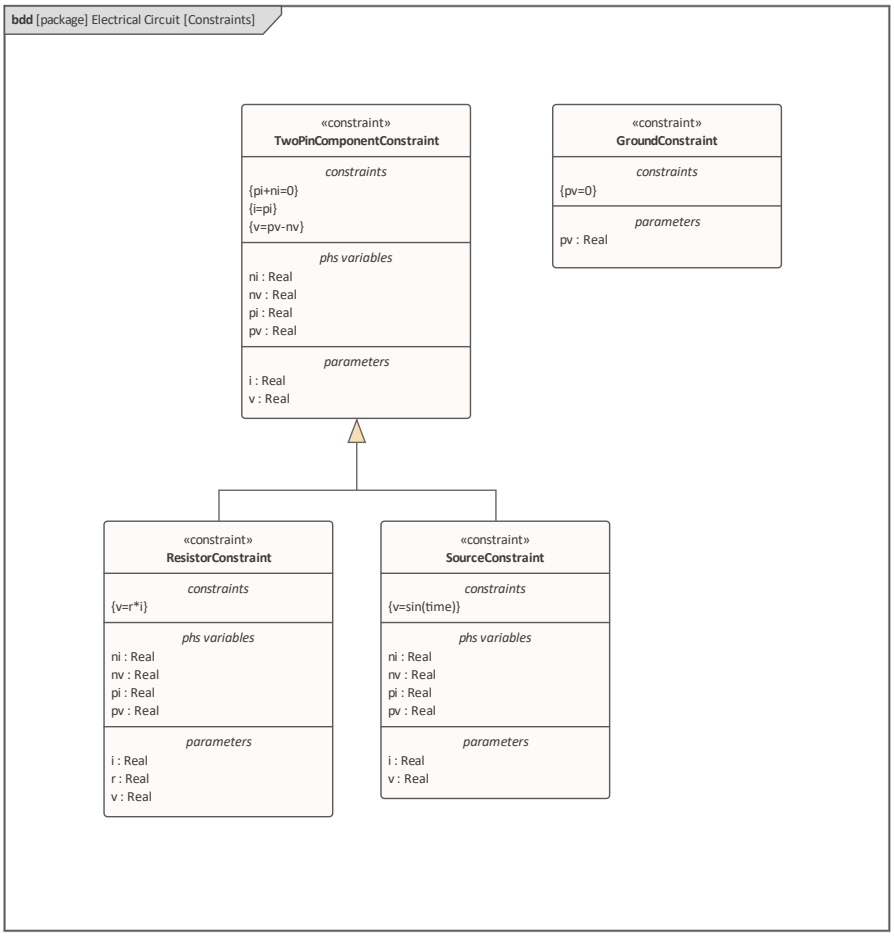


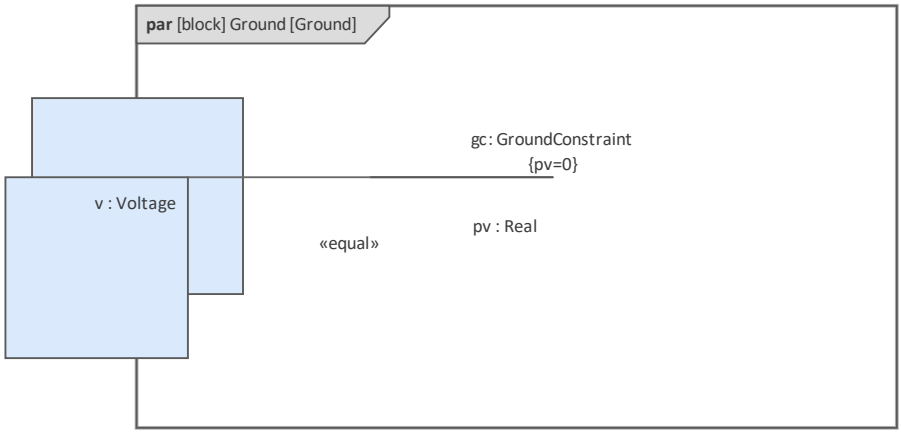
内部结构

为电路创建一个内部块图（图表）。为源、电阻和接地添加属性，由相应的块键入。用连接器连接端口。源的正极引脚连接到电阻器的负极引脚。Resistor 的正极引脚连接到源的负极引脚。地也连接到源的负极引脚。



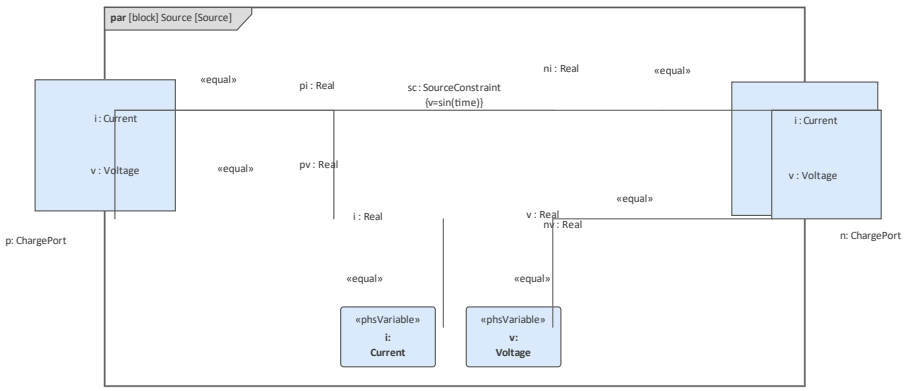
请注意，这遵循与原始电路图相同的结构，但每个组件的符号已替换为由我

	们定义的块键入的属性。
约束	方程定义数值属性之间的数学关系。在 SysML 中，方程表示为 ConstraintBlocks 中的约束。 ConstraintBlocks 的参数对应于块的 PhSVariables 和 PhSConstants（本例中为'i'、'v'、'r'），以及端口类型中存在的端口（'pv'、'pi'、'nv '，在本例中为 'ni'）。 创建一个 ConstraintBlock 'TwoPinComponentConstraint' 来定义源和电阻器共有的参数和方程。该等式应状态组件的电压等于正负引脚电压之差。组件的电流等于通过正极引脚的电流。通过两个引脚的电流之和必须加起来为零（一个是另一个的负数）。接地约束表明接地引脚上的电压为零。源约束将电压定义为sine ，以当前仿真时间为参数。此图显示了这些约束如何在 BDD 中呈现。 
绑定	约束参数的值等同于具有绑定连接器的可变值和常数值。在每个块上创建约束属性（属性类型为 ConstraintBlocks），并将块变量和常量绑定到参数，以将约束约束块。这些图分别显示了接地、源和电阻的绑定。 对于地面约束，将 gc.pv 绑定到 pv



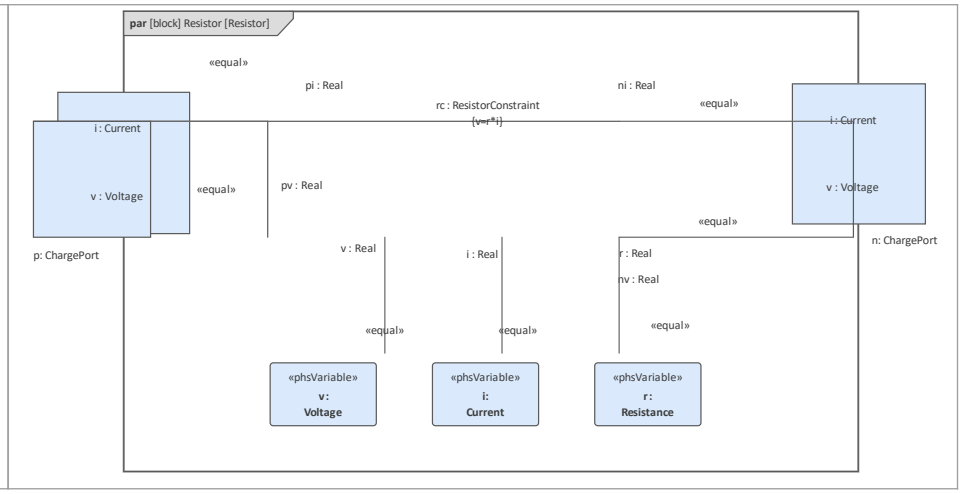
对于源约束，绑定：

- sc.pi 到 pi
- sc.pv 到 pv
- sc.v 到 v
- sc.i到我
- sc.ni 到 ni 和
- sc.nv 到 nv



对于电阻器约束，绑定：

- rc.pi 到 pi
- rc.pv 到 pv
- rc.v 到 v
- rc.i 到我
- rc.ni 到 ni
- rc.nv 到 nv 和
- rc.r 到 r



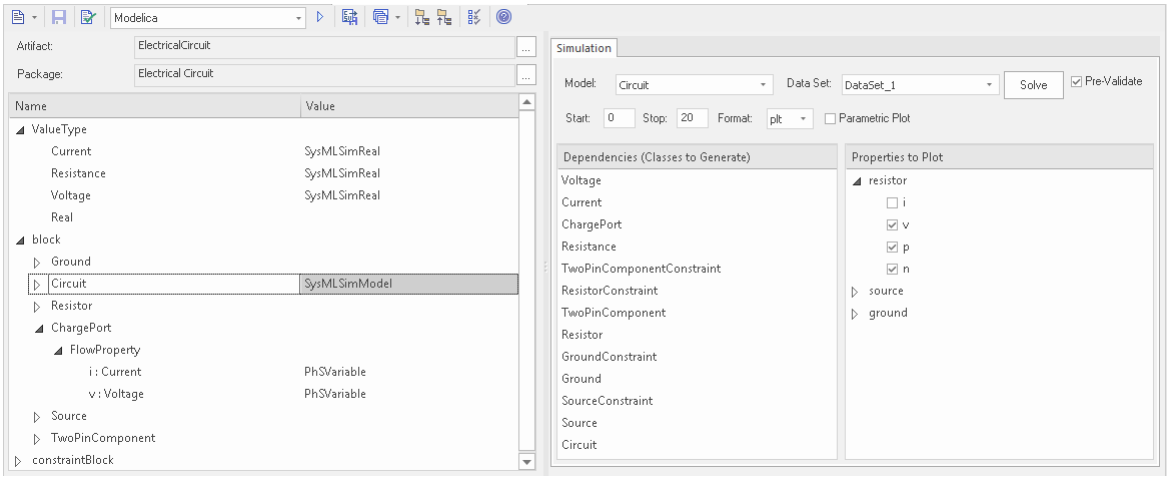
配置仿真行为

此表显示了 SysMLSim 配置的详细步骤。

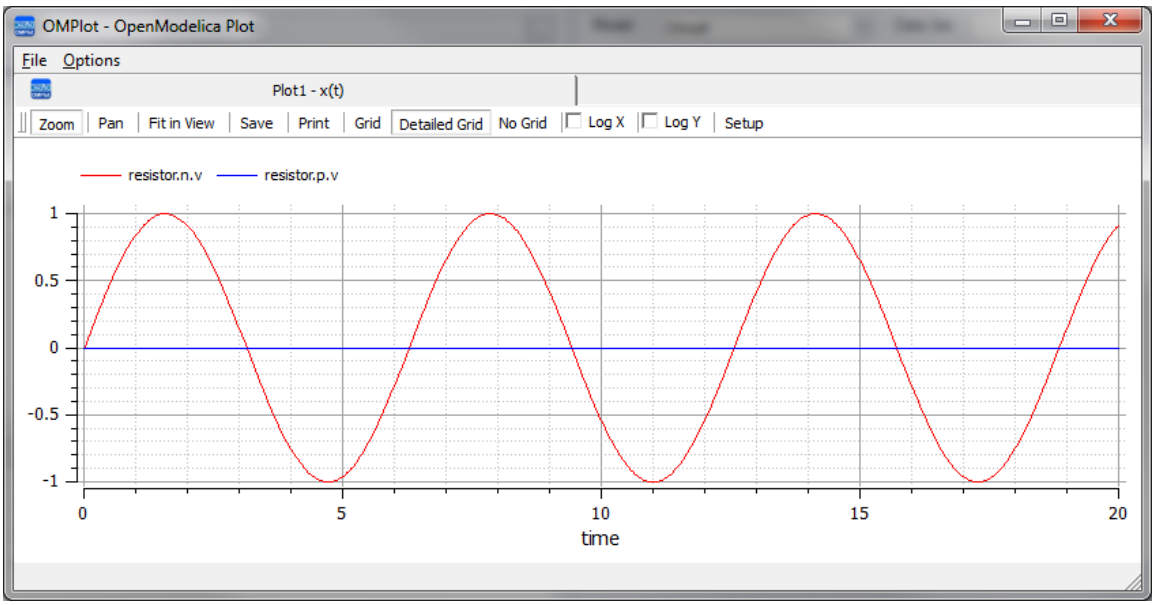
节	行动
工件适合	• 选择 仿真>系统行为> Modelica/Simulink >配置管理器” • 从第一个下拉菜单中，选择工具栏创建工具工件元素创建工件 • 选择拥有此 SysML模型的包
在配置管理器中创建根元素	• 值类型 • 块 • 约束块
值类型替换	展开 ValueType 并为 Current、Resistance 和 Voltage 中的每一个从 “Value”组合框中选择 “SysMLSimReal”。
将属性设置为流	• 将 块”扩展到 ChargePort |流属性 | i ：当前并从 值”组合框中选择 “SimVariable” • 对于 无素”，单击 … 按钮以打开 “ElementConfigurations”对话框 • 将 “isConserved”设置为 “True ”
SysMLSimModel	这就是我们要模拟的模型：将块'Circuit'设置为'SysMLSimModel'。

运行仿真

在 仿真”页面中，选中 “resistor.n”和 “resistor.p”复选框进行绘图，然后单击 求解”按钮。



如图所示，绘制了两个图例 “resistor.n”和 “resistor.p”。

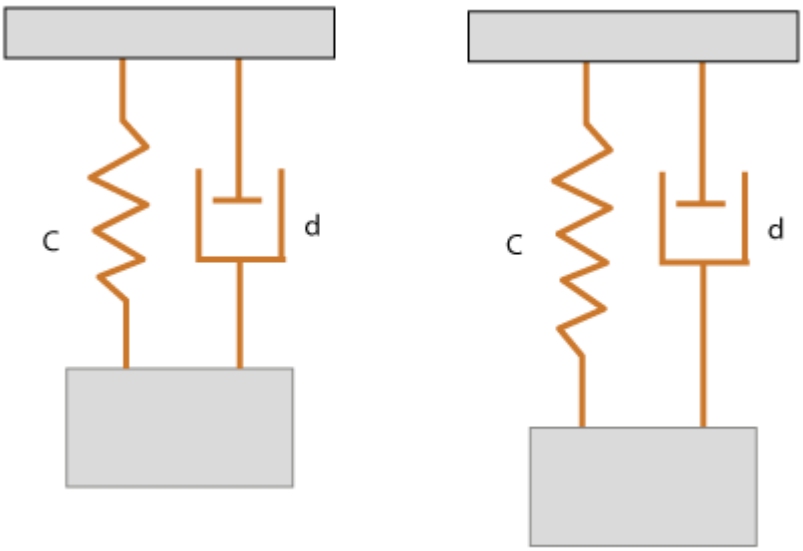


质量弹簧阻尼振荡器仿真示例

在本节中，我们将为由质量、弹簧和阻尼器组成的简单振荡器创建 SysML 参数模型，然后使用参数模拟来预测和绘制该机械系统的行为。最后，我们通过比较通过数据集提供不同参数值的两个振荡器来执行假设分析。

正在建模的系统

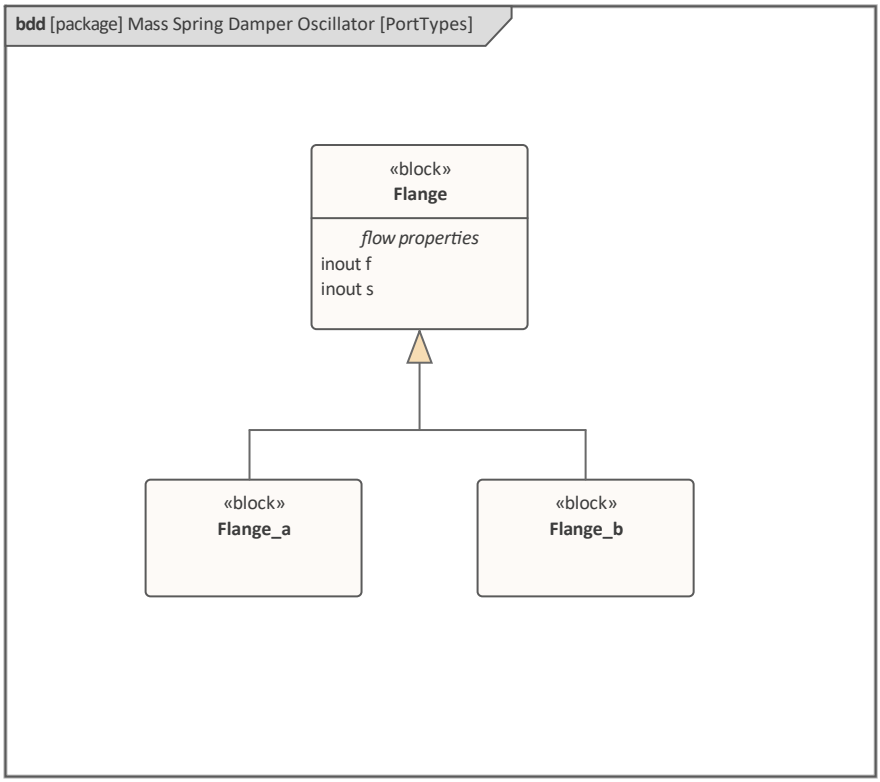
质量悬挂A弹簧和阻尼器上。此处显示的第一个状态表示时间=0 时的初始点，就在质量释放时。第二个状态代表当身体静止并且弹簧力与重力平衡时的最终点。

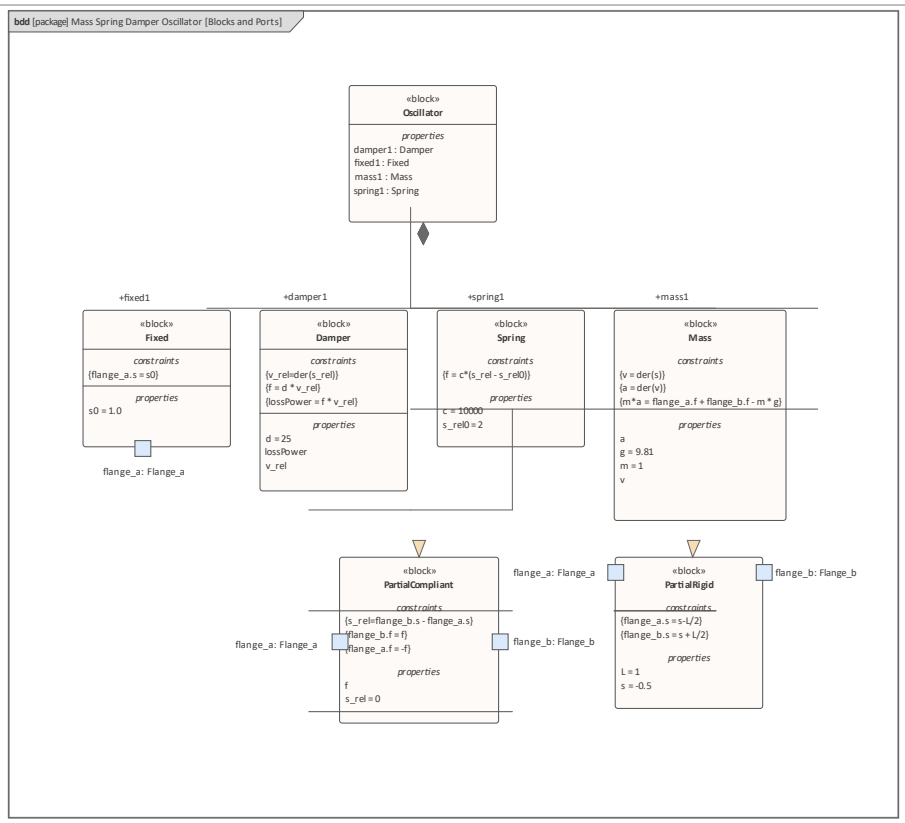


创建 SysML模型

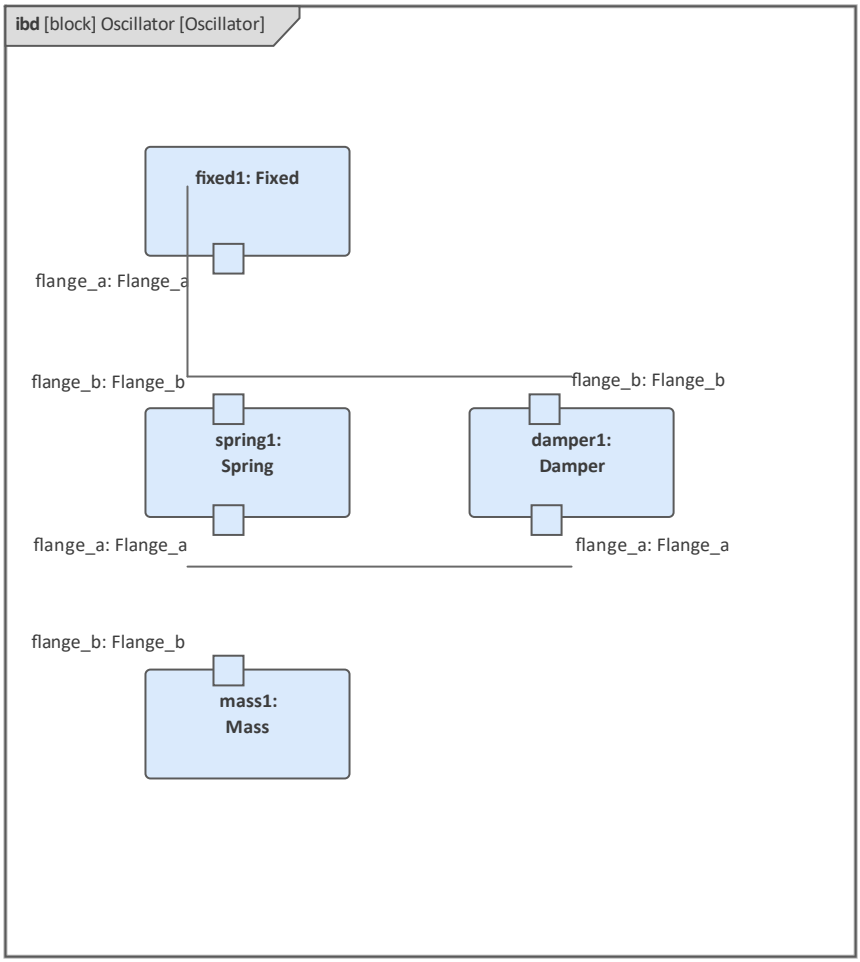
SysML 中的块模型有一个主要模块，即振荡器。振荡器有四个部分：固定顶板、弹簧、阻尼器和质量体。为这些部分中的每一个创建一个块。 Oscillator块的四个部分通过端口连接，代表机械法兰。

成分	描述
端口类型	用于 1D 过渡机械域中法兰的块 'Flange_a' 和 'Flange_b' 相同，但作用略有不同，有点类似于电气域中 PositivePin 和 NegativePin 的作用。力通过法兰传递。所以 flow 属性*Flange.f*的属性*isConserved*应该设置为True 。

	
块和端口	• 创建块 弹簧”、 阻尼器”、 质量”和 固定”以分别表示弹簧、阻尼器、质量体和天花板 • 创建一个块'PartialCompliant'，有两个端口（法兰），命名为'flange_a'和'flange_b'——它们分别是Flange_a和Flange_b类型； 'Spring' 和 'Damper' 块从 'PartialCompliant' 概括 • 创建一个具有两个端口（法兰）的块 PartialRigid”，命名为 flange_a”和“flange_b”——它们分别是 Flange_a 和 Flange_b 类型； 'Mass'块从'PartialRigid'概括 • 为天花板创建一个只有一个法兰的块'Fixed'，它的端口只有'flange_a'类型为 Flange_a



内部结构	为 "振荡器"创建内部块图 (IBD)。为固定天花板、弹簧、阻尼器和质量体添加属性，由相应的 Blocks 键入。用连接器连接端口。 • 将 "fixed1"的 "flange_a"连接到 "spring1"的 "flange_b" • 将 "damper1"的 "flange_b"连接到 "spring1"的 "flange_b" • 将 "damper1"的 "flange_a"连接到 "spring1"的 "flange_a" • 将 "spring1"的 "flange_a"连接到 "mass1"的 "flange_b"

	
约束	为简单起见，我们直接在块元素中定义约束；您可以选择定义约束块，在块中使用属性，并将它们的参数绑定到块的属性。

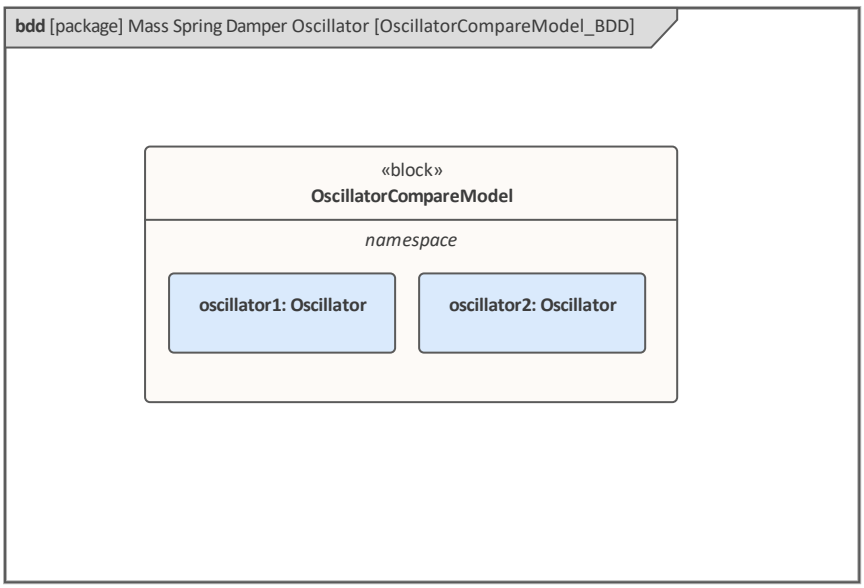
两个振荡器比较计划

在我们对模型进行建模后，我们想做一些假设分析。例如：

- 具有不同阻尼器的两个振荡器有什么区别？
- 如果没有阻尼器怎么办？
- 两个不同弹簧的振荡器有什么区别？
- 两个不同质量的振荡器有什么区别？

以下是创建比较模型的步骤：

- 创建一个名为 "OscillatorCompareModel"的块
- 为 "OscillatorCompareModel"创建两个属性，称为振荡器1 和振荡器*2*，并使用块振荡器键入它们



设置数据集并运行仿真

创建一个配置工件其分配给这个包。然后创建这些数据集：

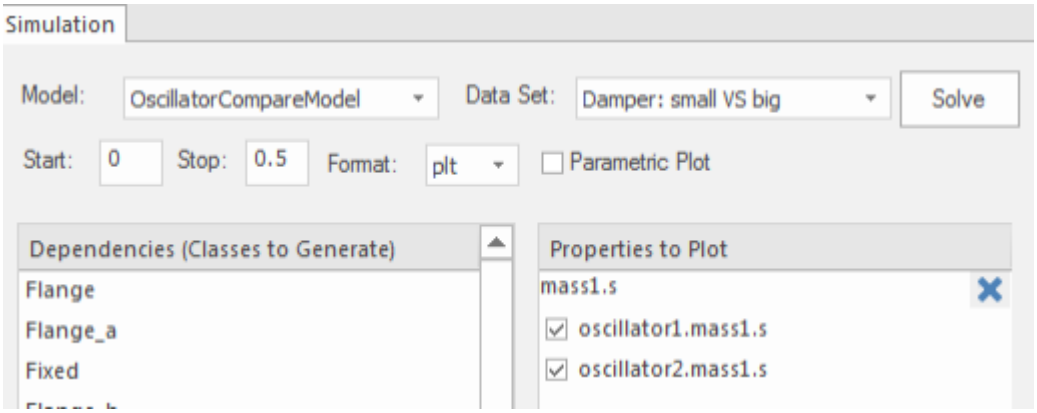
- 阻尼器：小与大
  为 'oscillator1.damper1.d' 提供值 10，为 'oscillator2.damper1.d' 提供较大的值 20
- 阻尼器：否 vs 是
  为 'oscillator1.damper1.d 提供值 0;（'oscillator2.damper1.d' 将使用默认值 25）
- 弹簧：小与大
  提供 'oscillator1.spring1.c' 的值 6000 和 'oscillator2.spring1.c' 的较大值 12000
- 质量：轻与重
  为 "oscillator1.mass1.m"提供值 0.5，为 "oscillator2.mass1.m"提供较大的值 2

配置的页面如下所示：

◢ OscillatorCompareModel	SysMLSimModel
▷ Part	
◢ Damper: small VS big	Click button to configure...
oscillator2.damper1.d	20
oscillator1.damper1.d	10
◢ Spring: small VS big	Click button to configure...
oscillator2.spring1.c	12000
oscillator1.spring1.c	6000
◢ Damper: no VS yes	Click button to configure...
oscillator1.damper1.d	0
◢ Mass: light VS Heavy	Click button to configure...
oscillator2.mass1.m	2
oscillator1.mass1.m	0.5

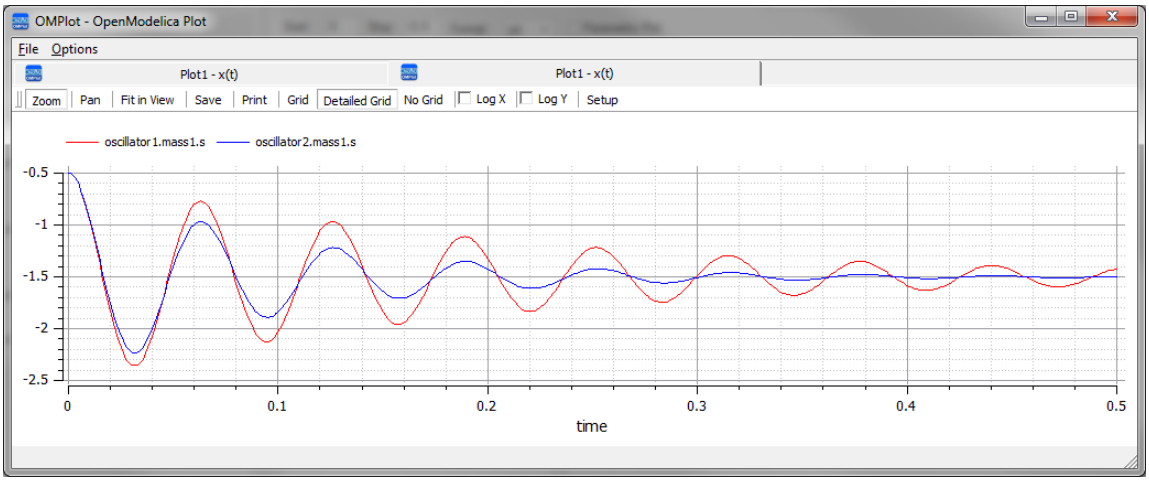
在 "仿真"页面上，选择 "OscillatorCompareModel"，绘制 "oscillator1.mass1.s"和 "oscillator2.mass1.s"，然后选择一个创建的数据集并运行仿真。

提示：如果绘图列表中的属性过多，您可以使用列表标题上的时间菜单切换过滤器栏，然后在本例中键入 “上下文”。

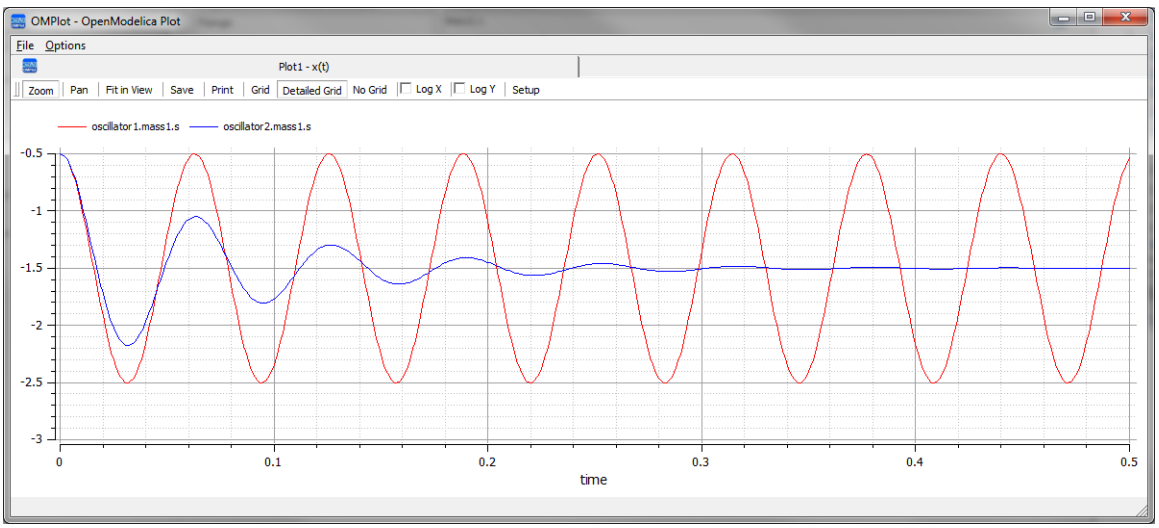


这些是模拟结果：

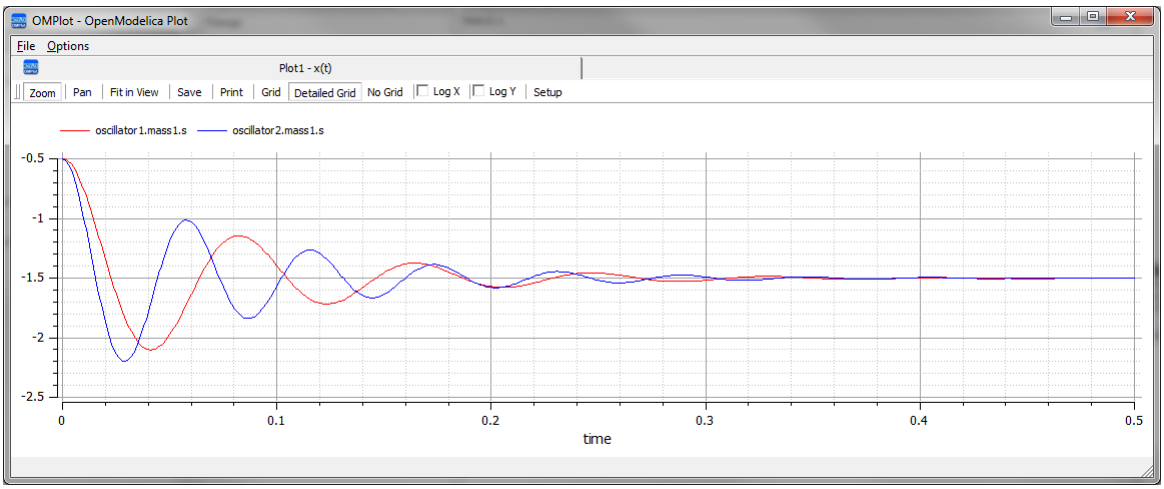
- 阻尼器，小与大：阻尼器越小，车身振动越大



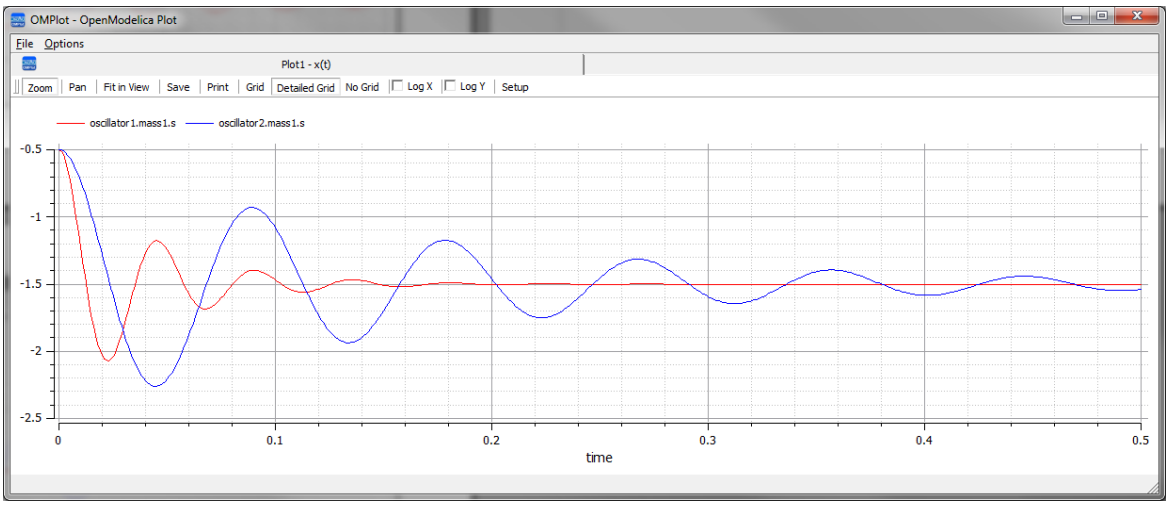
- 阻尼器，否 vs 是：振荡器在没有阻尼器的情况下永远不会停止



- 弹簧，小与大：“c”较　的弹簧会摆动得更慢



- 质量，轻与重：质量较小的object会更快地振动并更快地调节



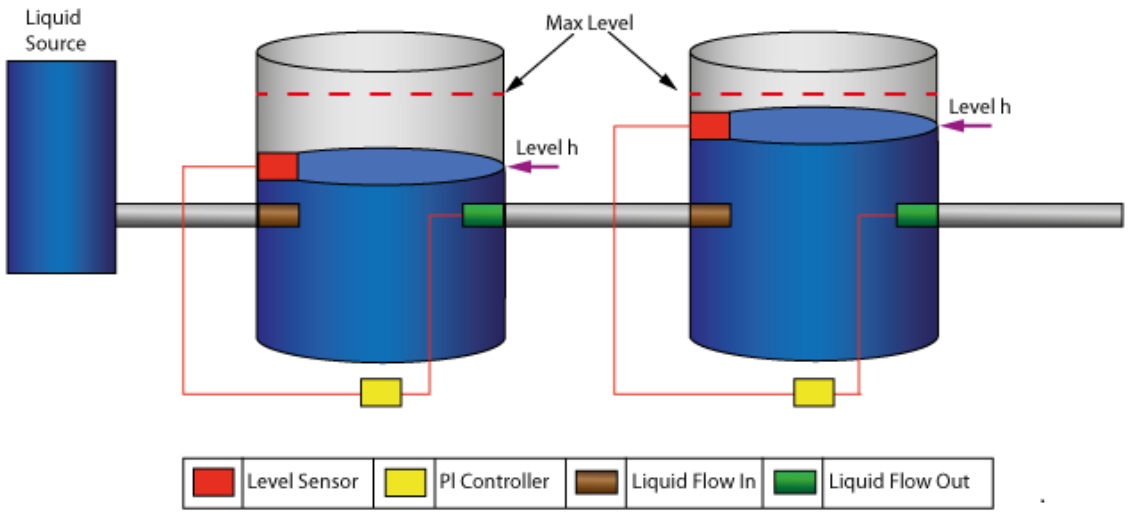
水箱压力调节器

在本节中，我们将介绍为水箱压力调节器创建 SysML 参数模型，该模型由两个连接的水箱、一个源和两个控制器组成，每个控制器监控水位并控制阀门以调节系统.

我们将解释 SysML模型，创建它并设置 SysMLSim 配置。然后我们将使用运行仿真。

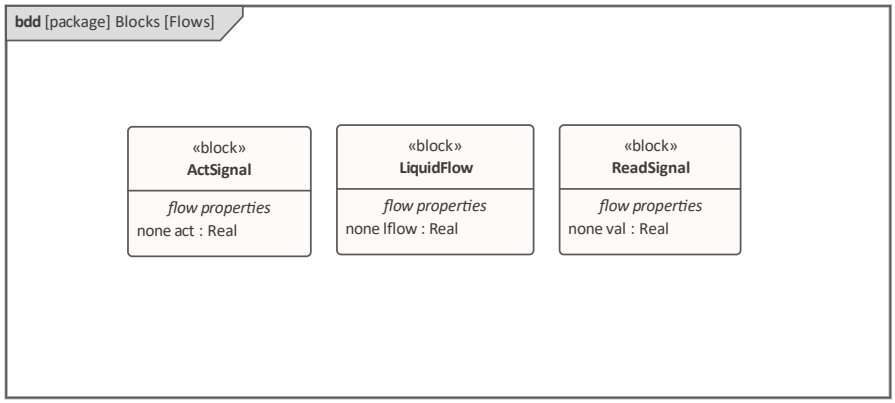
正在建模的系统

此图描绘了连接在一起的两个源，以及填充第一个水箱的水源。每个水箱都有一个与其相连的比例积分 (PI) 连续控制器，它将水箱中的水位调节到参考水位。当源向第一个水箱注水时，PI 连续控制器根据水箱的实际水位调节水箱的流出量。第一个水箱中的水流入第二个水箱，PI 连续控制器也试图对其进行调节。这是一个自然的、非特定领域的物理问题。



创建 SysML模型

部件	讨论
端口类型	坦克有四个端口，它们分别输入到这三个块中： • ReadSignal：读取液位；这有一个单位为 "m"的属性 "val" • ActSignal：用于设置阀门位置的执行器信号 • LiquidFlow：入口或出口处的液体流量；这有一个属性 "flow"，单位为 "m^3/s"



块定义图表

LiquidSource：进入水箱的水一定来自某个地方，因此我们在水箱系统中有一个液体源组件，属性*flowLevel*的单位为 ḿ³/s″。 A 端口″键入为 端口″。

水箱：水箱通过端口连接到控制器和液体源。

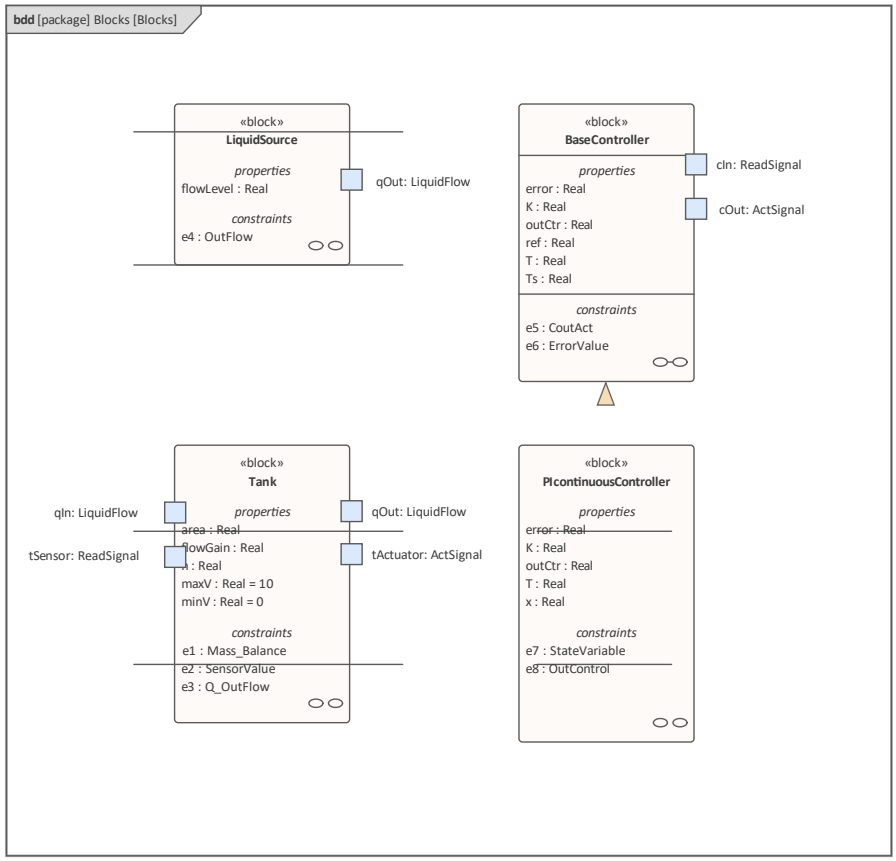
- 每个 Tank 有四个端口：
  - qIn：用于输入流
  - qOut：用于输出流
  - tSensor：用于提供液位测量
  - tActuator：用于设置阀门在出口处的位置
  坦克
- 属性：
  - 体积（单位='m³'）：罐的容量，参与质量平衡
  方程
  - h（单位='m'）：水位，参与质量平衡
  方程;它的值由传感器读取
  - flowGain（单位='m³/s'）：输出流量与阀门有关
  按流量增益定位
  - minV, maxV：输出阀流量的限制

BaseController：这个块可以是 PI Continuous控制器和 PI Discrete控制器的父级或祖先。

- 端口：
  - cIn：输入传感器电平
  -控件：控制执行器
- 属性：
  - Ts（单位='s'）：离散样本之间的时间周期（未使用
  在这个例子中）
  - K ：增益因子
  - T (unit = 's'): 控制器的时间常数
  - 参考：参考水平
  - 误差：参考水平与实际水平之间的差异
  水位，从传感器获得
  - outCtr：到执行器的控制信号，用于控制阀门
  位置

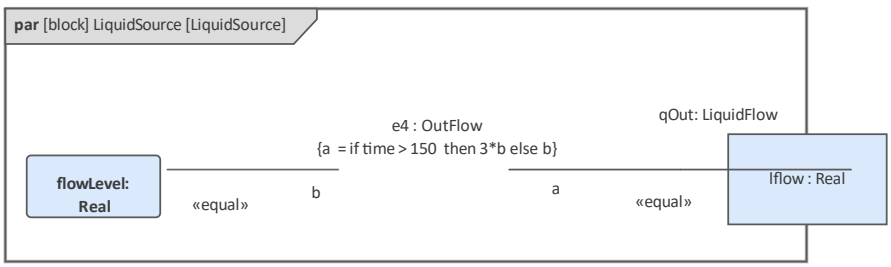
PIcontinuousController：从 BaseController 专门化

- 属性：
  -x：控制器状态变量



约束块

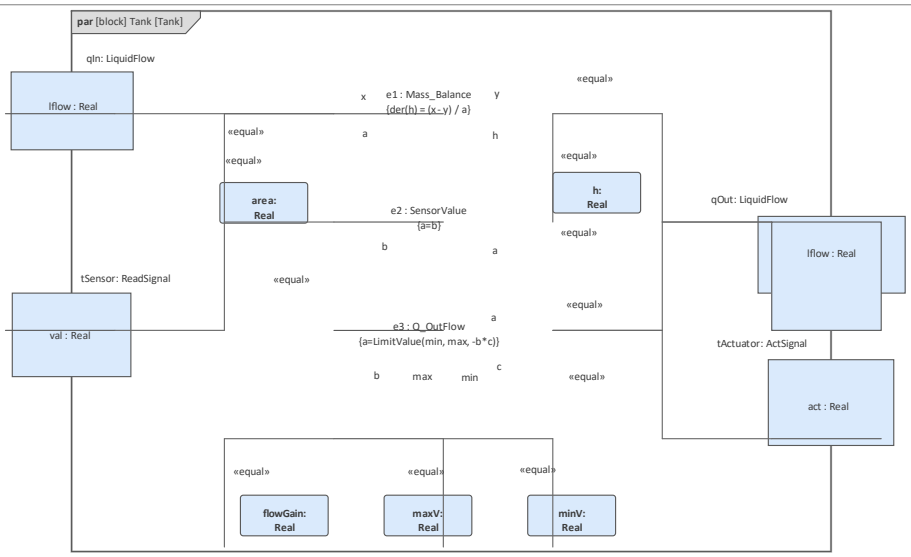
流量在时间=150 时急剧增加至先前流量水平的三倍，这产生了一个有趣的控制问题，罐的控制器必须处理。



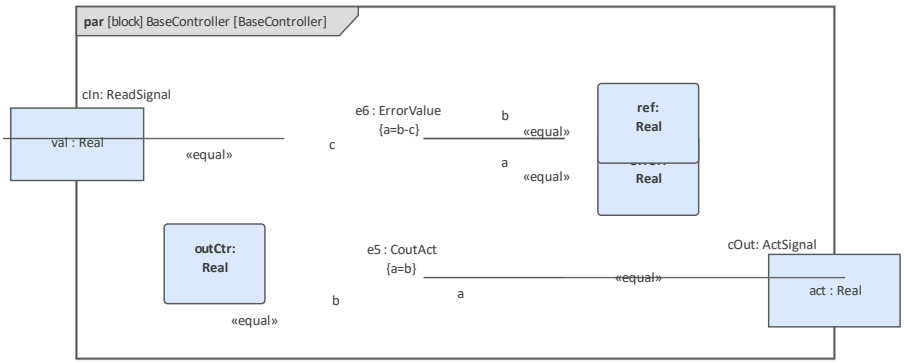
调节罐性能的中心方程是质量平衡方程。

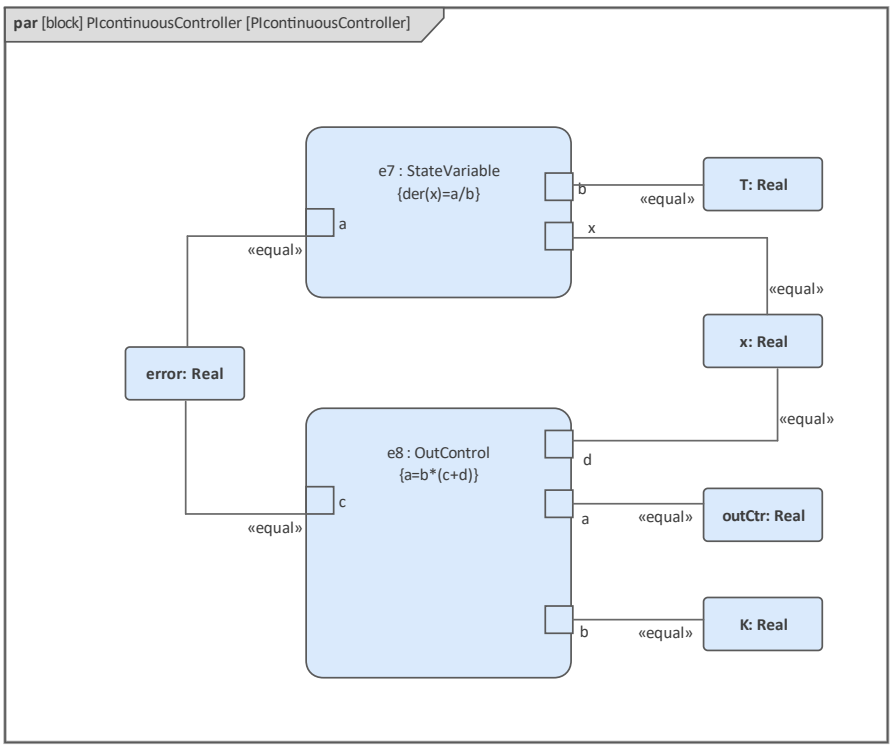
输出流量通过 "flowGain"参数 阀门位置相关。

传感器只是读取水箱的液位。

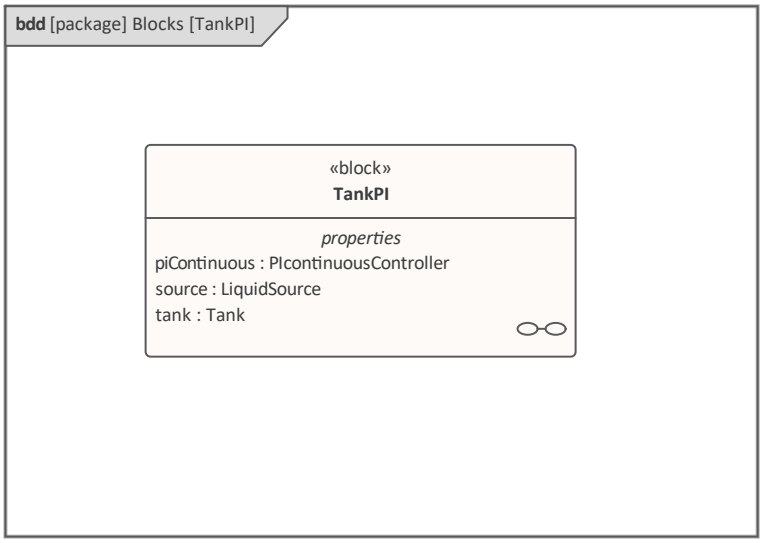


这些图中说明了为 "BaseController"和 "PIcontinuousController"定义的约束。

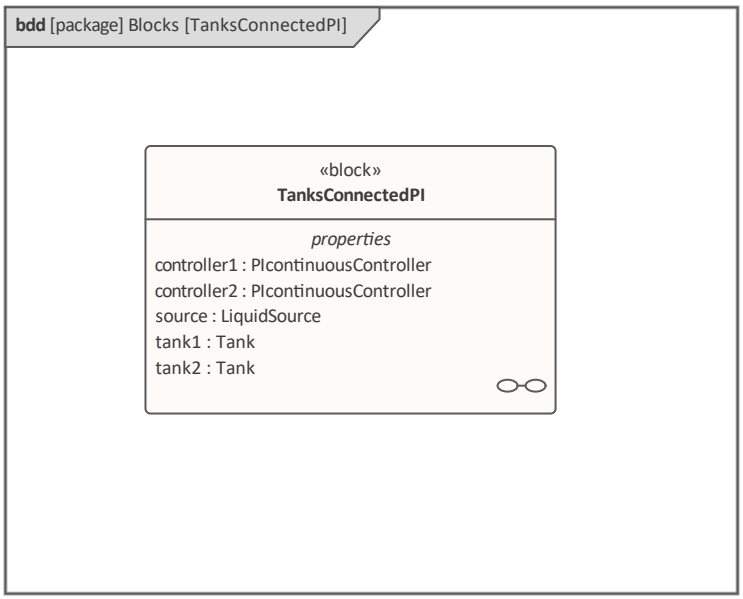




内部块图表	这是具有单个水箱的系统的内部块图。



这是具有两个连接罐的系统的内部块图。



运行仿真

由于*TankPI*和*TanksConnectedPI*被定义为 "SysMLSimModel"，因此它们将列在 "仿真"页面的 "模型"组合框中。

选择*TanksConnectedPI* ，并观察这些 GUI 发生的变化：

- "数据集"组合框：将填充*TanksConnectedPI*中定义的所有数据集
- 'Dependencies' 列表：将自动收集 TanksConnectedPI 直接或间接引用的所有 Blocks、约束、 *SimFunctions*和 ValueTypes（这些元素将生成为 OpenModelica 代码）
- '属性Plot'：将收集一长串'叶子'变量属性（即它们没有属性）；您可以选择一个或几个进行模拟，属性将显示在图例中

工件和配置

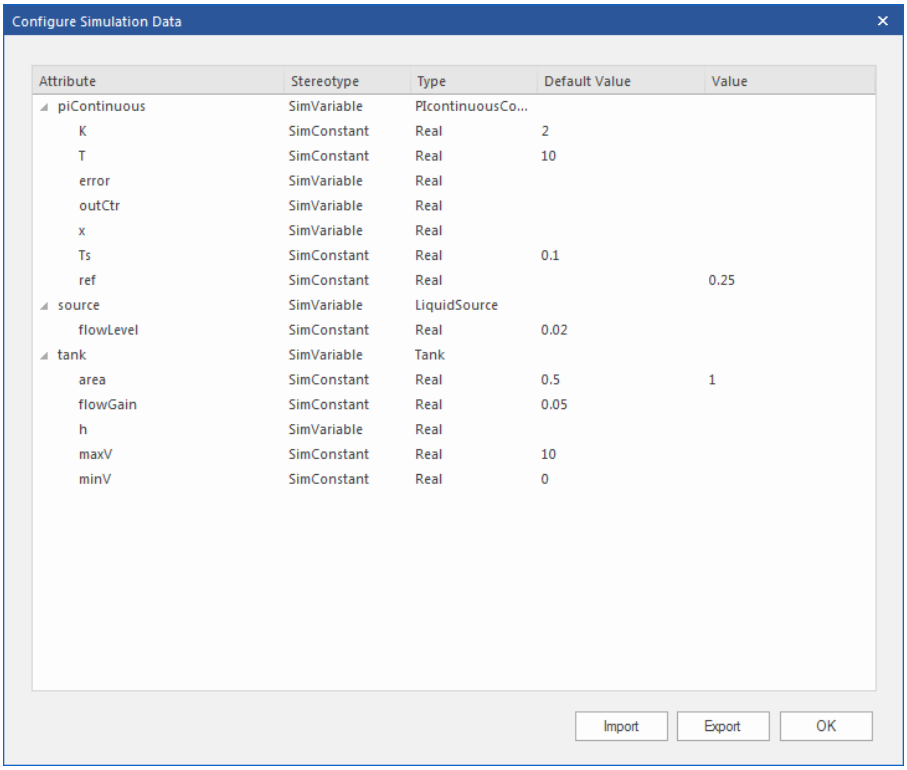
选择 仿真>系统行为> Modelica/Simulink >配置管理器”
包中的元素将被加载到配置管理器中。
配置这些块及其属性，如本表所示。
注记：未配置为 属性”的属性默认为 SimVariable”。

块	属性
液体源	配置为 配置”。 属性配置： • flowLevel：设置为 SimConstant”
坦克	配置为 配置”。 属性配置： • 区域：设置为 SimConstant” • flowGain：设置为 SimConstant” • maxV：设置为 SimConstant” • minV：设置为 SimConstant”
基本控制器	配置为 配置”。 属性配置： • K ：设置为 SimConstant” • T ：设置为 SimConstant” • Ts：设置为 SimConstant” • 参考：设置为 SimConstant”
PI连续控制器	配置为 配置”。
坦克PI	配置为 配置”。
TanksConnectedPI	配置为 配置”。

设置数据集

右键单击每个元素，选择 创建仿真数据集”选项，然后配置数据集，如本表所示。

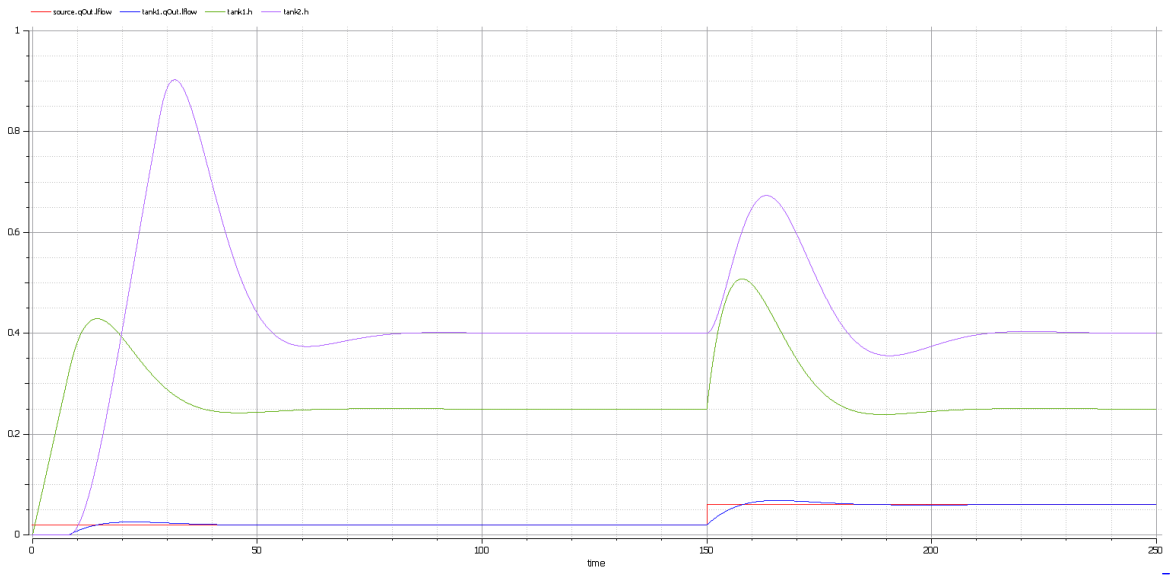
元素	数据集
液体源	流量等级：0.02
坦克	h.开始：0 流量增益：0.05 面积：0.5 最大V：10 最小伏特：0
基本控制器	T ：10

	K : 2 TS: 0. 1
PI连续控制器	无需配置。 默认情况下，特定块将使用超级块的默认数据集中配置的值。
坦克PI	这里有趣的是可以在 配置仿真数据”对话框中加载默认值。例如，我们在每个块元素上配置为默认数据集的值被加载为属性的默认值。单击每行上的图标可将属性的内部结构扩展到任意深度。  点击确定按钮，返回配置管理器。然后配置这些值： • tank.area: 1这会覆盖 Tank块数据集中定义的默认值 0.5 • piContinuous.ref：0.25
TanksConnectedPI	• 控制器1.ref：0.25 • 控制器2.ref：0.4

仿真分析1

选择这些变量并单击求解按钮。这个情节应该提示：

- 源.qOut.lflow
- tank1.qOut.lflow
- 坦克1.h
- 坦克2.h



以下是对结果的分析：

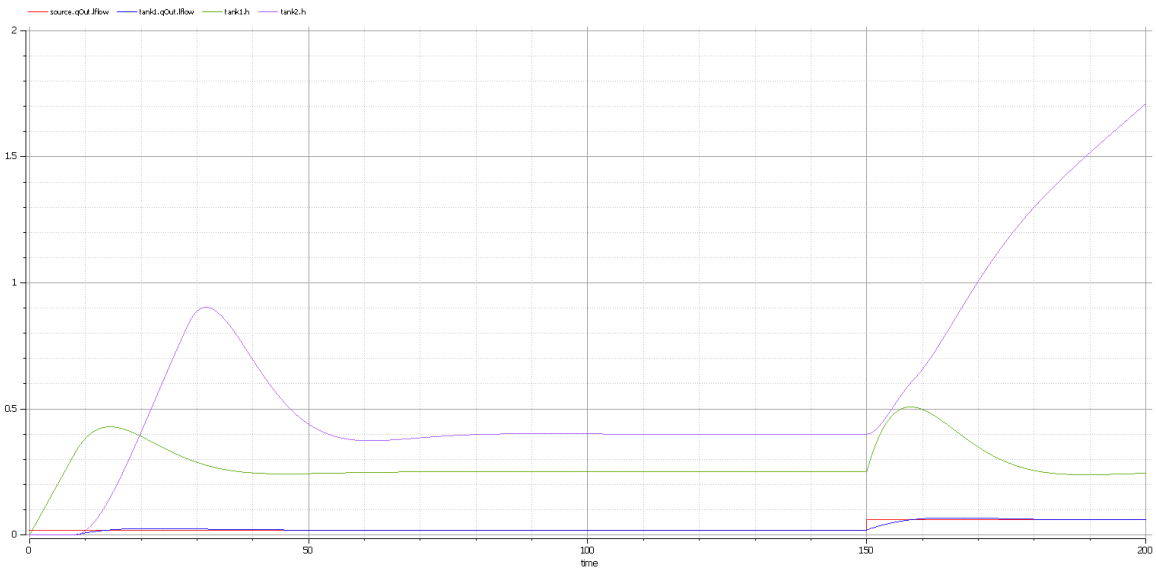
- 液体流量在时间=150 时急剧增加，达到 0.06 m^3/s，是之前流量 (0.02 m^3/s) 的三倍
- Tank1 调节高度 0.25，tank2 调节高度 0.4 符合预期（我们通过数据集设置参数值）
- 在模拟过程中，tank1 和 tank2 都进行了两次调节；首次调节流量0.02 m^3/s；第二次调节流量 0.06 m^3/s
- 在 tank1 有任何流量之前 Tank2 是空的

仿真分析2

在示例中，我们将坦克的属性 "minV"和 "maxV"分别设置为值 0 和 10。在现实世界中，10 m^3/s 的流量需要在水箱上安装一个非常大的阀门。

如果我们将 "maxV"的值更改为 0.05 m^3/s 会发生什么？基于之前的模型，我们可能会做出这些改变：

- 在 TanksConnectedPI 的现有 "DataSet_1"上，右键单击并选择 "复制数据集"，然后重命名为 "Tank2WithLimitValveSize"
- 单击按钮进行配置，展开 "tank2"并在 "属性"的 "值"列中键入 "0.05"
- 在 "仿真"页面上选择 "Tank2WithLimitValveSize"并绘制属性
- 单击求解按钮执行模拟



以下是对结果的分析：

- 我们的更改仅适用于 tank2； tank1 可以像以前一样在 0.02 m^3/s 和 0.06 m^3/s 上调节
- 当源流量为0.02 m^3/s时，tank2可以像以前一样调节
- 然而，当源流量增加到 0.06 m^3/s 时，阀门太小，无法让流出流量与流入流量匹配；唯一的结果是tank2的水位升高
- 然后由用户来解决这个问题；例如，换一个更大的阀门，减少源流量或制作一个额外的阀门

总之，此示例显示了如何通过复制现有 DataSet 来调整参数值。

Simscape集成

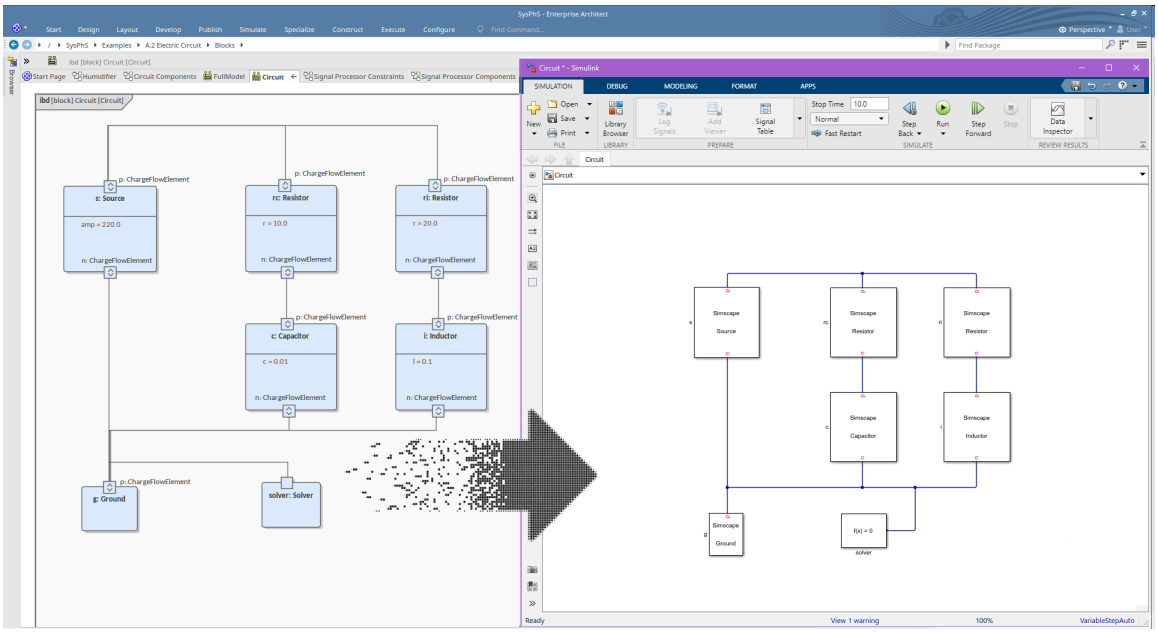
Simscape 用于通过物理流对物理系统模型，将 SysML内部块图转换为 Simulink模型，从而在许多不同的物理领域中开放 Simscape 的大量库块。

Enterprise Architect可以将 SysML内部块图转换为 MATLAB 的 Simscape，这是 Simulink 的一个可选扩展，允许对物理系统进行建模并指示 MATLAB 模拟和绘制所请求的输出。块表示物理对象，流表示物质或能量的物理流动（例如液体、电流、气体、通量、力/扭矩、热流等；例如，水从一个水箱流到另一个水箱，或电流流过电阻器）。您可以使用内置的 SysPhS 模式访问大量预构建的 Simscape 库块，或者使用 SimulinkBlock 构造型创建对您自己的自定义库块的引用。

有关使用 Simulink 和 Simscape 的详细信息，请参阅 建模物理网络的基本原理" 链接。

物理流属性必须由块输入，并包括一个保守的 PhSVariable 和一个非保守的 PhSVariable。使用SysML 视角下可用的 SysPhS模型生成器模式，其中包括：

- SysPhS 物理交互元素
- SysPhS 信号流元素



需求

- 必须在配置SysML仿真窗口中勾选 使用Simscape"选项
- 带有 "inout"方向（或无）的端口将被假定为物理流
- 任何块的 "inout"端口都将生成为 Simscape - 如果它也有 "in"或 "out"的端口信号，这些将被设置为 "physicalscape"这些可以连接到输出 物理信号"或 Simulink 输入和输出。执行此操作所需的 Simulink 转换器将在生成过程中自动生成并插入到模型中。

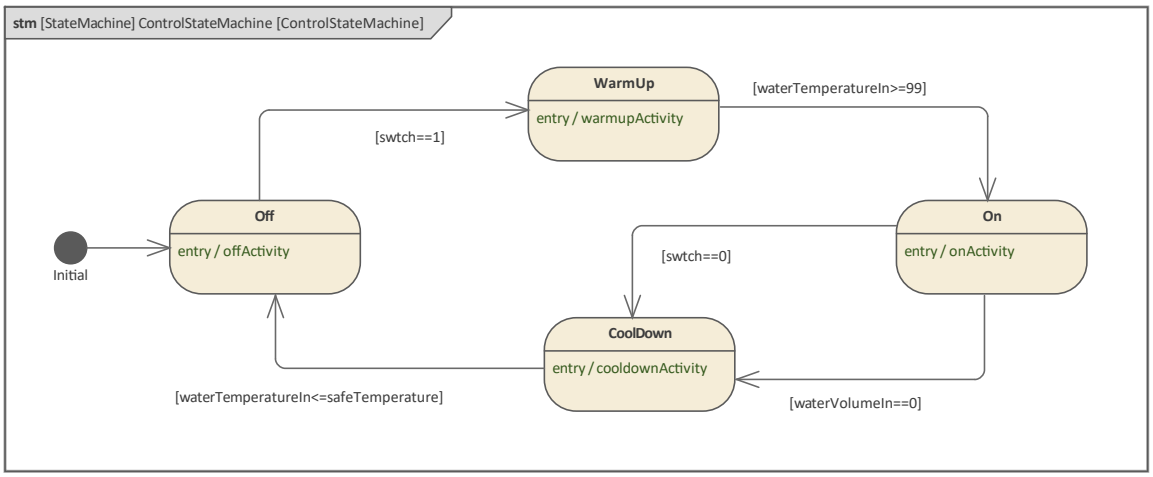
视频

Sparx Systems提供了一个使用 Simscape 生成 SysML仿真图的 YouTube 视频。看：

- SysML仿真绘图使用 Simscape

Simulink集成

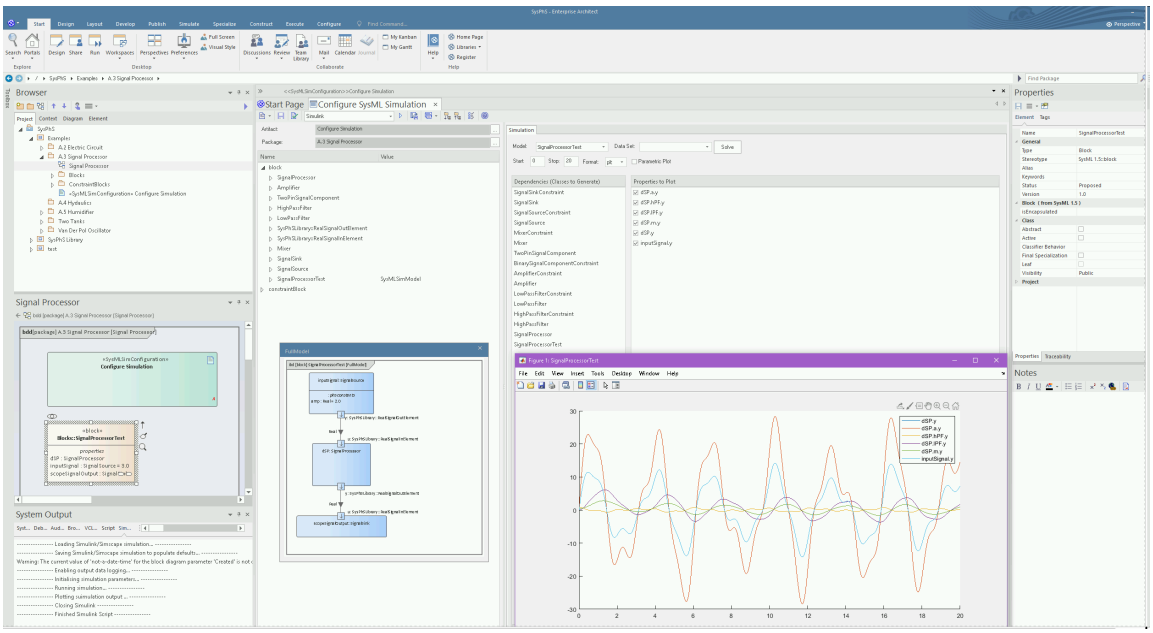
Enterprise Architect会将 SysML模型转换为 Simulink 格式并自动运行仿真，绘制所选变量的输出。生成的 Simulink 文件也可以直接在 Simulink 中打开，允许修改和微调仿真设置和输出功能。



鉴于 Simulink 使用信号流单向的块图方法， Enterprise Architect支持将模型导出到 Simulink 以模拟块之间的定向消息，生成模拟结果图表。

您可以直接通过Enterprise Architect模式拖放访问常见的内置 Simulink 库模块，或者您可以引用您自己的自定义构建模块以及 SysPhS 标准下的原型参数。

使用 Simulink 进行仿真是使用 OpenModelica 的替代方法，OpenModelica 也可以在Enterprise Architect中使用。



Simulink建模

对于可以导出到 Simulink 的建模图，您可以直接通过Enterprise Architect模式拖放访问常见的内置 Simulink 库模块，或者您可以参考您自己的自定义构建模块，并使用 SysPhS 标准下的原型参数。

Simulink 可用于模型信号流；也就是说，一个块块一条信息。这意味着，对于 Simulink 建模，所有端口都必须有一个定义为*in*或*out*的方向。端口之间的项目流必须匹配进/出方向。 （注记支持信号或物理系统的建模。）

需求

为了符合 Simscape 的单向要求：

- 端口将方向定义为 "进"或 "出"
- 端口的连接必须与它们的方向一致；例如，一个 "in"端口不能连接到另一个 "in"端口
- 约束参数将被视为变量，除非指定为构造型 "PhSConstant"（或在配置SysML仿真窗口中设置为 SimConstant）
- 约束方程必须在等号的左侧 (lhs) 有一个项，并且该项必须是输出或导数

视频

Sparx Systems提供了一段在 Simulink 中执行 SysML状态机仿真的 YouTube 视频。看：

- Simulink中的SysML状态机仿真

注记

- Simulink 只允许绘制输出端口；将来可能会有增强功能，以允许也绘制其他变量 -端口注记，输入与输出端口本质上相同，连接到它

学到更多

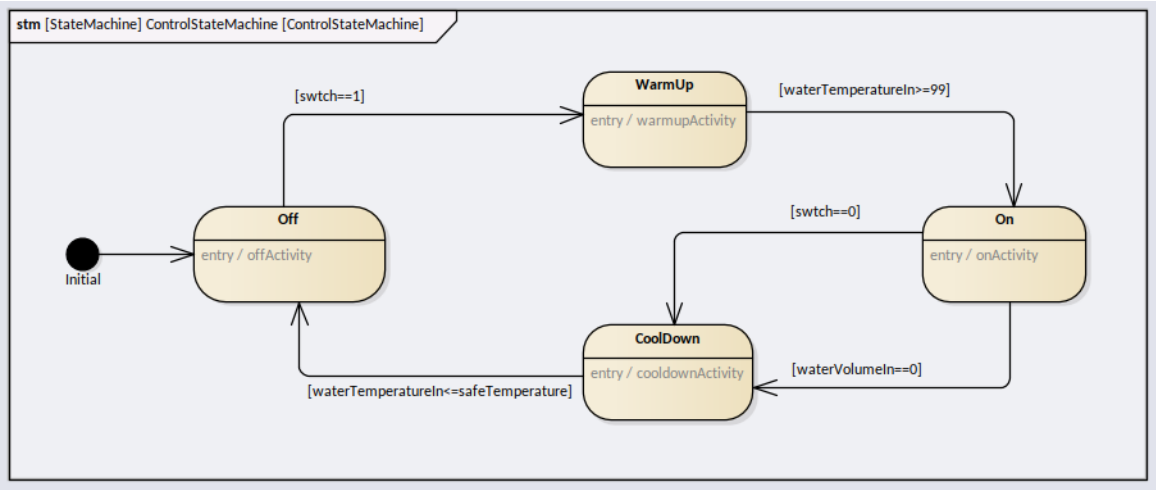
- Creating Simulink and Simscape Specific Blocks

状态流集成

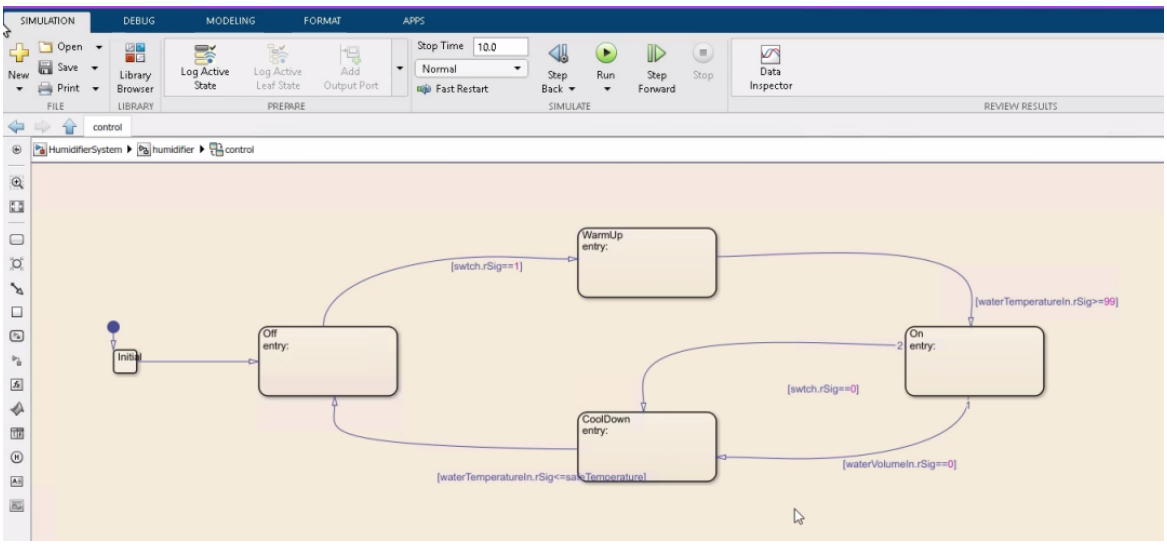
使用Enterprise Architect的 SysML 仿真时A重要特征是能够生成 MATLAB状态流图以在运行下运行，从而允许您使用在Enterprise Architect中建模并转换为状态流图的状态机来指导您的 SysML 仿真。

状态流图使用与 OMG状态机标准非常等效的元素和连接器，例如状态和转换。 Enterprise Architect支持所有状态流特征；但是，状态流仅使用 OMG状态机标准的一个子集。

这是Enterprise Architect中的状态机图示例：



这是上图中Enterprise Architect生成的代码如何呈现为状态流图的示例：



支持元素类型

可以转换为 StateFlow 对象的 SysML元素类型有：

- 状态
- 最初的
- 终点
- 连接点
- 选择（转换为状态流连接点）
- 历史

支持连接器类型

关键的 SysML状态机连接器类型确实有直接转换为 Simulink 连接器类型'转移
'。

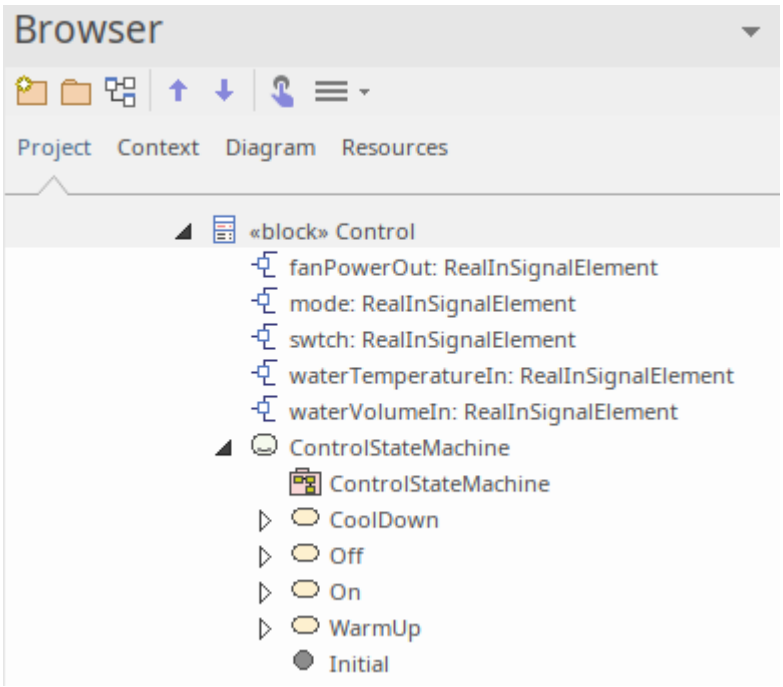
建模

在创建旨在呈现为状态流图的状态机图时，您有多种选择。这些包括可以使用哪些元素和连接器、状态机的放置、可以在模型中定义代码的位置以及可以使用什么格式的代码。

状态机放置

在创建要生成为状态流图的模型时，必须将状态机图放在一个SysML块下，并有一个子状态机元素，在该状态机元素下有一个状态机图。

例如本例中的 控件”有一个子状态机名称为 块”，它有一个子状态机图 “ControlStateMachine”：



注记该块必须仅包含状态机图；它不应包含其他图表。

定义代码

在状态机模型中放置脚本有多种选择。这样做的关键考虑因素是：脚本的格式是什么，它可以放在哪里？

代码格式

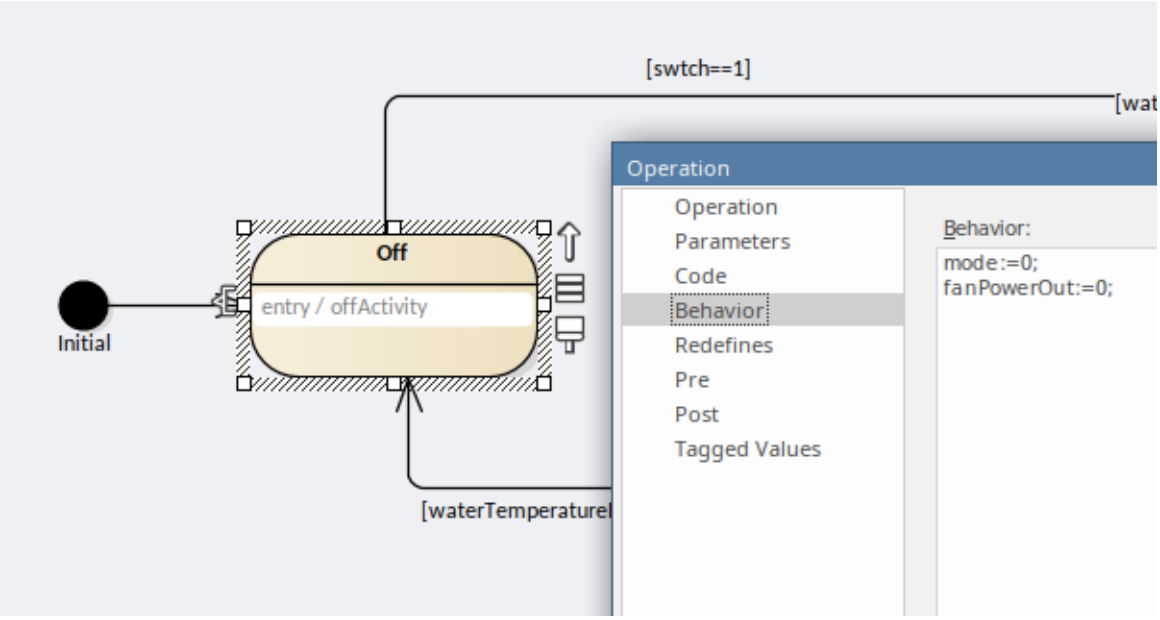
当您编写任何有效的 SysML 代码时，它将被转换为有效的状态流代码。 SysPhS 规范将 Modelica 数学语言定义为 标准”，这是Enterprise Architect接受的状态机图中的代码片段，并将这些转换为 MATLAB/状态流等效项。有关详细信息，请参阅*OMG SysPhS* 规范的第 10 节。

代码放置

在状态内部并且符合UML规范，状态流支持三个标准操作：

- 入口
- 出口
- 做

其中每一个都可以包含代码，这些代码在状态的行为下的操作中定义。例如，这是一个条目的代码片段：



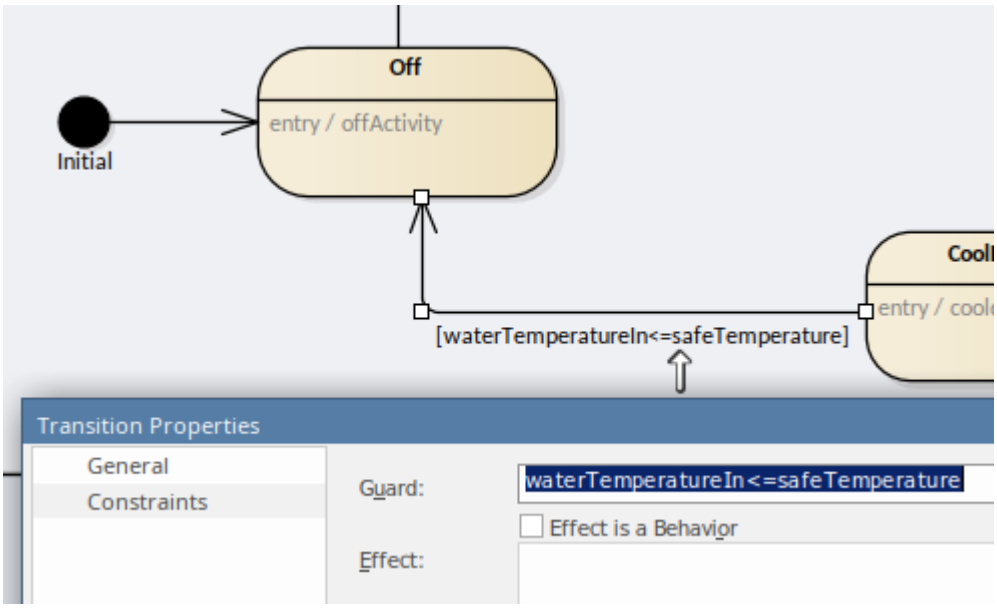
在转换中，使用代码有三个关键选项：

- 转移
  守卫条件
- 转移
  影响
- 触发器

例如，这是一个转移

中定义的条件语句转移

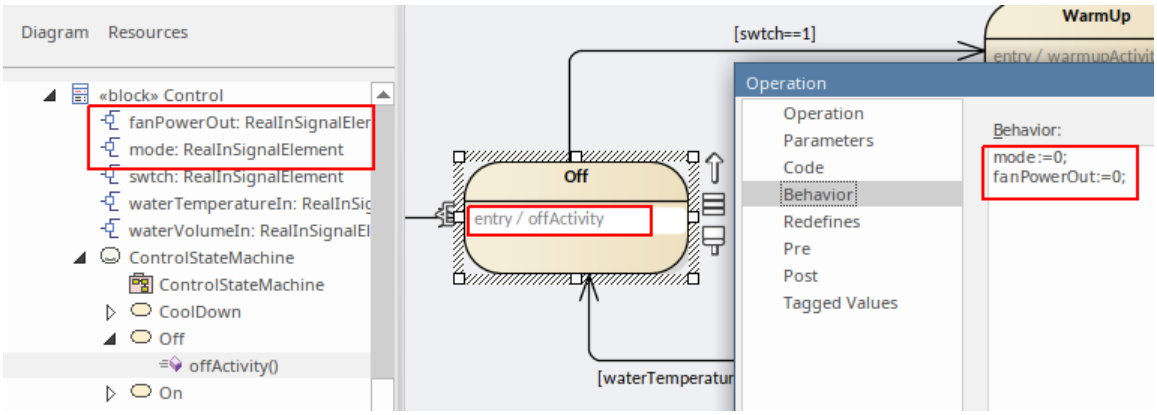
守卫条件：



使用块属性

对于代码，可以在脚本中引用常量、端口等属性块。在端口的情况下，输入到块的详细信息是使用输入端口名称派生的，并且可以将类似的值分配给与输出端口同名的变量。

例如：



注记

这些 SysML状态机特征在状态流中不可用：

- 深历史状态
- 具有要在所有默认条目上使用的传出转换（即初始状态）的历史状态
- 分叉/汇合
- 同步
- 连接点（只有选择-连接点才会被选择）
- 入口/出口点
- 引用其他状态机；状态流可以做子状态机但不能再次引用
- 每个 SysML块只有一个状态机

OpenModelica集成

OpenModelica 是一个基于 Modelica 建模语言的免费开源环境，用于建模、模拟、优化和分析复杂的动态系统。Enterprise Architect Archit 与 OpenModelica 集成，并支持在 SysPhS 标准（交互扩展用于物理行为和信号流仿真）下使用它来定义常量和变量，用于在 SysML 块中定义常量和变量，而不是在仿真仿真中这为定义和共享仿真提供了一种更简单的基于模型的方法。

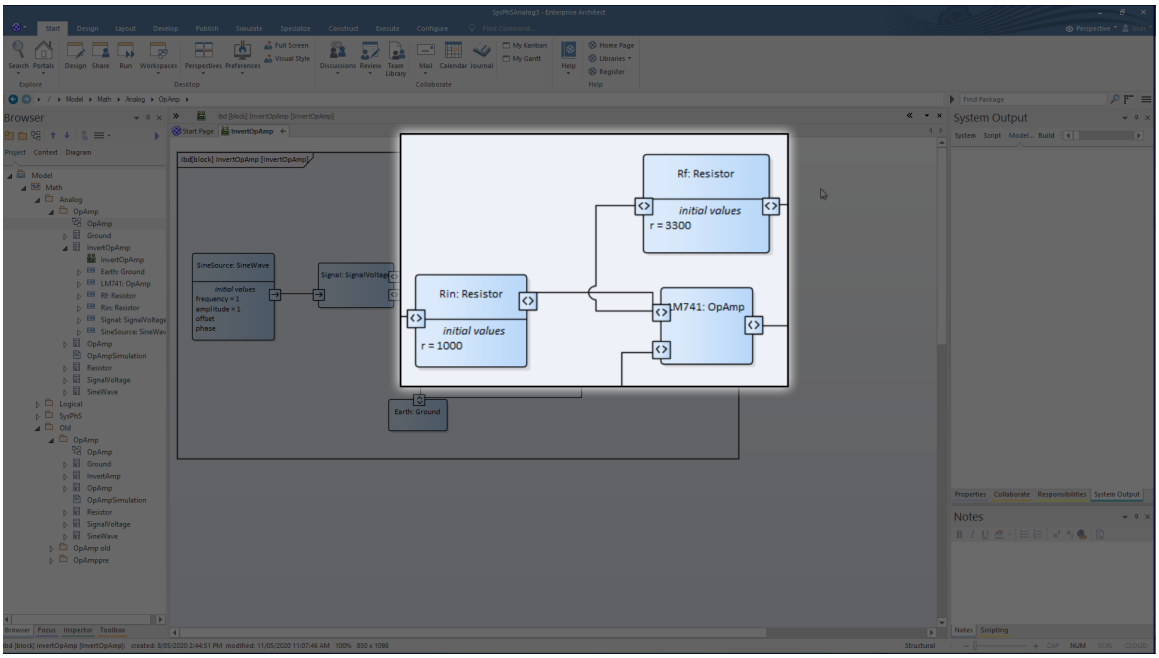
您还可以在 OpenModelica 连接编辑器 OMEdit 中的Enterprise Architect中显示模型中的 SysML块图，其中显示块的别名和注记。

您可以使用准备在 OpenModelica 中模拟的新 SysPhS 模式动态创建块，引用现有的 OpenModelica 库块或自定义用户定义的块。使用最新的 OpenModelica 代码生成，您可以在兼容的 OpenModelica 客户端（例如 OMEdit）中查看您的 SysML 组件，以及模拟绘图。

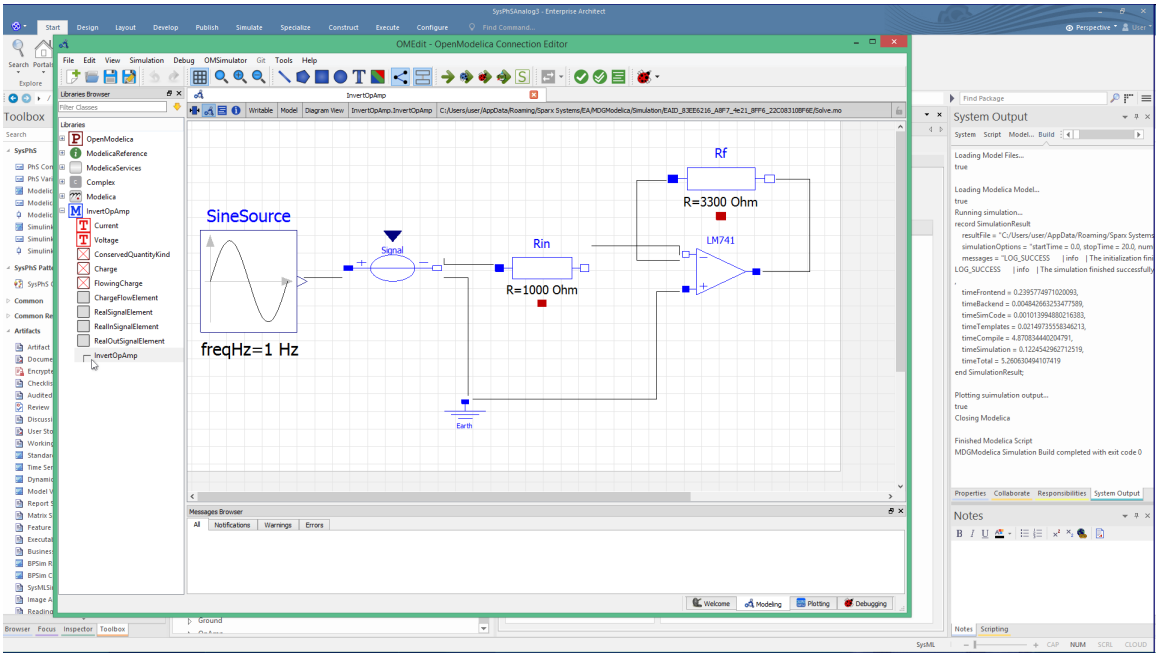
有关安装帮助并将Enterprise Architect连接到它的详细信息，请参阅安装Enterprise Architect平台的帮助主题。

使用 OpenModelica 是使用 MATLAB Simulink 在Enterprise Architect中执行参数模型仿真的替代方法。无论哪种情况，您都可以使用 SysPhS 标准配置您的模型，该标准定义了如何在 SysML模型与 OpenModelica模型或 Simulink/Simscape模型之间进行转换。

这是使用 SysPhS 特定的 SysML 部件定义的组件示例：



组件如 SysPhS模型生成的 OpenModelica 图表所示：



安装

平台	细节
窗口	如果Enterprise Architect安装在窗口平台上，请参阅窗口帮助主题窗口的 *OpenModelica* 。
Linux	如果Enterprise Architect安装在 Linux 平台上，请参阅 Linux 上的 *OpenModelica*帮助主题。

应用

使用 OpenModelica 库	有关引用 OpenModelica 库中可用资源的详细信息。
使用数据集进行模型分析	使用仿真配置，可以将一个块设置为针对它定义多个数据集。这允许使用相同的 SysML模型进行可重复的模拟变化。
故障排除仿真	本主题描述了使用 OpenModelica（或 MATLAB Simulink）进行仿真时可能出现的问题。

窗口上的 OpenModelica

安装 OpenModelica for Enterprise Architect在窗口平台上运行时，首先安装 OpenModelica 应用程序，然后在Enterprise Architect中配置设置以访问 OpenModelica。

安装 OpenModelica

节	行动
1	从以下位置下载 OpenModelica安装程序： https
2	双击 OpenModelica 安装程序并按照 向导"说明进行操作。 我们建议您接受默认安装路径。
3	选择您可以找到可执行文件 omc.exe。 例如：C:\OpenModelica1.9.2\bin\omc.exe

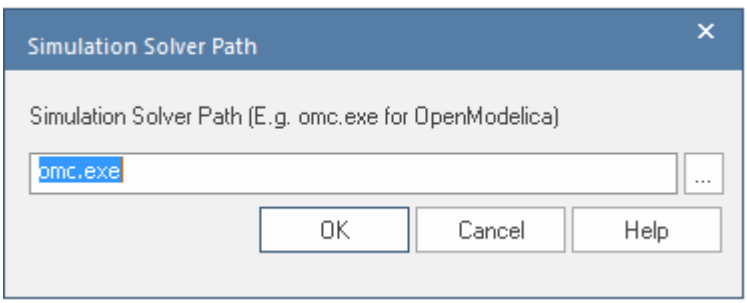
访问

使用这些访问路径中的任何一个来显示 仿真求解器路径"对话框，以配置求解器。

方法	选择
功能区	仿真>系统行为> Modelica/Simulink >配置管理器 > >配置仿真求解器
其它	使用工件构造型双击一个解决方案 > >配置仿真求解器

配置求解器

对于窗口， 仿真求解器路径"对话框如下所示：



类型输入或浏览到要在Enterprise Architect中使用的 OpenModelica Solver 的路径。

Linux 上的 OpenModelica

如果Enterprise Architect安装在 Linux 上，则必须与安装在同一平台上的 OpenModelica 一起操作。 OpenModelica Linux 安装是针对 Debian 和 Ubuntu 公开记录的；但是，它也可以安装在 Linux Mint 下。

本帮助主题提供以下方面的指导：

1. OpenModelica 的安装：
   - Linux Debian / Ubuntu
   - Linux薄荷
2. 配置Enterprise Architect以访问 OpenModelica。

Linux Debian / Ubuntu

要在 Linux Debian / Ubuntu 系统上安装 OpenModelica，请参考以下 URL：

https

这提供了 Debian / Ubuntu包的说明。

在终端上运行这些脚本。

节	行动
1	要将 OpenModelica 添加到您的附加存储库列表中： 用于 deb deb-src 中的 deb；做 echo "$deb http://build.openmodelica.org/apt `lsb_release -cs` nightly";完成 | sudo tee /etc/apt/sources.list.d/openmodelica.list
2	导入用于签署版本的 GPG 密钥： wget -q http://build.openmodelica.org/apt/openmodelica.asc -O- | sudo apt-key 添加 -
3	更新并安装 OpenModelica： sudo apt-get update sudo apt-get install openmodelica sudo apt-get install omlib-.* # 安装可选的 Modelica 库（大部分没有用 OpenModelica 测试过）
4	要检查此安装，请确保您可以找到文件 /usr/bin/omc，例如，在终端上执行以下命令： • ~ $ /usr/bin/omc --version 如果命令返回类似于以下的string ，则安装成功： • OpenModelica 1 .13.0~dev-1322-g53a43cf

Linux 薄荷糖

要在 Linux Mint 上安装 OpenModelica，您首先要为 Ubuntu 执行安装，然后修改 Linux Mint 代码名称以匹配 Ubuntu 代码名称。

这是 Linux Mint 代号到 Ubuntu 代号的映射列表（将在后面的步骤中使用）：

- Linux Mint 17.3 (Rosa) = Ubuntu 14.04 (Trusty): **rosa** = **trusty**

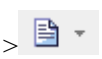
- Linux Mint 18 (Sarah) = Ubuntu 16.04 (Xenial): **sarah** = **xenial**
- Linux Mint 18. 1 (Serena) = Ubuntu 16.04 (Xenial)： **serena** = **xenial**
- Linux Mint 18.2 (Sonya) = Ubuntu 16.04 (Xenial): **sonya** = **xenial**
- Linux Mint 18.3 (Sylvia) = Ubuntu 16.04 (Xenial)： **sylvia** = **xenial**
- Linux Mint 19 (Tara) = Ubuntu 18.04 (Bionic)： **tara** =仿生

单击此处获取 Linux Mint 历史的完整列表以及与 Ubuntu 的映射。

节	行动
1	在终端中运行此脚本： 用于 deb deb-src 中的 deb；做 echo "$deb http://build.openmodelica.org/apt `lsb_release -cs` nightly";完成 | sudo tee /etc/apt/sources.list.d/openmodelica.list
2	要在 Linux Mint 中更改存储库 URL： • 在 Linux Mint 主屏幕上选择：'菜单 |搜索栏 |软件来源（输入密码）|其他存储库 |选择'Openmodelica' |编辑网址' • 将 Linux Mint 名称（例如， *rosa* ）更改为对应的 Ubuntu 名称（例如， *trusty* ），如本表顶部的列表中所示；那是： deb http://build.openmodelica.org/apt **rosa** nightly deb http://build.openmodelica.org/apt**值得信赖**的每晚 • 点击确定按钮
3	• 选择 Ópenmodelica（来源）”|修改网址 • 根据表顶部的列表更改Linux Mint 名称 例如，将 Linux Mint 名称*rosa*更改为对应的 Ubuntu 名称*trusty* • 点击确定按钮
4	要更新和安装运行，请在终端中运行这些脚本： sudo apt-get update sudo apt-get install openmodelica sudo apt-get install omlib-.* # 安装可选的 Modelica 库（大部分没有用 OpenModelica 测试过）

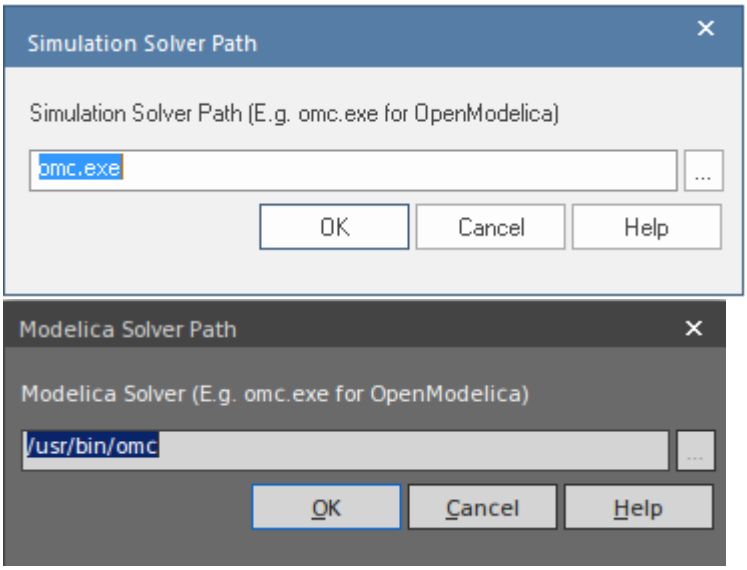
访问

使用这些访问路径中的任何一个来显示 仿真求解器路径”对话框，以配置求解器。

方法	选择
功能区	仿真>系统行为> Modelica/Simulink >配置管理器 >  >配置仿真求解器
其它	使用工件构造型双击一个解决方案 >  >配置仿真求解器

配置求解器

仿真求解器路径”对话框如下所示：

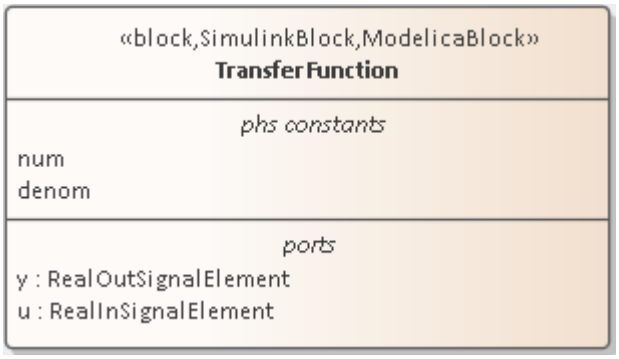


类型输入或浏览要使用的求解器的路径。

SysPhS仿真

交互扩展用于定义物理和信号仿真仿真规范（Ph），是物件管理组 (OMG) 规范的扩展，旨在为一致的模型提供通用建模平台。这些模型可以转换为两个关键仿真平台之一，Modelica 和 MATLAB 的 Simulink/Simscape。

OMG SysPhS 原型帮助您在模型本身而不是在模拟配置规范中定义模型模拟的特征。它们在浏览器窗口和属性窗口中以及在带有属性类型和初始值的特定元素隔间的图表中提供了更大的object或属性类型的可见性。



该标准在Enterprise Architect中由 OMG SysPhS配置文件表示，以及：

- SysPhS 用于信号流和物理交互的元素库（在 SysPhS 标准下执行模拟所必需的）
- 专门A工具箱页面
- 广泛A组件元素模式，可从中生成常见的仿真元素，例如电子、逻辑和流体组件；模式引用 OpenModelica 或 Simulink 标准库中的库模块
- 特征用于使用 Modelica 或 MATLAB 的 Simulink、Simscape 和状态流进行仿真绘图。

SysPhS特征

特征	描述
引用 SysPhS 库	使用 SysPhS 的关键资源是 SysPhS仿真库，其中包括您必须在模型中引用的可重用资源。
工具箱	图表工具箱的图表页面包含 OpenModelica 和 MATLAB Simulink 的基本 SysML 元素。
SysPhS模式	SysPhS模式提供引用等效 MATLAB 和 Modelica 组件的预定义 SysPhS 模块。在使用 SysPhS 模型时，这些简单的块可以用作启动器。
SysPhs 组件	SysPhS 组件使您能够设置对 Modelica 和 Simulink 组件的引用。
仿真	您可以使用附加信息定义 IBD 或参数模型以驱动仿真，然后使用仿真配置在 Modelica、Simulink 或 Simscape 中生成模型，以生成结果图。
SysPhS 示例	有几个使用 SysPhS 设置模拟的例子。
为 SysPhys 更新 SysMLSim	您可以更新旧的模拟配置（ Enterprise Architect 15.2 之前），以反映 SysPhS 标准的使用。

选项

变量和常量的额外选项，例如 isContinuous 和 isConserved，自动设置为标记值，再次避免了在配置规范中定义它们的需要。这些选项在块本身和停靠的属性窗口中也可见。

视频

- 用于电路仿真的SysPhS模式
- 使用仿真和 Modelica 仿真数字电子设备

SysML参数仿真

Enterprise Architect提供与 OpenModelica 和 MATLAB Simulink 的集成，以支持对 SysML模型在不同情况下的行为方式进行快速而可靠的评估。

OpenModelica 库是提供许多有用类型、函数和模型的综合资源。在Enterprise Architect中创建 SysML 模型时，您可以参考这些库中可用的资源。

Enterprise Architect的 MATLAB 集成通过 MATLAB API 进行连接，允许您的Enterprise Architect仿真和其他脚本根据任何可用的 MATLAB 函数和表达式的值进行操作。您可以通过 Solver类调用 MATLAB，或将您的模型导出到 MATLAB Simulink、Simscape 和/或状态流。

SysML仿真特征

这些部分描述了定义参数模型、使用附加信息对模型进行注释以驱动模拟以及运行模拟以生成结果图的过程。

部分	描述
SysML 参数模型简介	SysML 参数模型支持关键系统参数的工程分析，包括评估关键指标，如性能、可靠性和其他物理特性。这些模型通过捕获基于复杂数学关系的可执行约束，将需求模型与系统设计模型相结合。参数图是专门的内部块图，可帮助建模者将行为和结构模型与工程分析模型（例如性能、可靠性和质量属性）结合起来。 有关 SysML 参数模型概念的更多信息，请参阅 OMG SysML 官方网站及其链接资源。
创建参数模型	概述开发用于仿真的 SysML模型元素、在配置SysML仿真窗口中配置这些元素以及观察仿真结果。
工件适合	Enterprise Architect帮助您扩展 SysML 参数模型的实用性，方法是使用允许模拟模型的额外信息对它们进行注释。然后将生成的模型生成为可以使用 MATLAB Simulink 或 OpenModelica 求解（模拟）的模型。 模拟属性针对您的仿真模型进行工件。这保留了您的原始模型并支持针对单个 SysML模型配置的多个模拟。仿真工件在工件的工具箱上找到
用户接口	SysML 模拟的用户界面在配置模拟仿真主题中进行了描述。
使用数据集进行模型分析	使用仿真配置，SysML块可以有多个针对它定义的数据集。这允许在SysML模型的模拟上运行可重复的变化。
SysPhS 标准支持	*SysPhS* 标准是交互物理和辅助信号流仿真的 *SysML*扩展。它定义了在SysML模型和 Modelica模型或 Simulink/Simscape模型之间转换的标准方法，为共享仿真提供了一种更简单的基于模型的方法。请参阅*SysPhS* 标准支持帮助帮助。
例子	为了帮助您了解如何创建和模拟 SysML 参数模型，我们提供了三个示例来说明三个不同的领域。所有三个示例都使用 OpenModelica 库。 *SysML*仿真示例主题中描述了这些示例以及您可以从中学到的东西。

使用 OpenModelica 库进行建模和仿真

此特征可从Enterprise Architect版本 14. 1开始获得。

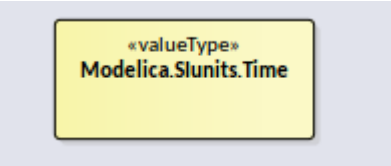
OpenModelica 库是提供许多有用类型、函数和模型的综合资源。在Enterprise Architect中创建 SysML 模型时，您可以参考 OpenModelica 库中可用的资源。

引用 OpenModelica 库中定义的类型

要配置模拟以引用 OpenModelica 库，首先创建一个指向 OpenModelica 库的 ValueType元素，并将其注册到仿真配置中。

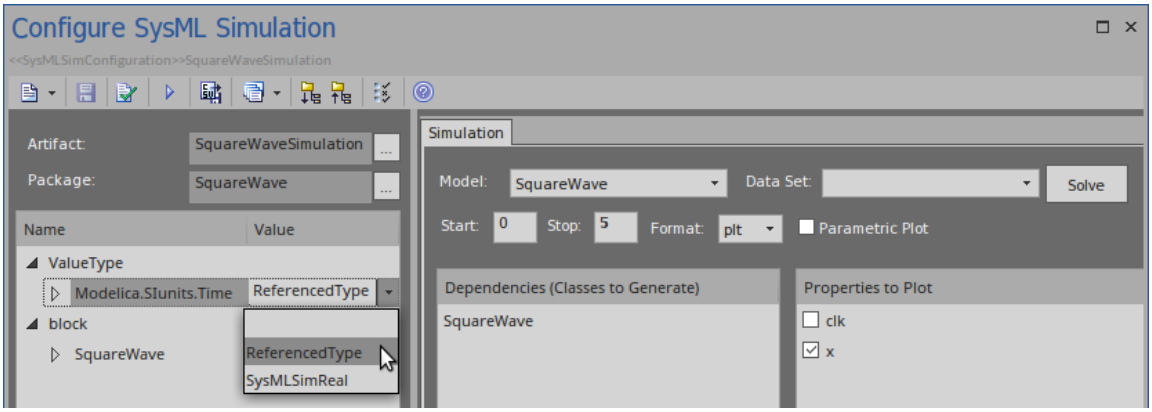
为引用的 OpenModelica类型创建一个元素

- 使用 OpenModelica 库路径的全名创建一个 ValueType元素



将配置元素配置为 'ReferencedType'：

- 双击 SysMLSimConfiguration元素以打开 "配置SysML配置"选项卡
- 导航到 ValueType元素
- 在下拉字段中将值设置为 "ReferencedType"



由于 ValueType元素配置为 "ReferencedType"，因此该元素不会显示在 "Dependencies"列表中，也不会生成为 OpenModelica 文件的新类定义。

将属性的类型设置为 ValueType元素

在Enterprise Architect中，可以将属性设置为原始类型或元素，例如块或 ValueType。

选项1 ：

- 选择属性（部件或端口）
- 按 Ctrl+2 打开属性窗口
- 切换到 "属性"选项卡并选择 "选择类型..."
- 浏览到您创建的 ValueType元素

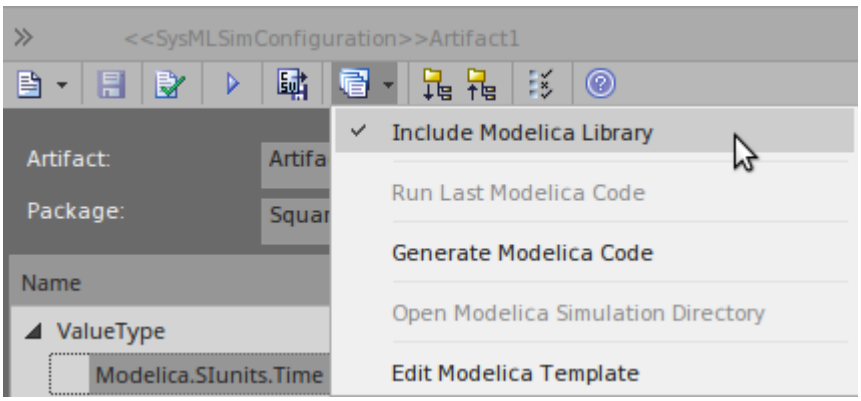
选项 2：

- 选择属性（部件或端口）
- 在属性上按 Ctrl+L
- 浏览到您创建的 ValueType元素

在仿真中包含 OpenModelica 库

在模型中使用 OpenModelica 库中的引用类型时，您必须在环境中加载 OpenModelica模型才能使仿真工作。

- 展开菜单选项并选择 包含 Modelica 库”



- 如果勾选这个选项，这个函数将默认生成到'Solve.mos'：
  加载模型（模型）；

点击这里查看 loadModel() 脚本函数的详细描述。

自定义脚本模板

您可以修改 OpenModelica 脚本模板以添加模型和仿真所需的额外库。选择功能区选项：

开发 >源代码 > 选项 > 编辑代码模板

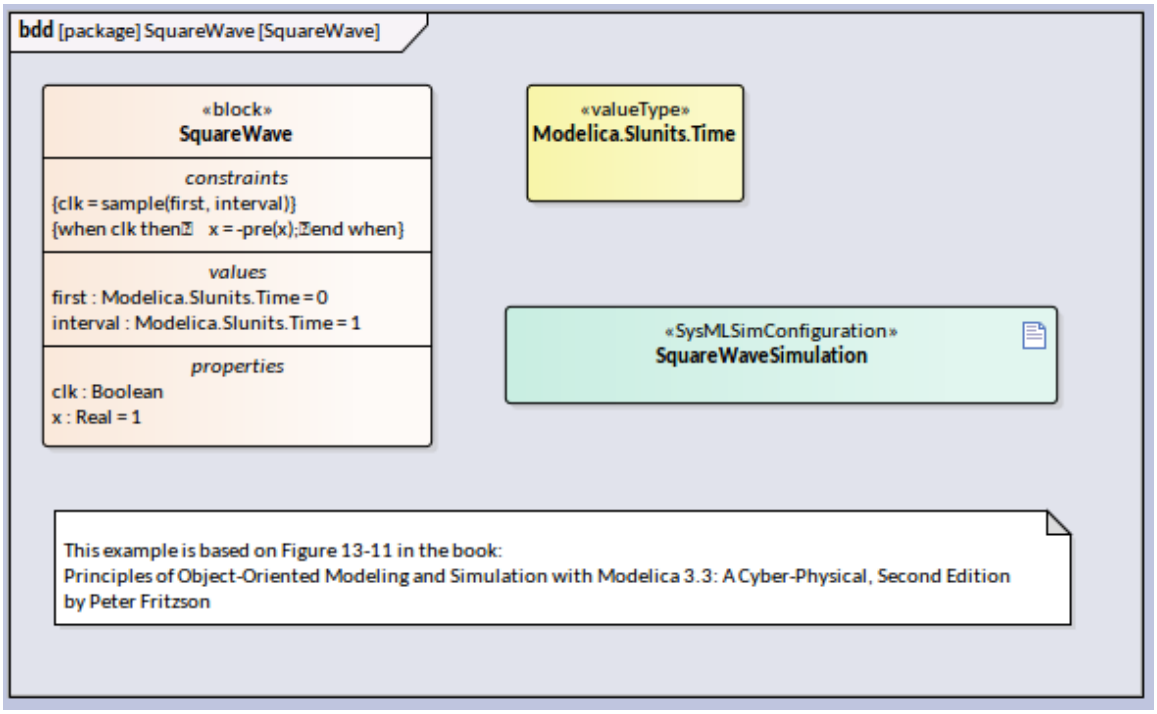
在 语言”字段中选择 Modelica”，在 脚本”列表中选择 脚本脚本”。

```
10 %if sysmlSim_IncludeLibrary == "T"%
11 msg := "---------------- Loading Modelica Model... ----------------";
12 loadModel(Modelica);
13 %endIf%
```

由于您在 loadModel(Modelica)”之 附加了额外的库，因此您的模型可以引用这些库的资源。

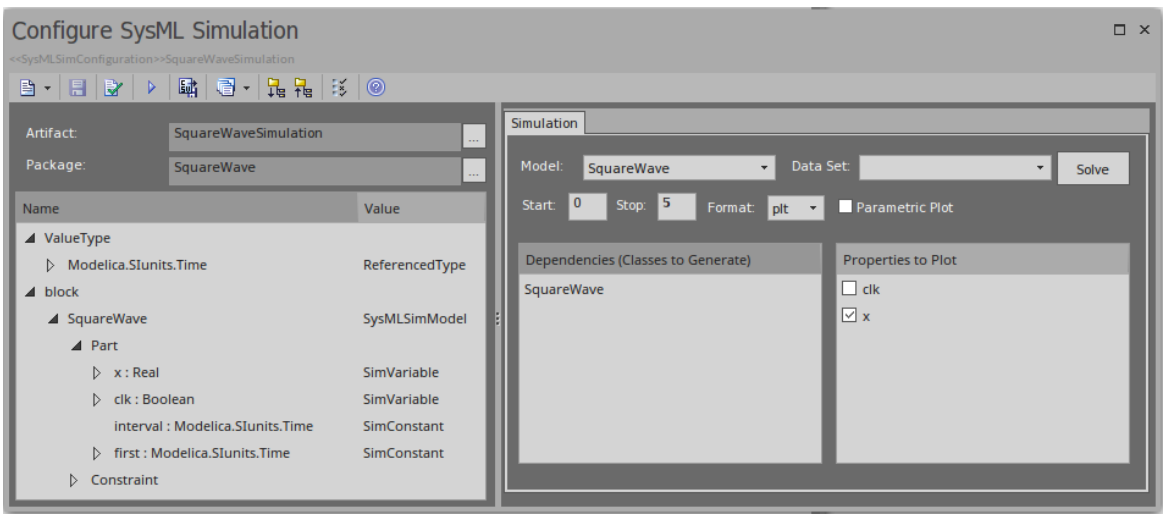
示例

此示例基于图 13-11 中的图 13-11：使用 *Modelica 3.3* 进行面向对象建模和仿真的原理： *A Cyber* -Physical，第二版，作者 Peter Fritzson。

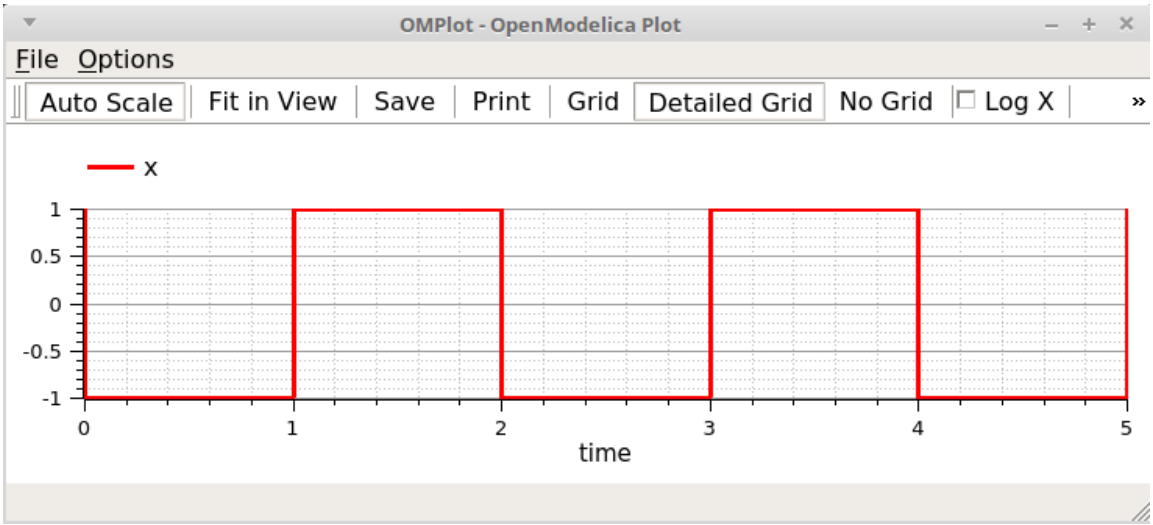


在这个例子中：

- 我们创建一个ValueType时间，用于块属性的属性*first*和*interval*
- ValueType时间在 SysML仿真窗口中配置为 "ReferencedType"
- 选择菜单项 "包括 Modelica 库"

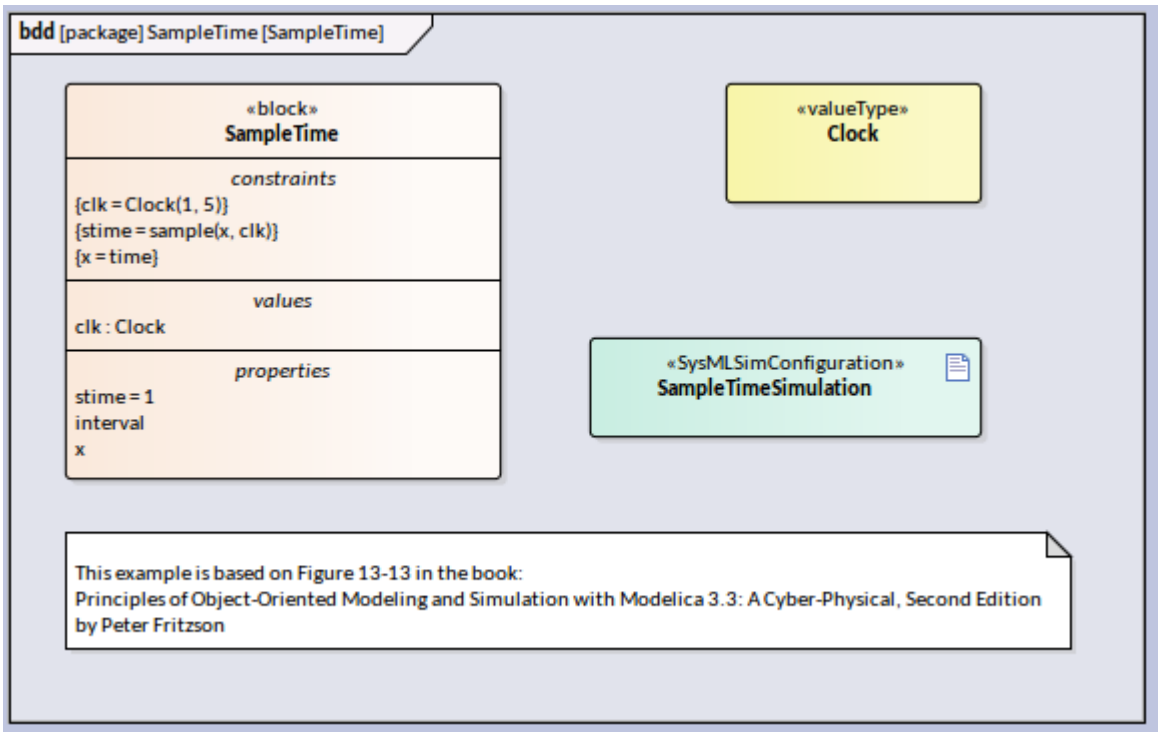


运行模拟；变量*x*绘制如下：



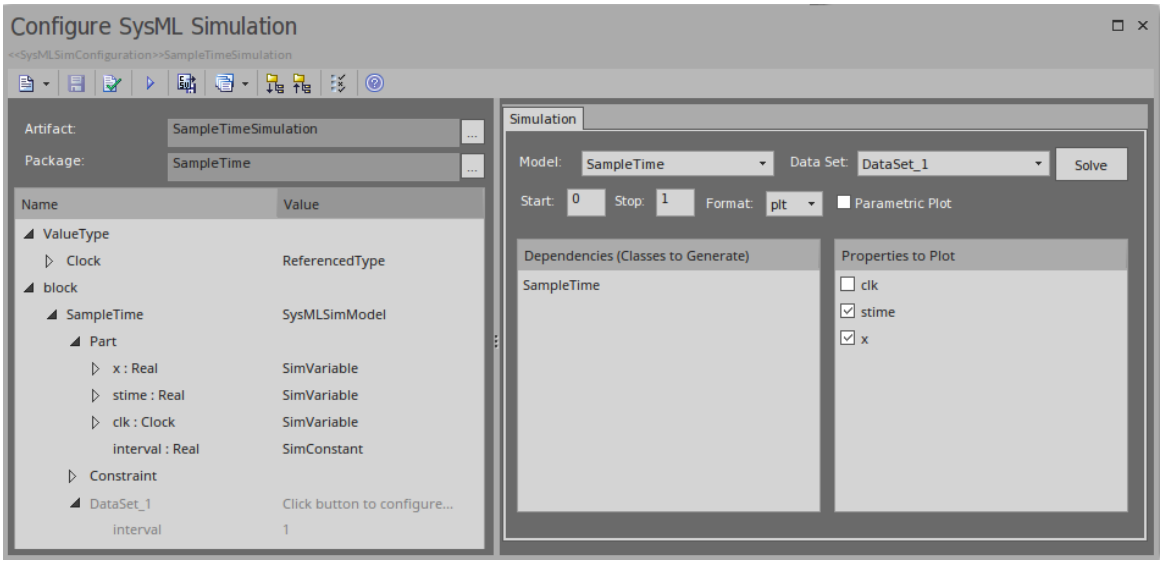
采样时间示例

此示例基于图 13-13 中的图 13-13：使用 *Modelica 3.3* 进行面向对象建模和仿真的原理： *A Cyber* -Physical，第二版，作者 Peter Fritzson。

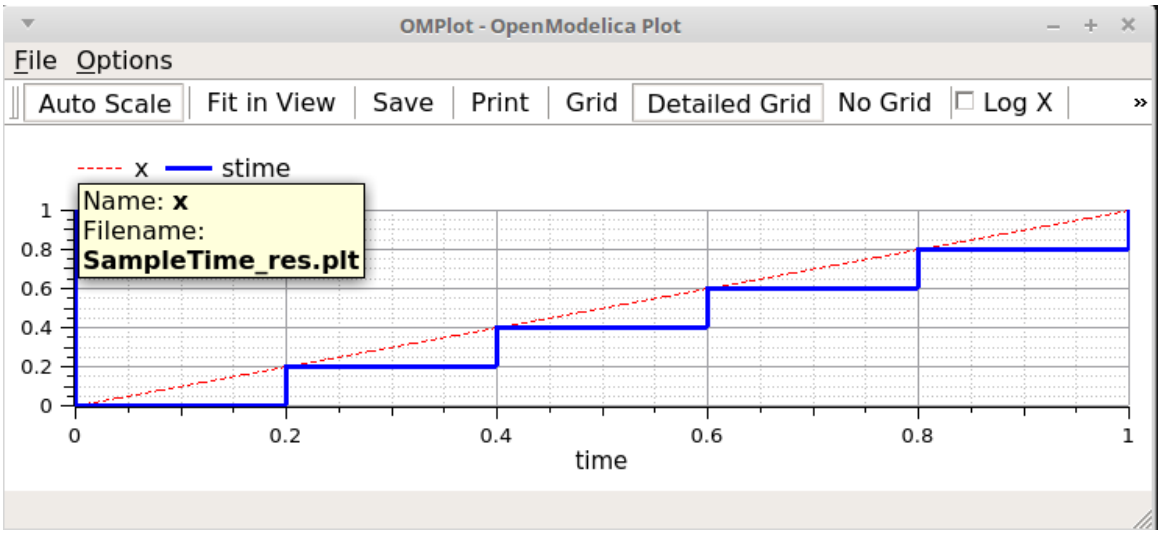


在这个例子中：

- 我们创建了一个块*Clock* ，用于*blockSampleTime*属性*clk*
- ValueType时钟在 SysML仿真窗口中配置为 "ReferencedType"
- 菜单项 包括 Modelica 库"未选中



运行模拟；变量x和时间图如下所示：



OpenModelica仿真疑难解答

虽然本主题描述了使用 OpenModelica 时可能出现的问题，但其中许多要点同样适用于使用 MATLAB Simulink 执行仿真。

公共仿真问题

此输出表描述了在使用 OpenModelica 时可能会阻止模型被模拟的一些常见问题。在系统输出窗口的 "编译"选项卡中选择输出。这些消息是从 OpenModelica 编译器 (omc.exe) 转储的，它通常会将您指向 OpenModelica源代码的行。这将帮助您找到大部分错误。

问题
方程的数量少于变量的数量。您可能忘记将某些属性设置为 "属性"，这意味着该值在模拟过程中不会改变。在开始模拟之前，您可能必须提供 "属性"属性值。（通过仿真数据集设置值。）
正在键入到端口的 Blocks 不包含保存的属性。例如，一个块 "ChargePort"包含两个部分—— "v：电压"和 "i：电流"。属性 "i": Current"应定义为 PhSVariable，属性 "isConserved"设置为 "True"。
PhSConstants 可能没有默认值——它们应该被提供。
PhSVariable A开始可能没有初始值——应该提供一个。
属性可能由配置包外部的元素（块或值类型）键入；使用包导入连接器来解决这个问题。

SysML仿真配置Filters

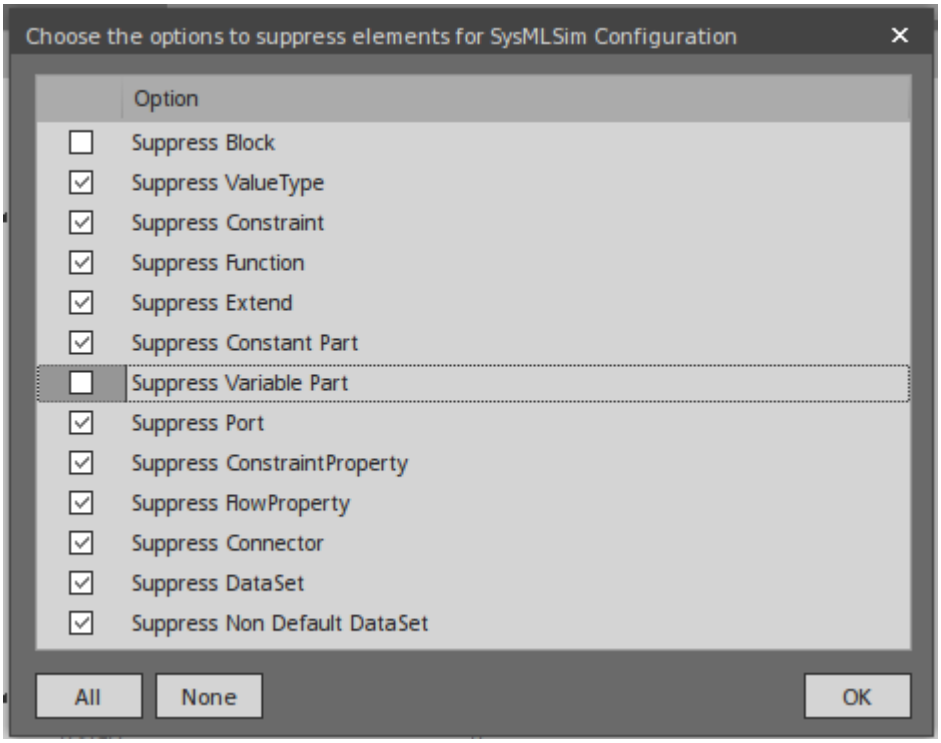
"SysML仿真配置"对话框默认显示包中的所有元素，包括值类型、约束、部件和块、端口和约束属性、连接器数据集。对于中型模型，完整列表可能会很长，您可能很难找到潜在的建模错误。

在 TwoTanks 示例中，如果我们清除 Tank.area属性'PhSConstant'，然后进行验证，我们会发现这个错误：

错误：方程太少，系统欠定。该模型有 *11* 个方程和 *13* 个变量。

此错误表明我们可能忘记为 "属性"设置某些属性。

我们现在可以做的是单击工具栏右侧的第二个按钮（用于配置的过滤器）并打开此处显示的对话框。单击全部按钮，然后取消选择 "抑制块"和 "抑制可变部件"复选框，然后单击确定按钮。



现在我们将有一个更短的变量列表，从中我们可以发现 面积"在仿真过程中没有变化。然后我们将其定义为"PhSConstant"并提供一个初始值来解决问题。

模型验证示例

信息	讨论
约束中未定义的变量	在 TwoTanks 示例中，当我们浏览到 "constraintBlock.Outcontrol.约束"时，假设我们发现了一个输入错误：我们在约束中输入了 "v""而不是 "b""。 所以，而不是： *a=b*(c+d)* 我们输入： *a=v*(c+d)* 单击工具栏上的验证按钮。这些错误消息将出现在 "Modelica"选项卡中： 正在验证模型... 错误：在范围 *OutControl* 中找不到变量 *v*。（表达式：*"a=v*(c+d)；"*） *Error: Error occurred while flattening*模型*TanksConnectedPI* 发现的错误和警告数：*2* 双击错误行；配置列表显示并突出显示约束。 更改'v' 改回 'b' 并再次单击 Validate 按钮。未发现任何错误，问题已解决。 提示：使用 SysML仿真配置视图是更改块或约束块约束的快捷方式。你可以： • 更改约束到位 • 删除使用约束的上下文菜单 • 使用块或约束块的上下文菜单添加新约束

重复的变量名称	在 TwoTanks 示例中，浏览到*block.tank.constraintProperty.e1* 。假设我们给两个属性取了相同的名字： • 右键单击*e1* ，选择 "在项目中查找浏览器"选项，并将名称更改为*e2*；重新加载 "SysML仿真配置"对话框 单击工具栏上的验证按钮；这些错误消息出现在 "Modelica"选项卡中： 正在验证模型... 错误：重复元素（由于继承的元素）不相同：（表达式：*"SensorValue e2;"*） *Error: Error occurred while flattening*模型*TanksConnectedPI* 发现的错误和警告数：*2* 双击错误行；配置列表显示并突出显示约束属性。 更改其中一个的名称从*e2*改回*e1*并再次单击 Validate 按钮；没有发现错误，问题已解决。
属性定义在 ConstraintBlocks Not Used	在 TwoTanks 示例中，在浏览器窗口中，我们浏览到元素 "示例模型.系统工程.ModelicaExamples.TwoTanks.constraints.OutFlow"。 假设我们添加了一个属性 "c"和一个潜在的新约束，但我们忘记了为实例同步 - ConstraintProperties。如果我们不运行验证，这将导致方程式太少，系统欠定错误。 在 "SysML仿真配置"对话框中重新加载包，然后单击工具栏上的 "验证"按钮。这些错误消息将出现在 "Modelica"选项卡中： 正在验证模型... 错误：*ConstraintProperty "e4"* 缺少在键入 *ConstraintBlock "OutFlow"* 中定义的参数。（缺失：*c*） 错误：方程式太少，系统未定。模型有 *11* 个方程和 *12* 个变量。 发现的错误和警告数：*2* 双击错误行；配置列表显示并突出显示 ConstraintProperty。ConstraintProperty 被键入到*outFlow*并且缺少新参数 "c"。 右键单击配置列表中的ConstraintProperty，选择 "在所有图表中查找"选项，然后右键单击图表上的 "约束"属性；选择'特征|部件/属性"并选中 "显示拥有/继承"复选框，然后单击 "c"。 在 "SysML仿真配置"对话框中重新加载模型，然后单击 "验证"按钮。这些错误消息将出现在 "Modelica"选项卡中： 正在验证模型... 错误：*ConstraintProperty "e4"* 没有参数 "c" 的任何传入或传出绑定连接器。 错误：方程式太少，系统未定。模型有 *11* 个方程和 *12* 个变量。 发现的错误和警告数：*2* 为了解决这个问题，我们可以根据实际逻辑做两件事之一： 1. 如果在ConstraintBlock中属性'c'是必需的，并且使用'c'定义了一个约束，那么我们需要在ConstraintProperty的上下文中添加一个属性并绑定到参数'c'。 2. 如果不需要属性'c'，那么我们可以在约束块中点击这个属性，然后按Ctrl+D键。（相应的 ConstraintProperties 将自动删除 "c"。）