ENTERPRISE ARCHITECT

用户指南系列

动态模拟

Author: Sparx Systems

Date: 2022-08-30

Version: 16.0

创建于 ENTERPRISE ARCHITECT

目录

动态模拟

模型仿真通过即时、实时的行为模型执行使您的行为模型栩栩如生。再加上管理触发器、事件、守卫、效果、断点和仿真变量的工具，加上在运行时可视化跟踪执行的能力，模拟器是一种 观察车轮转动"和验证正确性的多功能手段。你的行为模型。借助仿真，您可以探索和测试模型的动态行为。在企业版、统一版和终极版中，您还可以使用JavaScript作为运行时执行语言来评估守卫、效果和其他可编写脚本的行为项。

对触发器、触发器集、嵌套状态、并发性、动态效果和其他高级仿真功能的广泛支持，为构建交互式和工作模型提供了一个卓越的环境，有助于探索、测试和可视化跟踪复杂的业务、软件和系统行为。启用JavaScript后，还可以创建嵌入式 COM 对象，这些对象将执行评估防护和执行效果的工作——允许将仿真绑定到更大的依赖进程集。例如，COM object评估状态转移

上的保护条件转移

可能会查询本地运行的进程，读取和使用一组测试数据，甚至连接到 SOA Web 服务以获取一些当前信息。

由于Enterprise Architect使用动态的、脚本驱动的仿真机制，可以直接分析和使用UML结构，因此在运行仿真之前无需生成中间代码或编译仿真 可执行文件"。这导致了一个非常快速和动态的仿真环境，可以在其中快速进行更改和测试。甚至可以使用仿真控制台窗口实时更新仿真变量。这对于 劲态"测试替代分支和条件很有用，无论是在设置的仿真断点处还是在仿真器达到稳定点时（例如，当仿真器被 阻塞"时）。

在Enterprise Architect的专业版中，您可以手动浏览模拟 - 尽管没有JavaScript将执行 - 所以所有选择都是手动决定的。这对于测试行为模型的流程和突出可能的选择和处理路径很有用。在企业版、统一版和终极版中，您可以：

- 动态执行您的行为模型
- 评估用标准JavaScript编写的守卫和效果
- 定义并触发触发器以运行模拟
- 定义和使用触发器集来模拟不同的事件序列
- 自动触发触发器集以模拟复杂的事件历史，无需用户干预
- 即时"更新仿真变量以改变仿真的进行方
- 在仿真期间创建和调用 COM 对象以扩展仿真的范围和输入/输出的可能性
- 运行时检查仿真运行
- 设置脚本 序言"以在执行前定义变量、常量和函数
- 使用具有不同 序言"的脚本仿真各种条件下运行分析器

在统一版和终极版中，还可以模拟 BPMN 模型。

使用模型模拟器，您可以模拟包含行为的概念模型设计的执行。启动仿真时，会对当前模型包进行分析，并触发动态仿真过程来执行模型。

要启动并运行仿真，只需执行以下步骤：

- 编译行为图（状态或活动用于手动或动态执行，序列用于手动交互）
- 可选：加载 仿真仿真工作空间"布局——快速调出所有常用仿真窗口
- 点击模拟器播放按钮

如果图表包含任何外部元素（与图表不在同一个包中的那些），您将必须创建一个从图表包到包含外部元素的包的导入连接器。为此，将两个包从浏览器窗口拖到图表上，然后使用快速链接器箭头在它们之间创建连接器。

仿真概览

方面
模型模拟器概述

使用窗口及相关窗口的仿真及运行仿真
设置仿真并激活仿真脚本
设置和使用仿真断点
仿真物体的使用
不同类型行动在仿真中的使用
用JavaScript进行动态仿真
使用中防护条件的使用
使用中的使用触发器
调用行为和变量
仿真活动返回
仿真结构活动行为
仿真多线程进程
在单独的图表中仿真子流程
执行 BPMN 模拟
仿真Win32对话框行为

平台和可用版本

平台/版	细节
支持的型号和平台	模型模拟器目前支持在仿真平台上执行UML活动交互、状态机模型和业务模型： • UML基础 • BPMN
版支持	模型仿真在Enterprise Architect的各个版本范围内提供不同级别： • 专业-仅限手动仿真 • 企业及以上 - 添加动态JavaScript评估；目前为状态机和活动图启用了JavaScript ；没有为交互启用 • 统一终极-新增BPMN仿真

它看起来如何

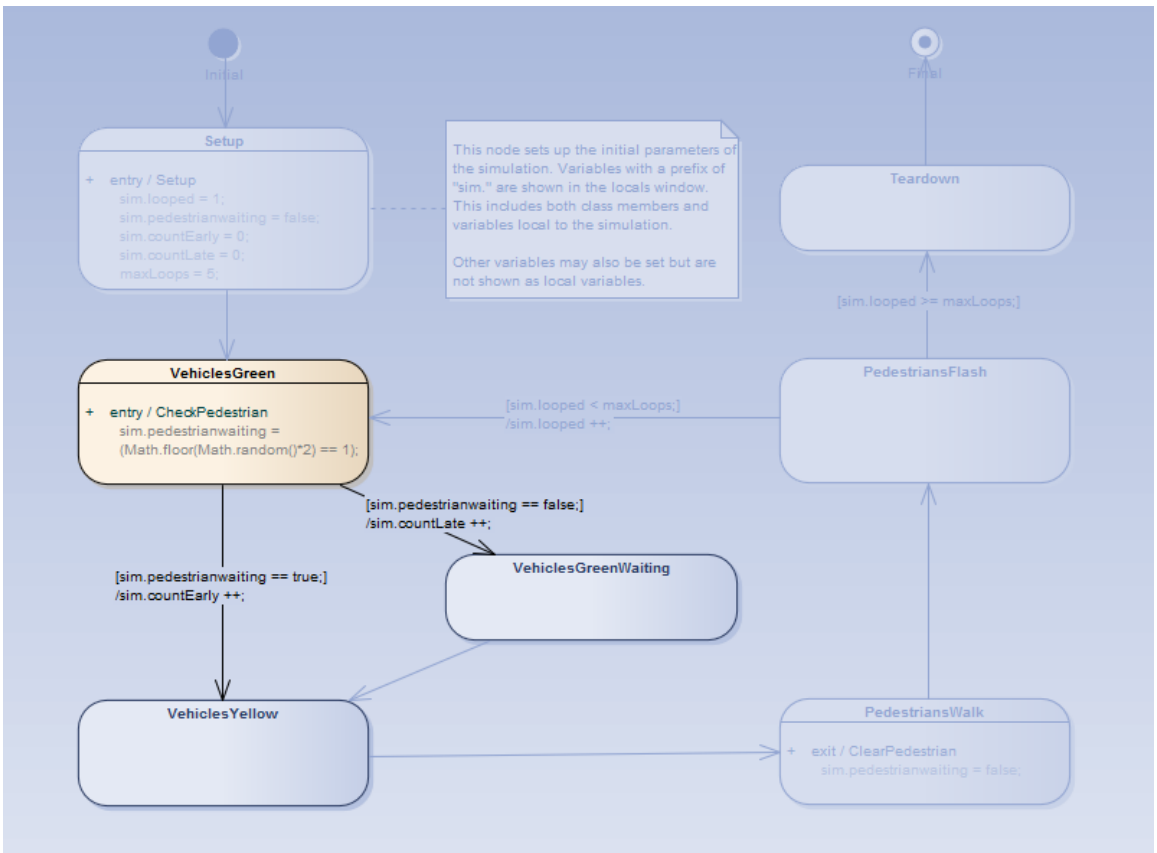
Enterprise Architect在仿真过程中有一种特殊的模型信息展示方式。这有助于聚焦集中在执行或活动节点上。

在仿真过程中， Enterprise Architect将动态跟踪并突出显示模型中的活动节点。如果另一个图表中的节点被激活，该图表将被自动加载并突出显示当前节点。仿真运行时可以修改图表；但是，在当前的仿真结束并开始新的仿真之前，所做的更改不会被识别。

仿真过程中活动节点的突出显示

在此示例中，当前活动节点 (VehiclesGreen) 以正常Enterprise Architect颜色突出显示，并且当前节点的所有可能转换都以全强度呈现。

作为当前活动节点的传出转换的可能目标的元素以半淡化样式呈现，以便它们易于阅读并且与图中的其他元素明显不同。所有其他元素都以完全褪色的样式呈现，以表明它们不是下一步仿真步骤的目标。随着仿真的进行（特别是如果自动运行），这种突出显示有助于将聚焦集中在当前项目及其视觉上下文。



仿真窗口

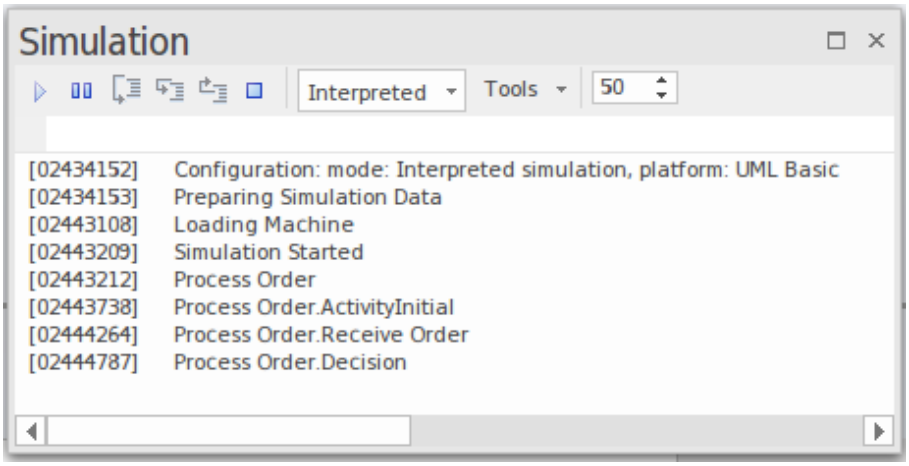
在Enterprise Architect中执行仿真时，可以设置断点、触发执行速度、检查变量执行速度、查看调用堆栈仿真可视化仿真活动节点。

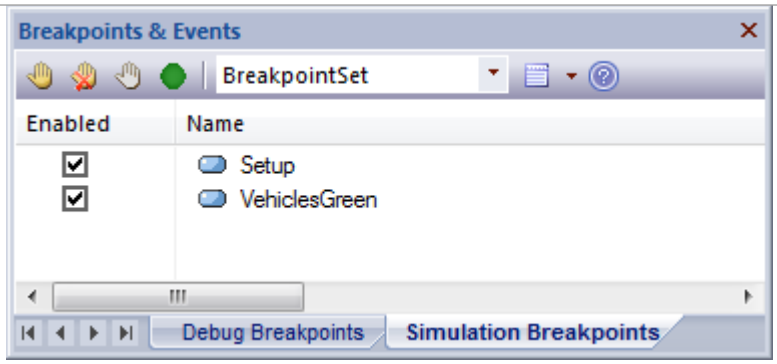
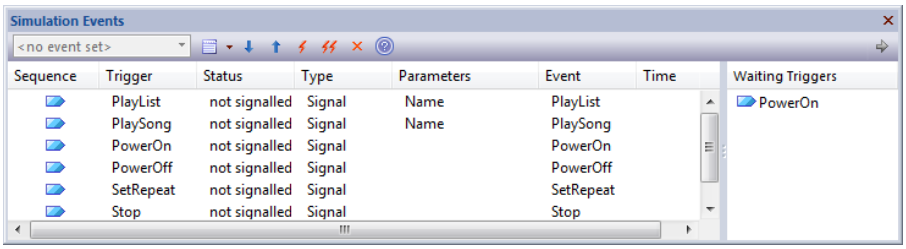
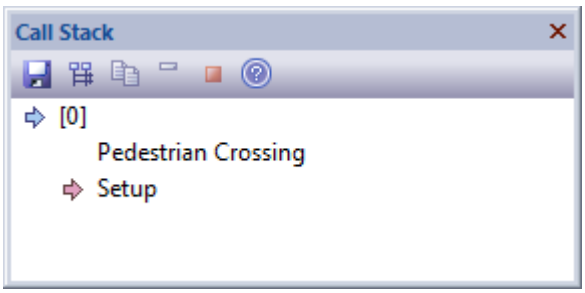
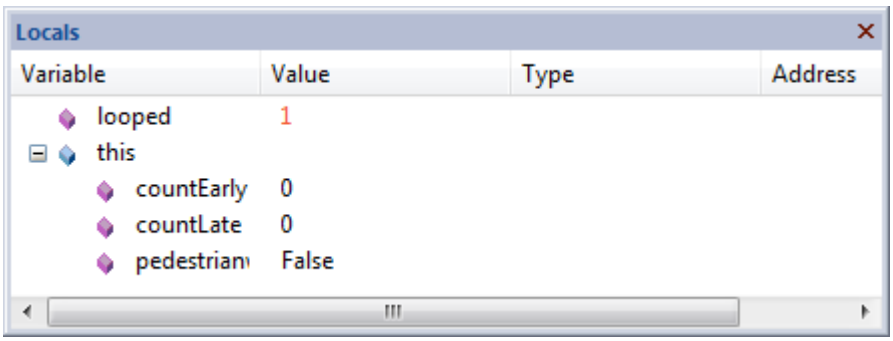
仿真运行时，输出和控制台输入等一些方面是在窗口本身中仿真的，而局部变量和调用堆栈使用标准执行分析器窗口。本主题概述了仿真期间使用的主要窗口。

访问

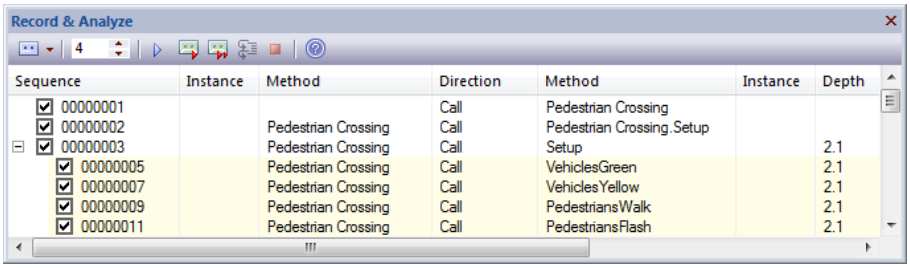
功能区	仿真>动态仿真仿真> 模拟器> 打开仿真窗口

窗口

窗户	目的
执行和控制台	仿真窗口提供启动、停止和步进仿真的主界面。在执行期间，它显示与当前执行步骤和其他重要信息相关的输出。有关工具栏命令的更多信息，请参阅运行模型仿真主题。 注记工具栏正下方的文本输入框。这是控制台输入区域 - 在这里您可以输入简单的JavaScript命令，例如： *this.count = 4;*将名为 "count"的模拟变量动态更改为 4。通过这种方式，您可以在运行时动态影响模拟。 
断点&事件窗口	仿真过程还使用了断点和标记窗口的 "仿真断点"选项卡（ 仿真>动态仿真仿真>断点"）。在这里，您可以在仿真中的特定元素和消息上设置执行断点。详见仿真断点专题。

	
仿真事件窗口	仿真事件窗口（ 仿真>动态仿真>事件”）提供了管理和执行触发器的工具。触发器状态机用于控制过渡的执行。 
调用堆栈窗口	调用堆栈仿真（'仿真>动态仿真>仿真调用堆栈'）显示当前线程和执行上下文的信息。 模拟器支持多线程模拟，并且将为每个活动和暂停的执行线程包括一个线程条目。对于每一个线程，调用堆栈窗口会显示该线程的开始或元素上下文（如状态机元素）加上当前活动的元素。如果当前活动元素是复合活动或状态的入口点，则堆栈还将包括该子上下文中的当前活动元素（以及所有进一步嵌套的活动组合子状态）。 
仿真局部变量窗口	模拟器使用标准的本地窗口窗口（ 仿真>动态仿真仿真>局部变量”）在模拟单步执行或在断点处暂停时显示所有当前模拟变量。注记可以使用动态更新这些变量模拟器控制台。 
记录	在执行模拟期间，会保留所有活动的记录并显示在 记录和分析”窗口中（ 执行 > 工具 > 记录器 > 打开记录器”）。这与可视化执行分析器中正常通话录音

的工作方式类似。



Sequence	Instance	Method	Direction	Method	Instance	Depth
00000001			Call	Pedestrian Crossing		
00000002		Pedestrian Crossing	Call	Pedestrian Crossing.Setup		
00000003		Pedestrian Crossing	Call	Setup		2.1
00000005		Pedestrian Crossing	Call	VehiclesGreen		2.1
00000007		Pedestrian Crossing	Call	VehiclesYellow		2.1
00000009		Pedestrian Crossing	Call	PedestriansWalk		2.1
00000011		Pedestrian Crossing	Call	PedestriansFlash		2.1

设置仿真脚本

您可以使用仿真脚本对仿真的启动方式进行精细控制。在一般情况下，您不需要设置仿真脚本，除非：

- 您想运行需要在仿真开始前初始化变量的解释性仿真；这对于设置全局变量和定义函数很有用
- （在企业版及以上版本中）您不想应用解释 Guards 的默认行为（即，您更喜欢使用手动执行），或者
- 您希望有多种方式运行同一个图表

对于大多数图表，可以简单地通过在开始元素之后的第一个元素或连接器中设置变量来初始化仿真脚本。对于状态图，这是退出初始元素的 Transit 连接器，对于活动模型，这是第一个行动元素。

作为替代方案，您可以使用仿真脚本在仿真开始之前初始化设置。这对于使用多个分析器脚本设置不同的初始值集非常有用，以便您可以在一系列预设条件下运行您的仿真。

要配置仿真脚本包，首先在浏览器窗口、浏览器、图表列表或模型搜索中选择包。然后，您可以使用执行分析器窗口为选定的包添加新脚本。您将使用 "执行分析器"对话框的 "仿真"页面来配置相关属性。

访问

使用此处概述的方法之一显示执行分析器窗口。

在执行分析器窗口中：

- 找到并双击所需的脚本，然后选择 "仿真"页面或
- 点击窗口工具栏中的 并选择 "仿真"页面

功能区	开发>源代码>执行分析器>编辑分析器脚本 执行 > 工具 >分析器
上下文菜单	浏览器窗口 |右键单击包|执行分析器
键盘快捷键	Shift+F12

配置仿真脚本

选项	行动
平台	对于UML活动，交互状态机模拟，单击下拉箭头并选择' UML Basic'。 对于 BPMN 图，单击下拉箭头并选择 "BPMN"。
入口	单击 按钮并选择： ● 仿真的入口点，和 ● 活动，交互状态机来模拟 如果不指定入口点，模拟器会尝试遍历整个包。
使用JavaScript评估防护条件	（在企业及更高版本中）不选中复选框以执行手动仿真，您可以在其中选择下一个要转换到的状态以及必须做出决定的点。 选中复选框以执行仿真中影响行为的代码。仿真在这些地方执行JavaScript代码：

	● 状态进入/退出/做操作 ● 转移 守卫/效果 ● BPMN活动循环条件和序列流条件表达式 除了守卫之外，所有这些都应该是一个或多个有效的JavaScript语句，包括分号。 守卫必须是有效的布尔表达式，也以分号结尾。 当到达仿真断点时，属于 'sim' 或 'this' 成员的变量会在本地窗口窗口中列出。 sim.count = 0;
输入	启用JavaScript后，您可以在此字段中键入脚本命令，这些命令将在仿真运行之前运行。
后处理脚本	使用 Post仿真脚本，您可以在仿真结束后运行JavaScript 。类型在脚本的限定名称中来自模型脚本控件。 例如，如果脚本组 "MyGroup"中有一个名为 "MyScript"的脚本，请输入值 脚本"。
确定	单击此按钮以保存您的更改。

注记

- 通常所有的仿真元素和关系都在为仿真配置的包内；但是，您可以通过从配置包创建包导入连接器到每个 外部"包来模拟包含来自不同包的元素的图表（或者，对于 BPSim模型，创建从配置包到每个外部元素的依赖连接器）

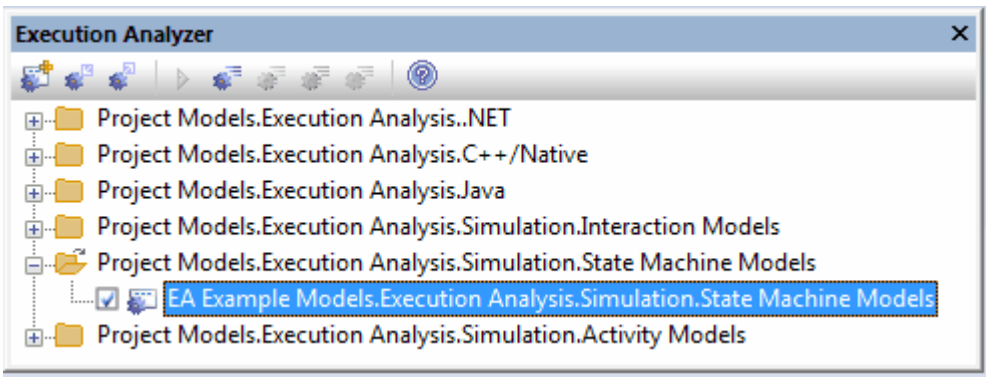
激活仿真脚本

定义仿真参数的模型包配置执行脚本。激活执行脚本的最常见原因是针对一个包配置了多个仿真脚本，并且您想要运行特定的一个。

访问

功能区	执行 > 工具 >分析器 开发>源代码>执行分析器
分析器窗口	点击分析器脚本使其激活
键盘快捷键	Shift+F12

激活仿真脚本执行

节	行动
1	在执行分析器窗口中，选择需要的执行脚本。这使它成为您打开模型的当前默认设置，以便单击仿真运行脚本仿真 
2	单击脚本左侧的复选框以激活它。
3	选择 仿真> 模拟器 > 打开仿真窗口”功能区选项来执行仿真。

运行模型仿真

A仿真逐步执行模型，使您能够验证行为模型的逻辑。当前执行步骤在模型图中自动突出显示，以便于了解仿真过程中发生的各种过程和状态变化。

启动模型仿真有几种方式：

- 当可以模拟活动图表时，主仿真窗口上的运行按钮将通过运行现有脚本或定义新的临时脚本来处理当前图表
- 当活动图无法模拟时，主仿真窗口上的运行按钮将运行活动图的仿真执行分析器脚本
- 通过右键单击执行分析器窗口中的仿真脚本并选择 开始仿真”选项
- 通过右键单击合适的图表并选择 执行仿真”选项之一

执行过程中有视觉提示。仿真运行时， Enterprise Architect会主动突出显示每个执行步骤的每个活动节点。此外，所有传出的转换和控制流将被突出显示，显示可能的前进路径。在可能的前进路径末端的元素将被淡化为一半强度，任何其他剩余元素将 90% 变灰”。这提供了一个非常动态且易 遵循的执行，不断将注意力重新集中在执行上下文上。

访问

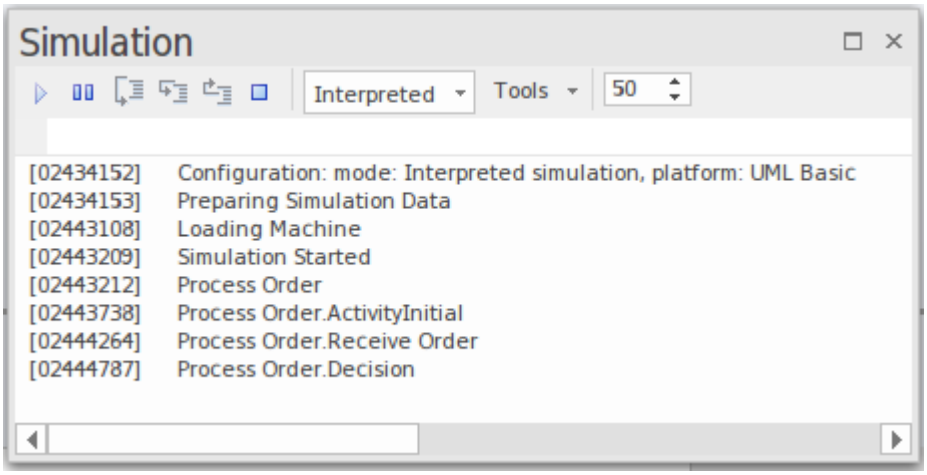
功能区	仿真>动态仿真仿真> 模拟器> 打开仿真窗口 仿真>运行仿真>开始

版具体细节

在专业版中，如果在执行中遇到分支，模拟器会提示您选择合适的路径来执行。

在启用了JavaScript的企业版、统一版和终极版中，仿真仿真会自动评估所有防护和效果，并在无需用户干预的情况下动态执行。如果仿真由于没有可能的向前评估为True的路径（或多个路径评估为True ）而阻塞，您可以使用仿真执行窗口的控制台输入即时修改仿真变量。

运行仿真使用工具栏



图标	行动

	为当前图表启动开始，或者，如果当前图表无法模拟，则使用运行的仿真脚本运行仿真。
	暂停仿真。
	当仿真器暂停时，通过跨步、跨步、跨步来控制模拟器在模型仿真中要求的步骤执行。
	停止仿真。
Manual Manual Interpreted Executable	点击下拉箭头，选择仿真运行类型： • 'Interpreted' - 动态执行仿真（企业版、统一版和终极版） • '手动'-节通过仿真手动（专业版中唯一可用的选项） • 'Executable' - 在可执行状态机上运行仿真时选择
Tools	单击下拉箭头，从选项菜单中选择对仿真脚本和输出执行特定操作的选项，例如编译、运行、生成和视图断点。
60	改变仿真的执行率，在 0% 和 100% 之间；在： • 100%，仿真以最快的速度执行 • 0% 模拟器在每条语句处中断执行

注记

- 只有激活了有效的仿真执行脚本，仿真工具才会激活
- 您可以通过在执行分析器窗口中设置仿真脚本的复选框来将其设置为当前默认值

仿真断点

仿真断点和事件窗口的 仿真断点”选项卡可让您中断和检查仿真过程。

当动态执行仿真（在企业统一和终极版本中）时，该过程将自动进行 - 如果您想在某个点停止执行以检查变量、检查调用堆栈或以其他方式与模拟器交互，您可以在一个模型元素的方式与使用一行源代码的方式大致相同。当模拟器到达断点时，执行停止并将控制权返回给Enterprise Architect 。

访问

功能区	仿真>动态仿真仿真>断点>仿真断点

断点

仿真逐步执行模型，使您能够验证您的行为模型的逻辑；仿真在到达定义为断点的元素时停止。

可以定义为断点的UML元素包括行动、活动、状态和大多数其他行为节点，例如决策、初始或终点。

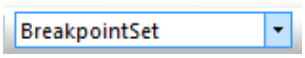
可以定义为断点的UML关系包括交互。

断点存储为给定Enterprise Architect项目的断点集。

元素中包含的、有断点的元素，在仿真过程中，仿真元素左上角附近用绿色圆圈标出。如果仿真没有运行，则不显示绿色圆圈。

启用JavaScript后，所有仿真变量都将显示在本地窗口窗口中，并且可以使用仿真窗口的控制台输入字段（工具栏下方）修改这些仿真变量。

工具栏按钮

物品	描述
	为仿真会话启用当前断点集中定义的所有断点。
	删除在当前断点集中为仿真会话定义的所有断点。
	禁用所有断点 在仿真会话的当前断点集中定义。
	将所选元素或序列消息的断点添加到当前断点集。
	修改选定的断点集以在仿真会话中使用。
	执行断点设置命令： • **新集**：创建一个新的断点集 • **Save As Set** ：以新名称保存当前的断点集 • **删除选定集**：删除当前断点集

	• **删除所有集**：删除为图表保存的所有断点集

仿真中的对象和实例

当一个给定的业务、系统或机械流程执行时，其中的活动和行动可能会生成特定类型的对象并对这些对象执行操作，甚至可能会消耗或销毁它们。您可以使用模拟模型来模拟此类对象的创建、使用和使用，该仿真模型使用类、实例对象、属性、操作和（行动销端口对象节点）等模型元素来表示对象和动作。作为同一过程的一部分，模型还可以在不同阶段创建、作用和销毁几个不同的对象。在仿真中表现模型数据或实物，可以使仿真更准确地反映真实过程。

物件概念

团队	描述
模拟类型	仿真元素的类型，如类、枚举或接口。这些可以是仿真中对象的分类器。
模拟对象	作为 SimType元素实例（由其分类）的object 。
属性	属性元素或指定节点（如 ActivityNode） A属性。
手术	SimType元素或指定节点（例如 ActivityNode） A行为。
端口	A类或物件的端口，一个行动销端口的行动，或object活动的物件节点端口
参数/ 活动参数	操作参数；活动参数，具体来说就是ActivityNodes的参数。
投币口	object中属性A实现。 Slot 有A运行时间值，可以通过运行的运行状态值来初始化。如果这些值不存在，系统将使用属性的初始值。
运行时环境	所有对象都存在于JavaScript运行时环境中，因此您可以使用JavaScript创建或更改模拟对象和模拟变量。
显示变量	所有模拟对象、仿真变量或事件在它们生效时都会在本地窗口窗口中被识别出来。在某些情况下，要显示变量，您可能需要在模型中添加断点以在变量存在时暂停处理。 由于显示了所有对象和变量，存在于模拟之外但对其重要的全局变量（例如定义流程的父类和活动元素）也自动表示为默认object变量。活动的预期输出也是如此，作为返回变量。

在仿真中创建对象

在仿真模型中，您可以创建类并创建它们的实例（全局对象）来表示过程中存在的对象，或者定义行动以在过程中的任何时间点生成一个或多个对象。

在仿真模型中创建对象有三个选项：

- 手动创建物件
- 通过 CreateObject行动元素动态创建物件
- 使用JavaScript函数使用("name") 作为行动元素的 影响"，再次动态创建物件

动态创建一个物件后，您还可以实例化该物件的任何内部对象，例如类上的活动属性，并对该内部object执行操作。

手动创建一个物件

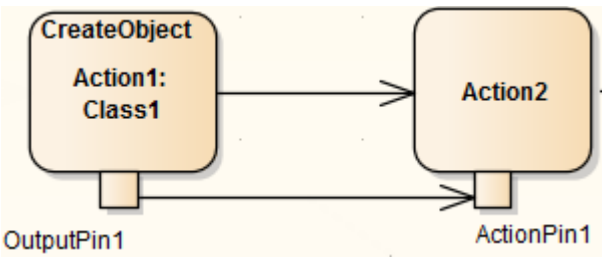
只需通过以下方式在模型的图表上创建一个物件元素：

- 从工具箱的 物件"页面图表一个物件元素并设置其分类器，或
- 从浏览器窗口拖动一个分类器元素并将其作为实例粘贴到图表中

在仿真模型中，您可以自行设置物件属性（例如设置运行状态以重新设置属性的初始值）或动作的行动以作用于物件（例如将其沿流程传递）)并观察仿真中的物件会发生什么。

通过 CreateObject行动创建一个物件

如果您的流程在运行时生成对象，您可以使用仿真行动来模拟这一点。

节	行动
1	在您的活动图表上，从工具箱中图表一个 行动"图标，然后选择 其它| CreateObject 的上下文菜单选项将其定义为 CreateObject行动元素。
2	将 CreateObject行动的分类器设置为物件将成为实例的类。这是在属性窗口 > CreateObjectAction >分类器中设置的，使用 [...] 按钮。
3	在行动上创建一个行动销，种类输出。
4	在处理行动中创建或选择下一个行动序列，并添加一个行动销类输入。 用控制器流量连接器连接两个行动，用控件物件流连接器连接两个行动销。 
5	在图上进行仿真。当创建对象行动被执行时，它创建一个具有分类器属性的物件，并将其存储在其输出销中。物件本身通过物件流连接传递到行动2的输入销，其属性可以在本地窗口窗口中列出，用于仿真。

使用JavaScript创建物件

您还可以在行动元素的 影响”字段中使用JavaScript命令动态创建仿真对象。命令是：

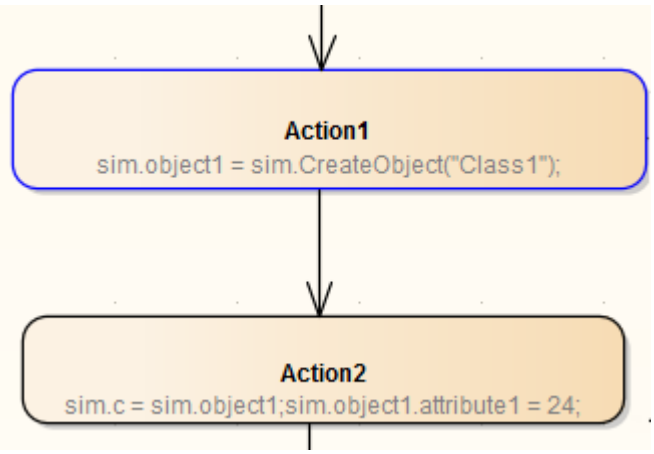
sim.newObject = sim.CreateObject("ClassName");

或者

sim.newObject = new SimObject("ClassName"); （自然的JavaScript ）

即： 仿真基 类<名称>的物件创作”。分类类将与行动存在于同一个包中。

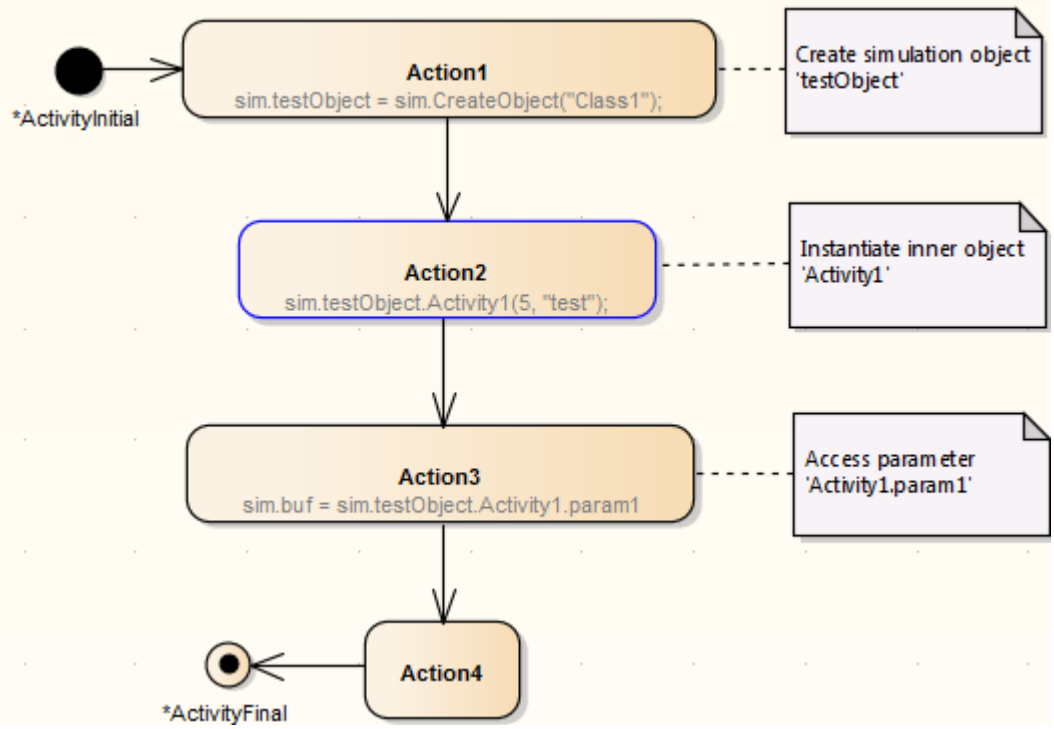
至于CreateObject行动元素，该物件是在仿真过程中创建的，可以被 下游”元素传递和处理。在这个例子中，创建的物件被标识为物件，并且在行动2 中它被访问，并且它的一个属性被赋予了不同的值（也被JavaScript作为行动的影响）。



实例化内部对象

如前所述，您可以使用JavaScript或 CreateObject行动创建一个物件。同样，您可以使用JavaScript或行动来实例化内部对象。

在这个例子中，使用JavaScript ，仿真首先创建了一个基于 Class1 的测试object 。类1有一个活动元素和图表，活动参数1设置为整数 5，活动参数2 设置为string “test”。活动参数1的值被捕获为缓冲区值 buf”。


*ActivityInitial
Action1
sim.testObject = sim.CreateObject("Class1");
Create simulation object 'testObject'
Action2
sim.testObject.Activity1(5, "test");
Instantiate inner object 'Activity1'
Action3
sim.buf = sim.testObject.Activity1.param1
Access parameter 'Activity1.param1'
Action4
*ActivityFinal

破坏仿真中的物体

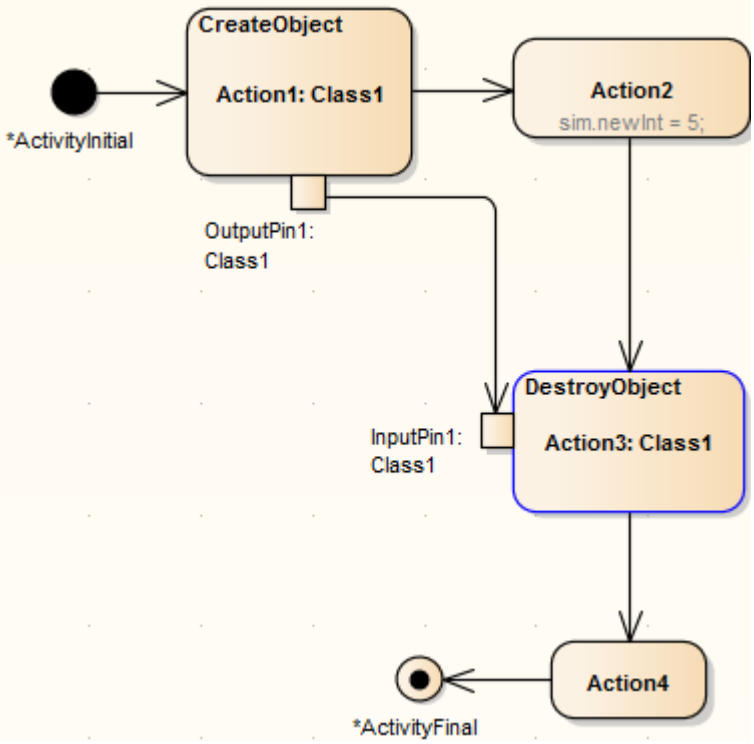
在您的仿真模型中创建或生成对象后，您可以定义行动以在过程中的任何时间点销毁这些对象。所有仿真物品在仿真完成后自动销毁。

销毁仿真模型中的对象有两种选择：

- 通过 DestroyObject行动元素动态销毁 Objects
- 在行动元素中使用JavaScript动态销毁对象

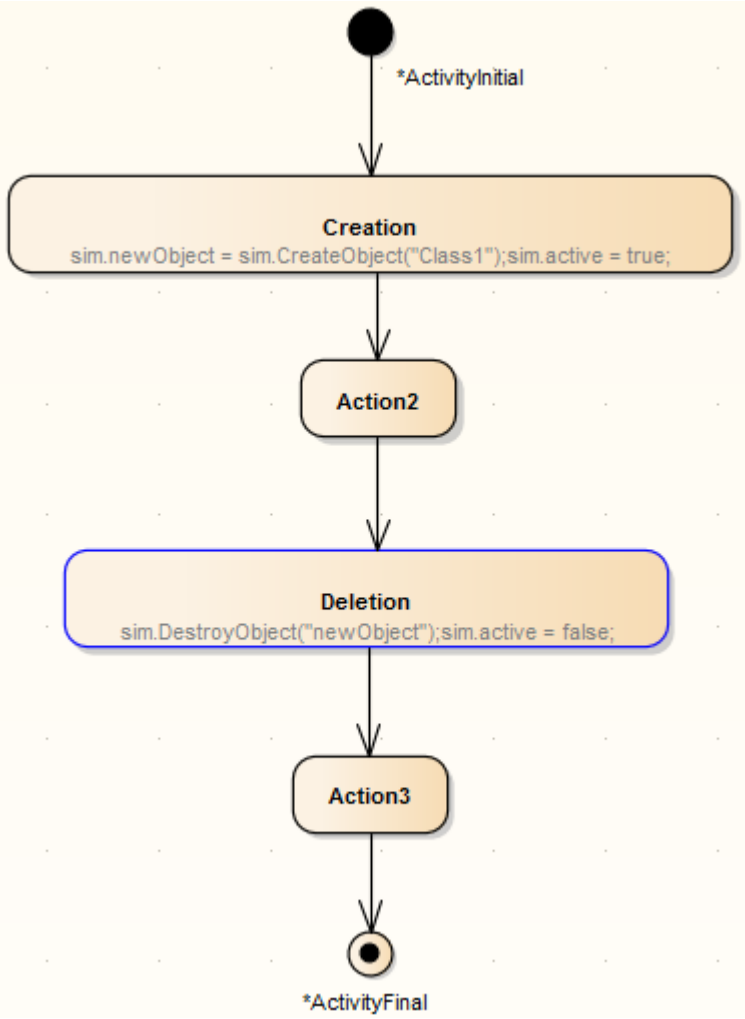
删除的结果可以在局部变量的变化中观察到，在局部窗口上。

通过 DestroyObject行动摧毁物件

节	行动
1	在您的活动图表上，从工具箱中图表一个 行动"图标，然后选择 其它| DestroyObject 的上下文菜单选项将其定义为 DestroyObject行动元素。
2	将 DestroyObject行动的分类器设置为物件是实例的类。 （高级|设置分类器）。 在行动上创建一个行动销，类型输入。
3	将输入行动销连接到来自上一个在行动上操作的行动的物件物件流连接器。在此示例中，对行动进行行动的最后一个物件是创建它的动作。 
4	在图上进行仿真。该过程将物件名称或值作为参数传递到输入行动销。当执行行动动作时，它会从模型中删除具有该名称或值的物件。 示例中，Class1的实例在Action4处理之前被专门销毁，但Action2的结果不受影响。

使用JavaScript销毁一个物件

在行动元素的 属性”对话框中，在 影响”页面的 影响”字段中，键入：
sim.DestroyObject（ 对象名”）
或者
删除 sim.objectFullName
例如：



注记

- 在任何一种情况下，您还可以通过标识执行销毁的行动的物件来销毁全局object（在流程之外创建的对象）；在端口行动的情况下，通过行动物件流连接器将物件名称从物件传递到销上的输入

用JavaScript进行动态仿真

Enterprise Architect的企业版、统一版和终极版提供了使用JavaScript评估仿真时间内的守卫、效果和其他方面的行为的上下文。这为您的状态或活动模型提供了完全自动化、智能的执行，并对断点、执行速度和仿真变量进行了精细控制。

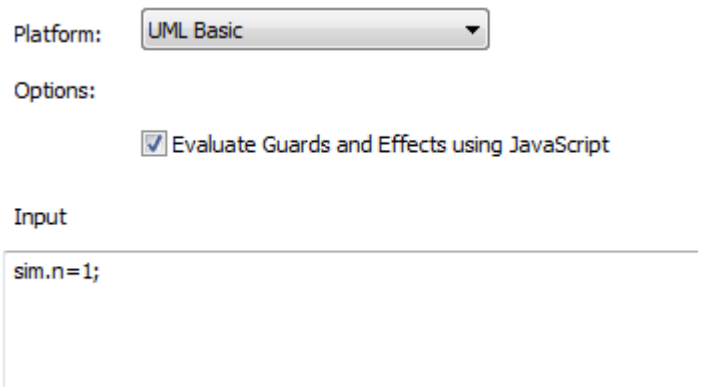
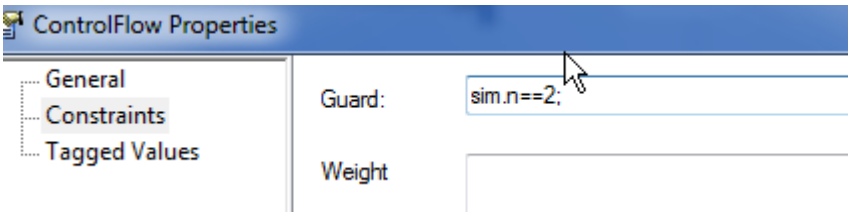
您可以编写使用任何变量的JavaScript 。为了使您能够通过用户界面显示这些变量的值，定义了两个内置对象 - **sim**和**this** - 其成员可以显示在 Local Variables 窗口（也称为本地窗口窗口）中。可以显示的变量示例如下：

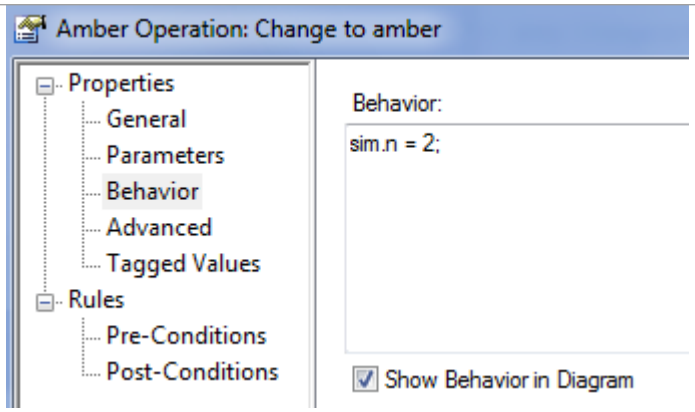
- sim.logger
- sim.顾客.姓名
- 这个.count
- this.账户.金额

推荐的约定是将所有未在 owning类中声明的全局变量或控制变量添加到**sim** object 。相反，将拥有分类器的属性添加到**this** object是正常的。

此处提供了一些使用JavaScript设置仿真行为的位置和方式的示例。 Enterprise Architect附带的 EAExample.eap模型中提供了更多示例。

使用JavaScript

环境	行动
分析器脚本	如果您在执行分析器窗口的 输入"字段中输入J JavaScript代码，该代码将被注入仿真并在仿真开始前执行。这对于建立 COM 变量、全局计数器、函数和其他初始化很有用。 
转移 和控件流卫士	这是仿真功能的主力。由于Enterprise Architect在仿真中评估每个节点的可能前进路径，它会测试 Guards 的传出转换和控制流，并且只有在有一条真实路径可遵循时才会前进 - 否则仿真被认为是 阻塞"和人工干预是必须的。您必须使用 "=="运算符来测试是否相等。 
元素行为	可以为状态定义进入和退出行为。此类代码将在适当的时间执行，并可用于更新仿真变量和进行其他分配。

	 您还可以通过模型中的物件实例和活动来模拟类的行为。
使用 COM	在Enterprise Architect的模拟器中实现JavaScript的一个非常重要的特征是它支持 COM 对象的创建。这提供了将正在运行的仿真与几乎任何其他本地或远程过程连接起来的能力，并且可以根据外部数据影响仿真，或者根据当前仿真状态潜在地改变外部世界的数据或行为（例如，更新机械Enterprise Architect外部的模型或软件仿真）。创建 COM 对象的语法如下所示： this.name="奇偶"; var logger = new COMObject("MySim.Logger"); 记录器.Show(); logger.Log("仿真开始");
使用求解器	Enterprise Architect中Anywhere有JavaScript代码的地方，例如在动态仿真中，您现在都可以使用名为 "Solver"（Solver类）的JavaScript构造与外部工具集成，并直接使用每个工具中的功能来简单直观地执行复杂的数学和图表功能。这些调用可帮助您轻松地在内置JavaScript引擎和每个环境之间交换变量。支持的两个数学库是 MATLAB 和 Octave。 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，如产品文档中所述。 作为JavaScript引擎的一部分，这些 Solver 类也可以立即被插件 访问插件 作家创建基于模型的JavaScript插件。 有关详细信息，请参阅*Octave Solver*、 *MATLAB Solver*和*Solvers*帮助主题。
有信号行动	可以使用JavaScript直接引发信号事件（触发器）。 BroadcastSignal() 命令用于引发可能影响仿真当前状态的命名触发器。例如，此片段引发名为 "CancelPressed"的信号（触发器）。 BroadcastSignal("CancelPressed"); 注记记名为 CancelPressed 的触发器必须存在于当前仿真环境中，并且必须具有唯一的名称。 您还可以使用其GUID调用信号。例如： 广播信号（"{996EAF52-6843-41f7-8966-BCAA0ABEC41F}"）；
IS_IN()	如果当前仿真在与传入名称匹配的任何线程中具有活动状态，则 IS_IN（状态）运算符返回True 。例如，为了有条件地控制执行，可以编写如下代码： if (IS_IN("WaitingForInput")) 广播信号（ 取消按下"） 注记该名称在所有上下文中必须是唯一的。

跟踪()	跟踪（语句）方法允许您在仿真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。 JavaScript仿真将截断超过定义的跟踪消息最大长度的字符串。

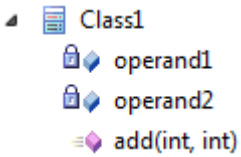
调用行为

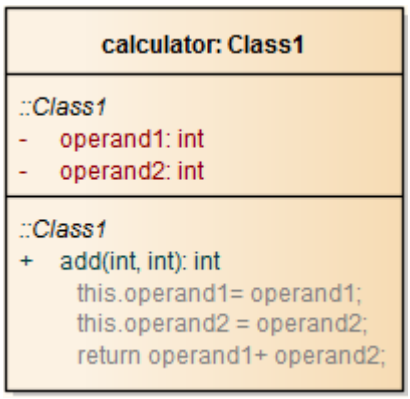
在模拟过程的过程中，您可以在模型中执行类（通过其仿真物件）或活动中定义的行为。在每种情况下，您都使用JavaScript来调用行为。

调用类的行为

模型中A类定义了您要模拟的行为。这种行为在类的操作的行为页面中定义。

例如，类旨在通过操作**add(int , int)**将两个整数相加。在这种情况下，整数是操作的参数，由类、操作数 1 和操作数 2 的属性定义。



节	行动
1	在操作的属性窗口中，选择 行为"选项卡并编辑 行为"字段以将JavaScript仿真对象（ *this*或*sim* ）应用于行为定义。 在示例中： this.operand1=操作数1； this.operand2=操作数2； 返回操作数 1+操作数 2
2	将类拖到您的仿真活动图上并将其粘贴为实例。 在示例中，该物件被称为 计算器"。为清楚起见，此处显示的元素设置为在图表上显示继承的属性和操作以及行为代码。 
3	在仿真图上，对于适当的行动元素，打开 属性"对话框，并在JavaScript中的 影响"页面类型中捕获和模拟物件的行为。 在示例中， JavaScript定义了一个值，该值将通过模拟来自物件的操作行为来提供，就像对两个提供的整数执行的操作一样。那是： sim.result=sim.calculator.add(7,9)
4	运行仿真，并在本地窗口中观察仿真进度。类行为最终体现在仿真结果中。 在示例中：结果 = 16。

调用活动的行为

一个活动元素可以有一个行为，由该元素中的操作定义。举个简单的例子，一个活动可能有一个名为 Get Result 的操作，行为返回 ÓN”;。

您可以在活动的子图（即活动的内部）中模拟此行为，并在适当的行动元素的 影响″字段中使用JavaScript语句。在示例中，这可能是：

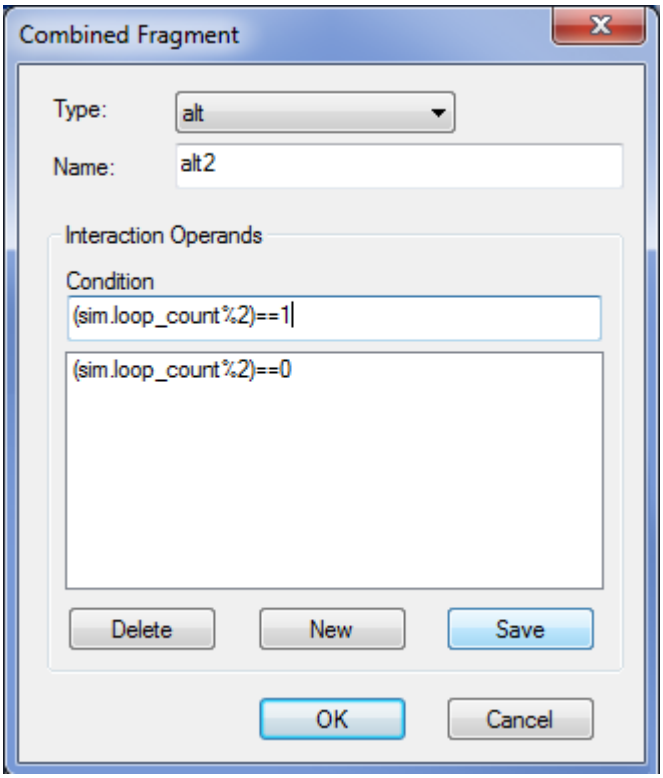
sim.result=this.GetResult();

该语句调用父活动的操作 GetResult 并将该操作行为的结果分配给 sim.result。您可以在本地窗口窗口中观察仿真的进度和模拟行为的结果，其中（在本例中）最终显示值结果 ÓN″。

交互的条件和信息行为

当你使用一个序列图，你可以模拟条件过程中的情景交互行为，来控制演示过程的仿真。
图表中A信息序列可以链接到操作，因此操作的行为也可以在仿真过程中使用。

交互条件

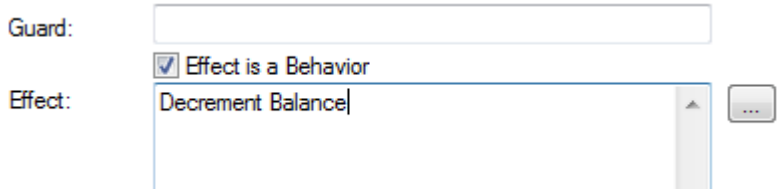
字段/列	描述
操作条件	交互条件是条件语句，只要模拟器确定下一步要走哪条路径，就会被评估。操作数条件通常具有以下特征： ● 在 组合片段"对话框中定义 ● 用JavaScript ● 可以参考仿真过程中定义的变量
添加操作数条件	添加操作数条件： 1. 双击CombinedFragment元素打开 "Combined Fragment"对话框。 2. 单击新建按钮。 3. 在 条件"字段中，键入条件的JavaScript 。 4. 单击保存按钮。 
评估语义	在执行期间，模拟器评估组合条件内的任何操作数条件； CombinedFragment的类型和评估的结果可以决定仿真继续走的路径。

防护条件

防护条件用于控制仿真流程，在仿真过程中执行附加动作或效果。

防护条件

概念	细节
警卫	警卫是条件语句，只要模拟器必须确定下一步要走的路径，就会对其进行评估。守卫通常具有以下特征： ● 定义了过渡和控制流，以控制仿真如何进行 ● 用JavaScript ● 可以参考仿真过程中定义的变量
添加警卫	警卫是在转移 上定义的转移 或选定连接器的 属性"对话框中的 流控件"。守卫条件A是一段JavaScript ，其计算结果为True或False 。例如，这里有一个条件语句，表示当前变量（Balance）大于零。注记使用前缀 *"this"*表示该变量是当前类时间的上下文。 Guard: this.Balance >0; Effect is a Behavior Effect:
评估语义	在执行期间，模拟器将检查所有可能的前进路径并评估任何保护条件。该评估可以确定： ● A有效的前进路径评估为True ；模拟器将遵循该路径 ● 存在两条有效路径；模拟器将阻塞，等待通过控制台窗口进行一些手动输入以解决死锁 ● 不存在有效路径；与找到两条路径时的响应相同 -上下文使用控制台窗口等待用户更改执行时间 ● 没有路径评估为True但存在默认（未保护路径）；模拟器将采取无人看管的路径
效果	效果是在特殊时间执行的已定义行为： ● 进入状态 ● 从某个状态退出时 ● 从一种状态过渡到另一种状态时（过渡效果） 效果可以是一段JavaScript代码，也可以是当前模拟中对另一个行为元素的调用。
JavaScript效果	JavaScript效果可能类似于A示例，其中 Balance 变量递减： Guard: Effect is a Behavior Effect: this.Balance--;

调用行为效果	在此示例中，影响是呼叫行为效果。在这种情况下，它调用了一个模型活动，即在别处定义的名为减量平衡的模型活动。然后模拟将进入该图表/行为并继续执行，直到返回到调用影响的点。 
效果的执行顺序	在可能涉及从深度嵌套状态转换到不同父上下文的其他深度嵌套状态的复杂模拟中，重要的是要考虑有关执行顺序的这些规则： • 离开嵌套时间时遇到的所有退出操作（效果）按照嵌套上下文到嵌套最浅的顺序执行 • 接下来执行在转换上定义的所有动作（效果） • 最后，所有的入口效果都是从嵌套最上下文的时间到最深嵌套的时间执行的 所以基本规则是：所有退出动作，然后是所有转换动作，最后是所有进入动作。
关于JavaScript变量的注记	在仿真执行期间要访问和引用的JavaScript变量属于： • sim（例如，sim.pedestrianwaiting）- 通常用于全局仿真变量，或 • this（例如，this.CustomerNumber） - 通常用于引用拥有类属性 这对于让JavaScript引擎知道您指的是仿真变量而不是在例如基本计算期间使用的简单局部变量非常重要。您可以创建任意范围和深度的仿真变量 - 例如，this.customer.name 是合法的限定名称。

触发器

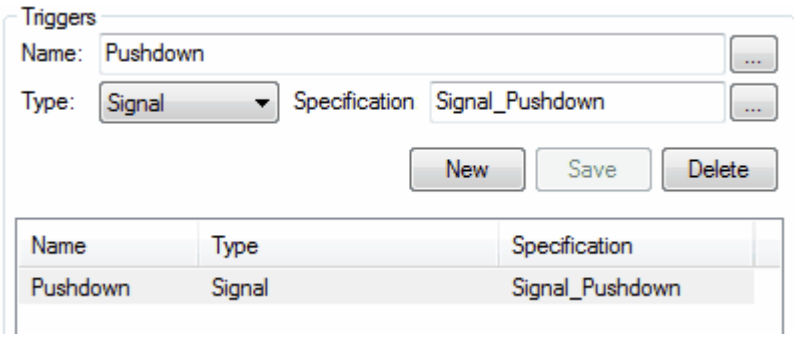
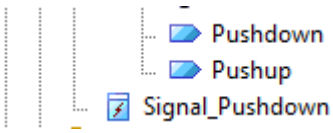
触发器表示可以激活离开当前状态的转换的信号和事件。触发器可能代表真实世界A信号或事件，例如：

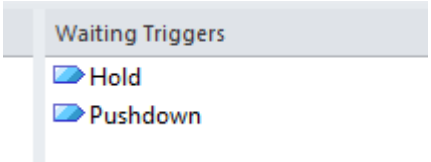
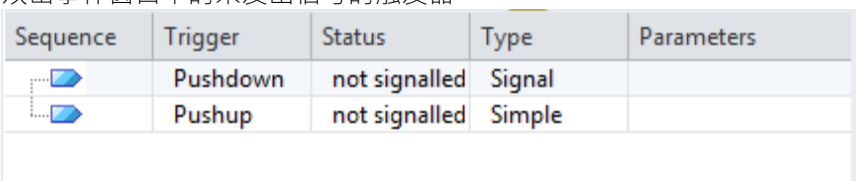
- A按钮被按下
- 收到A消息
- A踏板
- A开关被抛出
- 正在进入或退出A并发区域中的状态

让触发器产生影响

- 必须定义在模拟接收到信号或事件时将触发的转换
- 当前仿真状态或其父级必须具有接受该触发器的传出转换
- 激活的转换必须是无人看守的，或者有一个将评估为True的看守

管理触发器

行动	细节
创建触发器	触发器的元素要么被创建为信号的实例，要么被创建为匿名事件。触发器连接到 转移 "中的转换转移 属性对话框，如下所示。在此示例中，已根据信号 "Signal_Pushdown"定义了一个名为 触发器"的简单示例。 ● 省略类型和规范细节会导致简单的匿名触发器。 ● 如果需要参数，这些参数在信号上定义并且必须在事件触发时提供  触发器A出现在浏览器窗口的 项目"选项卡中，如下所示： 
使用触发器	触发器通过将它们连接到转换来部署，如前面的示例中一样，并在模拟期间通过根据需要将它们 触发"到正在运行的模拟中来使用。 使用触发器时应考虑以下几点： ● A发出有效触发器信号或触发之前，不会发生 触发"转换

	• 当收到触发器时，它将激活依赖于该触发器的所有当前等待转换（即，触发器是广播的） • 对当前子状态的所有父级的所有触发器进行评估；这允许父状态在必要时退出所有子状态 • 一旦在模拟中使用，触发器就会被消耗，如果再次需要，必须重新触发 • 可保存触发器集并手动或自动触发，以方便不同事件模型下的自动模型模拟
射击触发器	触发触发器意味着在当前模拟中发出信号或激活触发器。这可以根据当前仿真的状态和并发性激活零个、一个或多个等待转换。 触发触发器可以通过多种方式实现。最有效的是 等待触发器"列表。 在模型仿真过程中，如果模拟器由于所需的触发器不可用（触发）而陷入僵局，则所有可能的候选触发器列表显示在触发器事件仿真事件窗口的 等待"列表中。  双击此列表中的触发器会将其触发到仿真程序中。其它方式启动触发器包括： 1. 双击事件窗口中的未发出信号的触发器。  Sequence | Trigger | Status | Type | Parameters Pushdown | not signalled | Signal Pushup | not signalled | Simple 您还可以在这些事件上使用上下文菜单来发出未发出信号的事件，或重新发出先前已触发的事件的信号。 2. 使用转移 的上下文菜单转移 需要启动并选择 触发器仿真的信号"菜单选项。

行动行为按类型

你可以通过定义（甚至重新定义）它的类型来改变一个行动元素发起的行为。在仿真中，您可以使用本行动中描述的类型和组中的表来应用和观察许多不同的行为。

行动类型

类型	描述
物件行动	物件行动以特定方式对object进行操作，例如创建、销毁或读取object 。他们包括： ● 创建对象 ● 销毁对象和 ● 阅读自我
可变行动	变量行动在运行时有一个以标记值变量形式与object名称的值关联的变量。它们不仅以object的形式提供变量，还以object的属性（例如属性或端口）的形式提供变量。他们包括： ● ReadVariable ● 写变量 ● ClearVariable ● 添加变量值 ● 移除变量
结构行动	结构行动作用于结构特征，即活动的属性或对象的分类器的object 。他们包括： ● 读取结构特征 ● 写结构特征 ● 清晰的结构特征 ● 添加结构特征值 ● 移除结构特征值
调用和接受事件行动	调用和接受事件动作定义了事件的触发器和信号。他们包括： ● 发送信号 ● 广播信号 ● 接受事件 ● 发送对象 ● 呼叫行为 ● 呼叫操作 ● 接听电话
杂项行动	行动评估一个值；它必须有一个输入值和一些评估代码作为它的行为或效果。

结构活动仿真

行为模型中更复杂的结构之一是结构活动，它对嵌套结构或评估和执行过程中的一系列动作进行建模。结构活动的评估类型有Conditional节点和Loop节点，你可以很容易地模拟这两种节点。

条件节点

A条件节点基本上由一对或多对测试/体分区组成，每一对被称为一个子句。测试分区由测试条件的活动图元素组成，如果满足该条件，则执行体分区中的进一步活动图元素以产生结果。

如果有一个子句，条件节点要么输出体分区的结果，要么不输出结果。如果有多个子句，则控制从一个测试流向下一个测试，直到满足某个条件并执行一个体分区以产生结果，或者所有测试都失败。

仿真目前支持在属性窗口的 条件"选项卡中使用 确定"复选框设置。其他两个复选框设置被忽略。如果 'Is Assured' 复选框是：

- 选中，至少要满足一个测试，所以执行它的体并输出一个结果；如果没有满足测试并且没有结果输出，则条件节点被阻塞并且处理不能继续超出它
- 未选中，可以满足一个测试并输出一个结果，但是如果没有满足测试并且没有结果输出，超出条件节点仍然可以继续处理

您可以通过键入JavaScript *sim* 来模拟一系列可能的路径和结果。在每个条款的每个分区内的行动元素的 影响"字段中定义或导致特定测试结果和体结果的语句。这些模拟。语句必须标识设置的条件节点、子句和输出销的完整路径。例如，在测试一个人是否有资格成为老年人时：

if (sim. Person .age >=65)

sim.AgeCondition.Clause1.Decider1=true；

else

sim.AgeCondition.Clause1.Decider1=false；

条件节点称为*AgeCondition* ，测试在*Clause1*中，该测试的 OutputPin 是*Decider1* 。

Loop节点

循环结构活动节点通常表示 While、Repeat 和 For 循环语句A建模等价物。每个 Loop节点有三个分区：

- 设置 - 启动要在循环退出条件中使用的变量；它在进入循环时执行一次
- 测试——定义循环退出条件
- 体- 重复执行，直到测试产生False值

您可以通过将活动图表元素从工具箱页面拖入设置、测试和体分区来定义循环节点。体分区可以包含相当复杂的元素节点结构，定义了 Loop 在这个过程中实际产生的内容。

Loop节点有多个行动销：

- 循环变量（输入） - 要通过循环处理的初始值
- 循环变量（输出） - 执行测试的变化变量
- Decider -测试分区内的一个输出销，每次执行测试后都会检查其值，以确定是否执行循环体
- 体输出-体分区中处理的输出值，为循环的下一次迭代更新循环变量输出销，以及
- Result -测试分区最后一次执行的值（也就是最后一次执行体分区传回来的值）

您可以通过输入JavaScript *sim* 通过循环模拟不同动作和输出的效果。在每个分区内的行动元素的 影响"字段中定义或导致特定测试结果和体结果的语句。这些模拟。语句必须标识正在设置的 Loop节点和输出销的路径。例如，在测试分区中的一个行动中：

sim.LoopNode1.decider = (sim.LoopNode1.loopVariable>0);

活动返回价值仿真

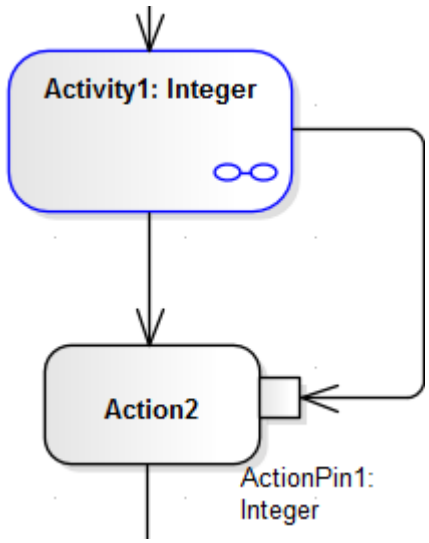
一个活动可能会产生一个返回值作为它所代表的过程的输出。您可以在以下三种情况下模拟该返回值如何传递到流程的下一个阶段：

- 活动只是产生一个返回值，直接传递给下一个行动
- 活动具有一个或多个活动参数 - 在图表上由活动节点表示 - 接受输入值到活动的子活动或保存来自活动行动输出值，并且输出活动参数收集并传递返回值
- 活动由复制活动行为并向前传递返回值的行动实例化

活动返回价值通输出

（这种方法是Enterprise Architect模拟所独有的，模仿活动参数的效果而不必存在。）

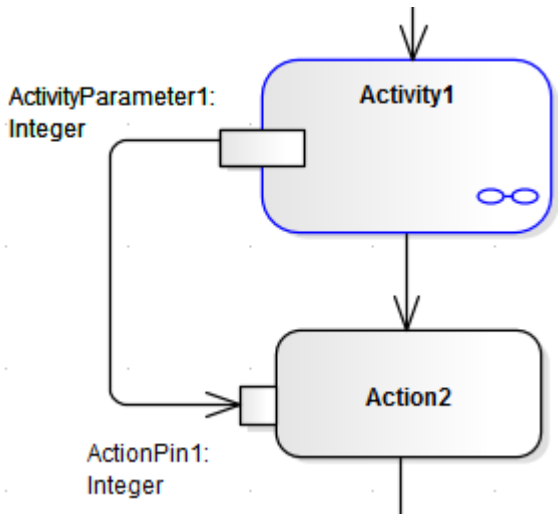
活动有一个返回值，它通过一个行动物件流连接器从活动元素销行动



您可以通过设置一个简单的JavaScript语句来设置活动的子元素中的返回值（例如行动=12;）来模拟这一点，然后运行模拟，在本地窗口中销传递给活动元素的值窗户。

活动参数输出

如果该活动有一个活动参数，则其值传递给相应的活动节点，然后通过一个物件流连接器，行动给过程中下一个活动的输入活动行动销，如图所示：

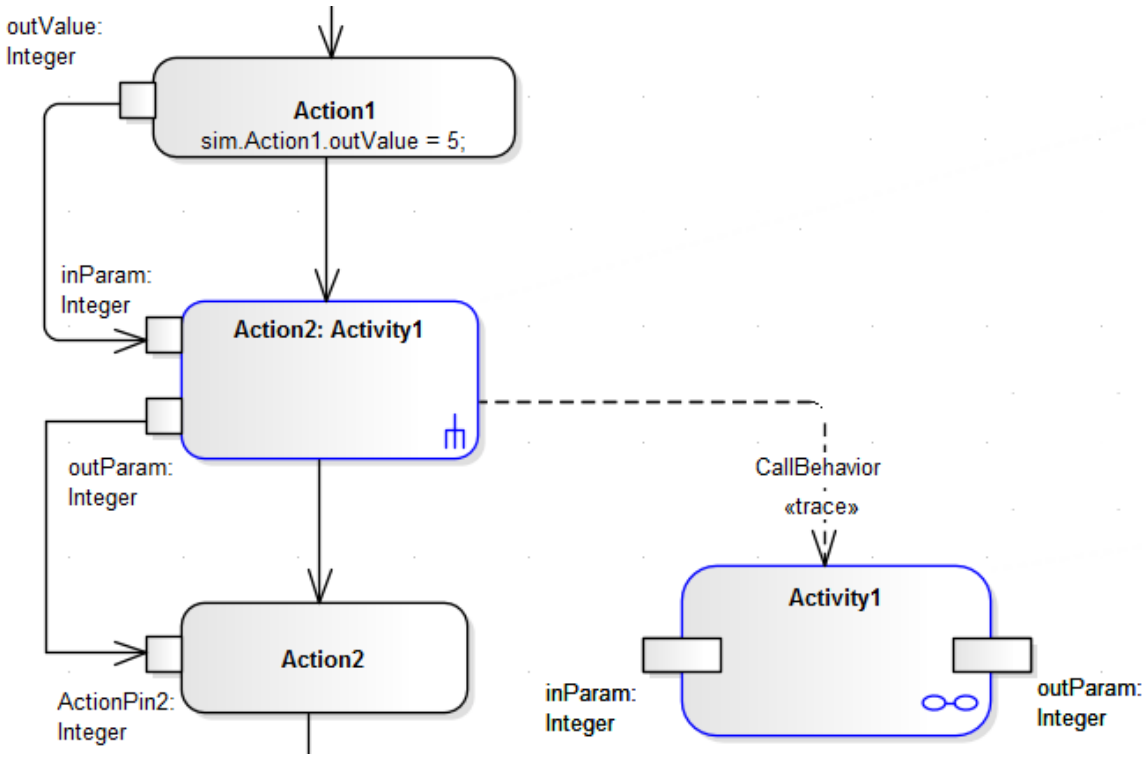


在本地窗口窗口中，您可以观察参数的默认值传递给行动销，也可以在活动的子行动中使用JavaScript来模拟活动中的值的更新。例如：

this.ActivityParameter1=20;

呼叫行为行动

一个活动可能会在一个进程中使用多次，在这种情况下，您可能希望每次都生成一个单独的活动实例。您可以使用行动来创建活动object并执行其行为。输入和输出活动参数与行动上的相应输入和输出行动销（参数）绑定。



当您模拟包含活动的过程部分时，您提供一个输入值（如在行动1中），该输入值传递到行动行动上的输入行动销中，从而创建活动的一个物件。行动执行活动的行为，使用输入行动销作为输入活动参数，使用输出行动销接收返回作为输出活动参数。然后使用行动物件流连接器将活动返回值传递给下一个行动上的行动销。您可以在活动的行动中提供JavaScript语句来作用于输入值并生成返回值，例如：

sim.buf=this.inParam;和

this.outParam=sim.buf + 11：

仿真事件窗口

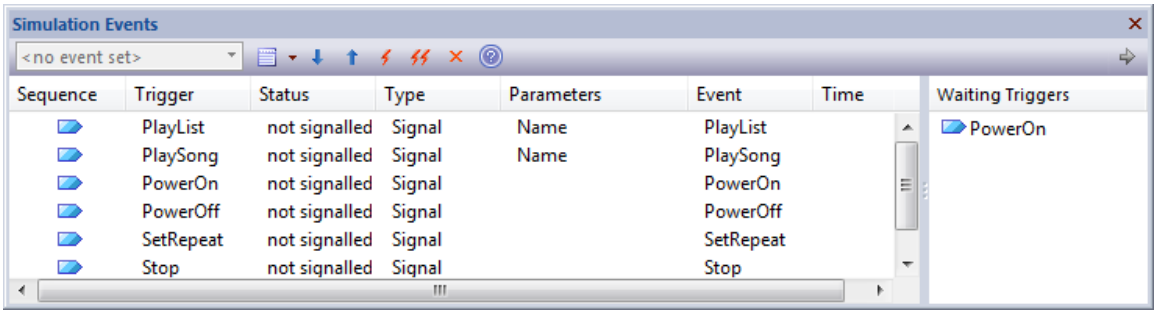
仿真事件窗口是您在仿真中管理触发器和事件集的地方。其主要功能是：

- 为模拟添加、删除和重新排序一组触发器
- 显示当前运行模拟的已触发、丢失和等待事件列表
- 提供选项启动任意任意触发器进入当前模拟
- 提供一个方便的 等待"模拟正在等待的触发器列表
- 保存触发集以供以后在手动和自动仿真中使用
- 接受从浏览器窗口拖入当前列表的触发器
- 在触发前输入等待触发的触发参数

由于在模拟中消耗了触发器，它们的状态和位置记录在仿真事件窗口的主体中。

您可以将触发的触发器的log保存为触发器集或事件集，以便在另一个仿真运行中重新应用，您可以手动或自动执行。有关构建和使用触发器集合的更多信息，请参阅触发器集合和自动触发主题。

此图说明了仿真事件执行过程中的窗口。



访问

功能区	仿真>动态仿真>事件

列详细信息

字段/列	行动
序列	在模拟期间和模拟之后，指示触发或预期触发触发器的序列中的位置。注记如果触发的触发器超出序列，它将被移动到已发出信号的事件部分的底部。
触发器	触发器的名称 -触发器用于启动事件的简单方式。
状态	表示触发器的状态。值可以是： • used - 触发器已被触发并且处理已通过 • lost - 触发器已在列表中触发，但没有效果 • 已发出信号 - 一个或多个转换触发并消耗了触发器 • 未发出信号 - 触发器尚未触发

类型	指示触发器的类型。目前仅支持： ● 信号 ● （无类型）匿名触发器
参数	对于一个触发器的信号，最初显示信号规范触发所需的参数。例如， 登录 "信号可能包括用户名和密码参数——每个触发的调用可以使用不同的参数。 每次模拟触发触发器时，系统都会提示您输入值。当触发器设置为未发出信号时，您还可以直接在列表中编辑值。 参数对于测试模拟中的条件逻辑以及模拟来自模拟外部的各种输入和数据非常有用。
事件	为一个： ● 信号的触发器，标识信号规范 ● 对于匿名触发器没有任何价值
时间	触发信号的模拟时间。注记这是一个绝对（真实世界）时间，而不是相对模拟事件时间。
等待触发器	触发器可从当前状态中选择的简单状态，包括在单个转换中可能有多个触发器的状态。双击触发器以将其添加为当前事件序列中的下一个触发器并发出信号。 您可以通过单击面板正上方的灰色箭头来显示和隐藏此面板。

工具栏项

选项	行动
Set1	使用此下拉列表来选择和使用先前定义的触发器集。 在运行模拟之前，选择一个先前定义的触发器集以用于下一次模拟运行。您可以通过选择 <no event set> 选项来选择不使用触发器集。
	单击以创建和删除触发器集： ● 保存集——将当前触发列表保存为新的触发集；系统提示您输入新集合的名称 ● 将集另存为 - 以新集名称创建当前集的副本 ● 删除选定集 -删除当前触发集 ● 删除所有图表图表-删除当前图表的所有已保存触发器集
	在触发器的触发序列中将所选触发器向下移动一行。 如果所选行下方没有信号触发器，则此选项不可用。
	在触发器的触发序列中将选定的触发器条目向上移动一行。 如果所选行上方没有信号触发器，则此选项不可用。
	单击启动选定的触发器。您也可以通过双击触发触发器。
	单击以打开和关闭自动触发。

	自动触发将按顺序触发触发器集中的未发出信号的触发器。如果您的设置与有效的执行路径匹配，则模拟将运行。输出序列或未使用的触发器将 丟失 ″。 断点会暂停自动触发，您需要单击下A触发器才能恢复自动触发模拟。
	从列表中删除选定的触发器。

上下文菜单选项

选项	行动
信号	信号，或触发，选定的未发出信号的触发器。
删除选定	从序列中删除未发出信号的触发器。
重新选择信号	再次触发已使用或已发出信号的触发器。
全部设置为无信号	将所有已使用或已发出信号的触发器设置为未发出信号。
触发器列表	清除窗口中的所有触发器，无论它们的状态如何。

等待触发器

当模拟达到任何状态（对于任何线程）需要状态触发器地继续进行的点时，模拟将被有效暂停并且控制返回给系统。模拟现在正在有效地等待某种形式的事件（现实世界的信号）继续进行。等待触发器列表有助于触发器应该手动发出信号。

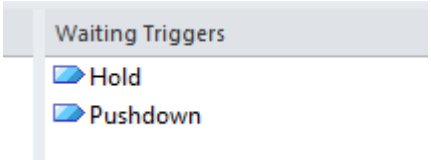
访问

功能区	仿真>动态仿真>事件 右侧窗格列出可用触发器。

仿真事件窗口的触发器名单如下：

- 在每个仿真周期中填充任何触发器的说明，如果发出信号就会立即生效
- 填充离散集（任何重复都不会显示为简单的触发器有效地广播到所有转换）
- 通过双击感兴趣的触发器来激活
- 包括所有可能的触发器——包括那些激活当前嵌套状态的父级的转换

这个例子表明，当前的模拟已经达到了触发器可能影响执行流程的程度。



由于简单的触发器及其影响，该列表可以同样有效地引用这些示例情况中的每一个：

- 单个状态A两个输出转移，分别等待Hold和Pushdown；触发其中之一将激活模拟中的相关转换
- 单个状态A两个或多个可能触发相同的转换，例如安全摄像头被运动检测器、声音检测器或热检测器打开
- 两个（或更多）线程（并发区域）每个都有状态等待保持或下推；触发其中一个触发器将导致线程等待该触发器继续进行，而其他线程将保持阻塞状态
- A状态正在等待其中一个触发器，而父状态正在等待另一个；触发触发器将导致关联的转换被触发，并且子或父相应地继续
- 这些的任意组合

重新信号触发器

可以重新发出简单的信号作为触发器其他触发器实例以发出信号的快捷方式。

访问

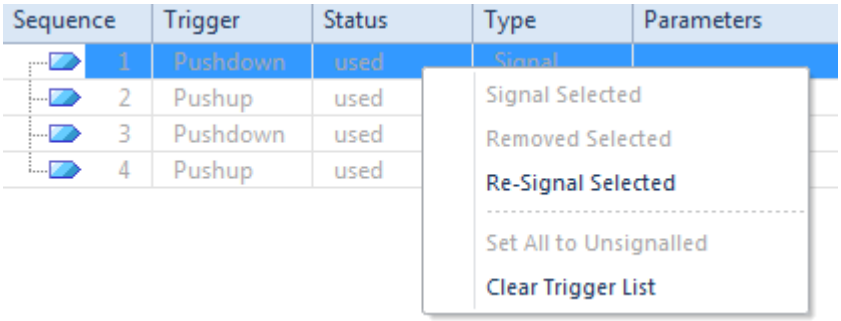
显示仿真事件窗口，然后右键单击该窗口中的触发器并选择 重新发出选定信号”选项。

功能区	仿真>动态仿真>事件>右键单击现有触发器>重新选择信号

触发器列表

仿真事件窗口包含已触发的触发器事件列表。通过右键单击触发器你想再次发出信号的简单，你可以使用时间来上下文重新发出信号。

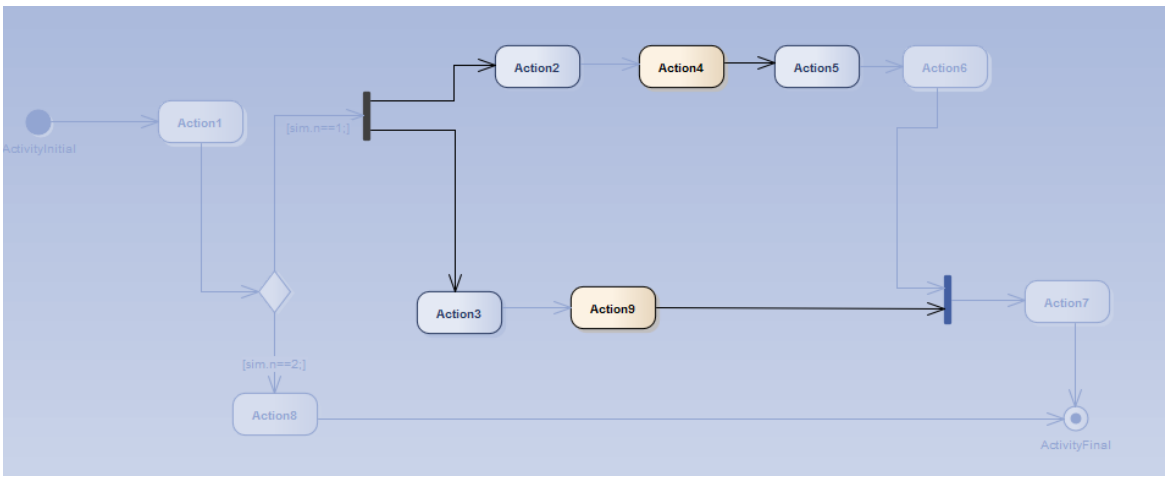
此图像演示了重新发送信号的作用。当重新发出信号时，会制作一个新副本并将其放置在已发出信号的触发器列表的末尾，并在此处再次自动触发。



多线程-分叉和汇合

模型模拟器提供了使用分叉和汇合节点处理多线程模拟的能力。

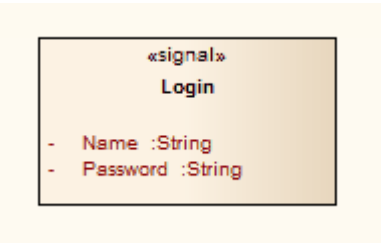
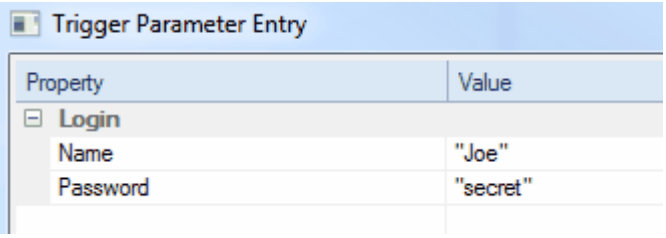
- 在示例中，当前执行点已分叉为两个线程，每个线程都有自己的活动节点
- 随着这个例子的进行，下层分支将在汇合节点等待，直到上层分支完成所有行动
- 一旦两个线程合并为一个线程，仿真将作为一个线程继续进行，直到完成
- 当自动单步执行时，每个线程将被视为在一个模拟 周期”中执行单个步骤 - 尽管当单步执行或在断点时，行为是在线程之间交替执行，因为每个线程接收处理时间
- 注记在示例中， 调用堆栈窗口将显示两个活动线程和一个 pa”线程；一旦线程合并，将返回单线程执行
- 另请注记，局部变量在所有线程之间是共享的（全局的）；如果要在线程上仿真私有变量，则必须在每个线程开始时创建新的仿真变量 - 使用现有全局数据预加载此类变量

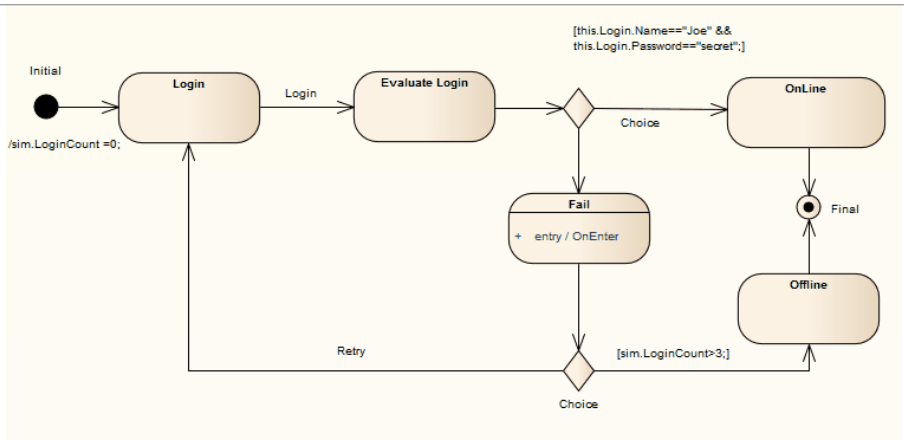


触发器参数

触发器是在触发时与触发器一起传递给模拟的参数。它们允许根据在运行时通过触发的触发器（事件）传递到模拟的变量和数据来指定复杂的行为决策。

参数

参数	细节
介绍	要使用触发参数，您： ● 首先创建一个具有适当属性的信号元素，这些属性将在运行时成为您的参数 ● 在图表中合适的转换上，创建一个基于之前创建的信号的触发器 ● 在运行时，会提示输入合适的参数——然　它们与触发器一起传入
信号	信号元素是A模板或规范，可以从中构建实际的触发器。这个例子有两个参数，一个名称和一个密码。这些将在执行时手动或作为预定义触发器集的一部分填充。 
触发器参数	触发器'提示'要求为每个参数提供合适的值。注记你需要用双引号将字符串括起来，否则解释器会认为你指的是其他变量。 
示例图表	这是一个使用触发参数的示例图。在 Evaluate Login状态下，模拟检查作为触发参数传入的变量，并决定是接受凭据还是拒绝凭据。


[this.Login.Name=="Joe" &&
this.Login.Password=="secret";]
Initial
Login
Login
Evaluate Login
Choice
OnLine
/sim.LoginCount =0;
Fail
+ entry / OnEnter
Final
Offline
Retry
[sim.LoginCount>3;]
Choice

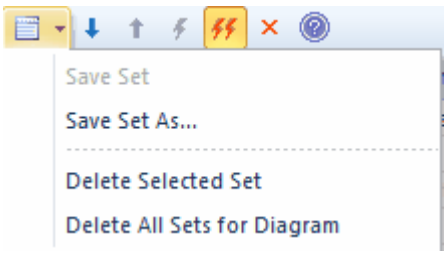
触发器的设置和自动射击

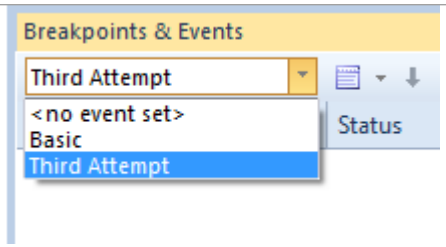
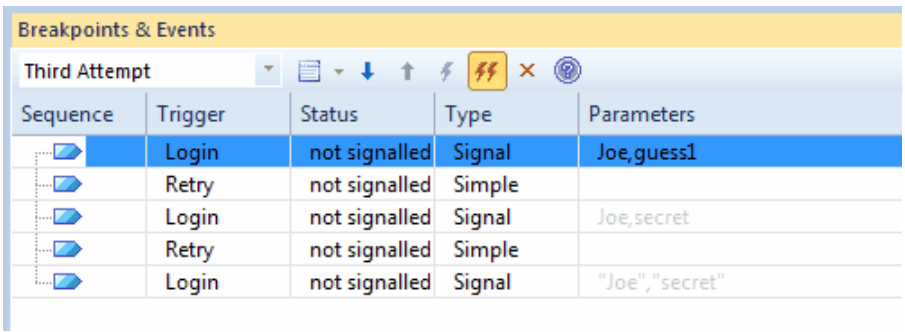
触发器集合是自动化和简化仿真模型的执行、测试和验证的有效手段。通过重复使用触发器集（带或不带参数），可以快速有效地遍历许多模拟场景，无论是手动还是使用 自动触发”工具自动进行。

访问

功能区	仿真>动态仿真>事件

关于触发器

方面	细节
触发器套装	• 与关联图一起存储 • 由一组序列的列表触发器 • 可以在必要时包含触发器参数 • 可以根据需要手动双击触发器启动 • 可用作 自动触发”行为的一部分以自动执行 • 由仿真事件窗口管理
管理集	可以通过手动将触发器拖动到活动触发器列表中来创建触发器的集合，然后使用 管理触发器”下拉菜单保存新集合。 还可以将在单个模拟设置期间建立的一组触发器保存为新组。这便于通过模拟创建多个测试路径，基于为每个测试用例保存手动触发的触发器。  您还可以删除一个集合并删除当前图表的所有集合。 也可以加载一个集合，修改参数和/或触发顺序，并用新名称保存集合。这是快速创建一套模拟测试脚本的便捷方法。
使用集合	要使用触发器集，您首先从触发器集下拉列表中按名称选择它，如本示例图像所示。选择后，它会加载带有定义的触发器集的触发器列表窗口。 注记特殊项<*no event set*>表示当前未选择任何集合。在每次模拟开始时，如果选择了一个集合，它将重新加载以进行下一次运行。如果选择<*no event set*>，触发器列表将被清除。

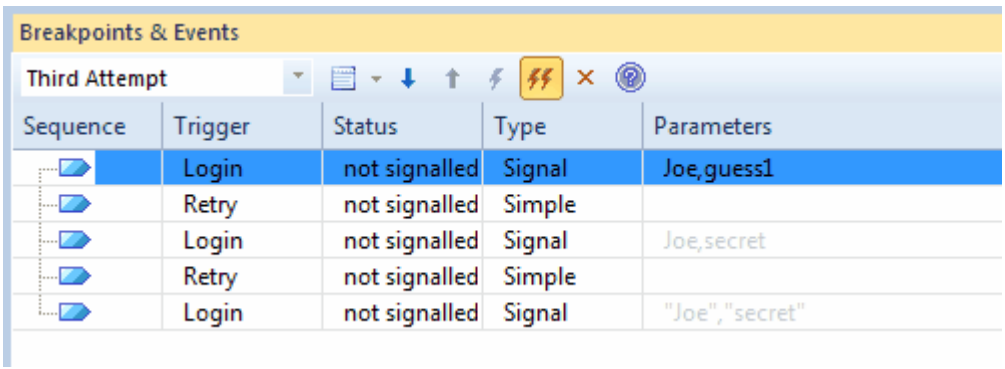
	 选择触发器集并加载触发器列表后，您有两个选项： ● 根据需要手动触发触发器 ● 使用自动触发特征来完全自动化模拟 
自动射击	自动触发是一种简化模拟的便捷方式。加载触发器集后，如果您选择自动触发按钮  ，则Enterprise Architect将在模拟陷入僵局时自动拾取等待的触发器。在实践中，这意味着与模拟路径完全匹配的触发器集将自动运行，无需您的干预。 由于您可以保存任意数量的具有不同路径和触发参数的触发集，因此您可以有效快速地测试和处理许多不同的场景。
自动触发规则	当模拟在启用自动触发的情况下运行时， Enterprise Architect将等待直到达到模拟 阻塞"或稳定的点，等待一个或多个触发器来推进模拟。届时，列表中第一个未触发的触发器将被拾取并触发到模拟中。结果取决于相关性，也可能取决于触发器的参数。 ● 如果触发器与 等待"触发器匹配，则立即消耗它并进行模拟 ● 如果触发器不匹配 等待"触发器或可能的父转换，则触发器 丢失"并且模拟保持在当前状态；这对应于一种场景，例如用户连续多次按下 开启"按钮 - 除了第一次按下之外没有其他效果

使用触发器来仿真一个简单的事件序列

作为触发器集有用性的一个简单示例，请考虑此示例触发器集和关联图。

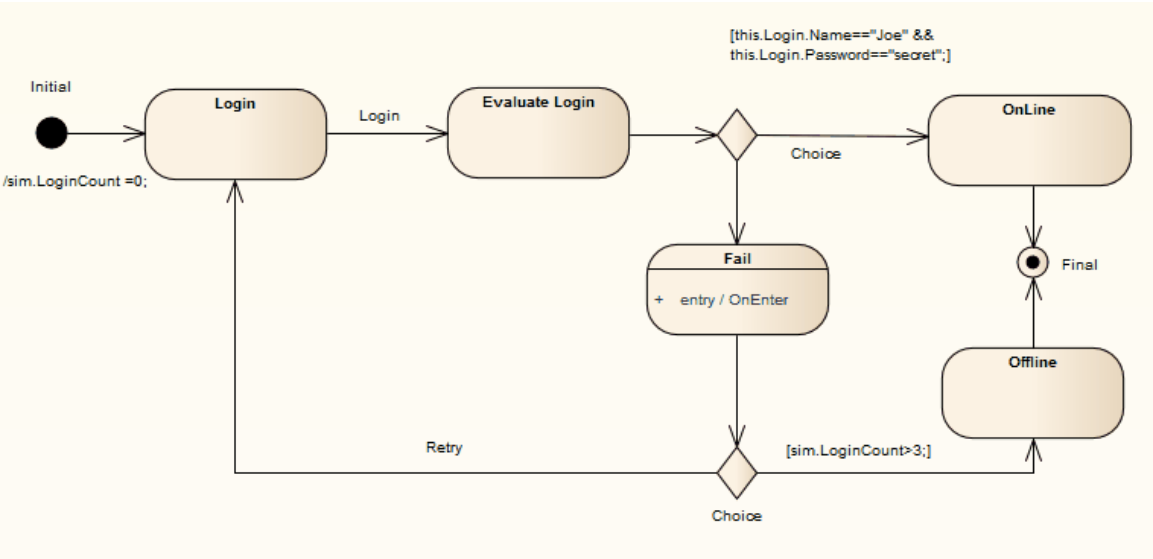
在此示例中，我们使用用户名和密码模拟了一个简单的 "三击即出"登录过程。成功路径是等待名字 "Joe"和密码 "secret"。（注记- 引用字符串的参数用引号括起来非常重要，否则解释器认为该名称指的是模拟中的另一个变量。）

- 通过1尝试乔和猜测1 - 失败
- Pass 2 尝试*Joe*和*secret*，但因为它们指的是变量，而不是字符串 - 这也失败了
- Pass 3显示了引用触发参数的正确方法，模拟成功



Sequence	Trigger	Status	Type	Parameters
	Login	not signalled	Signal	Joe,guess1
	Retry	not signalled	Simple	
	Login	not signalled	Signal	Joe,secret
	Retry	not signalled	Simple	
	Login	not signalled	Signal	"Joe","secret"

这是一个模拟需要用户名和密码对的登录过程的简单图表。



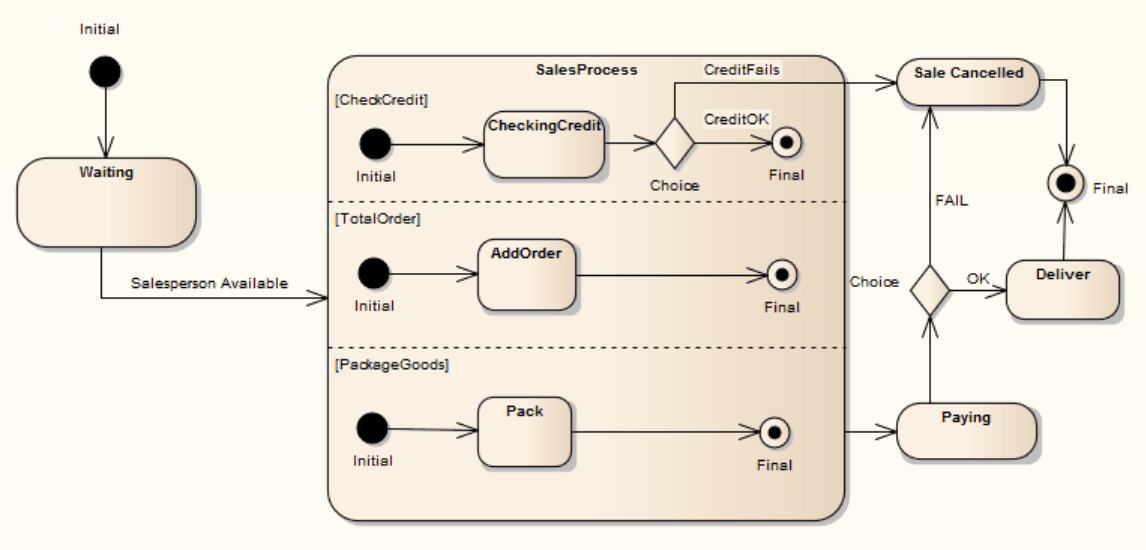
多线程-并发状态区域

状态中的并发区域表示在一个整体父状态中并行发生的状态变化和处理。当一个区域引发事件或修改另一个区域所依赖的模拟变量时，这尤其有用。例如，一个区域可以包含一个模拟计时器，该计时器在设定的时间间隔内引发事件，从而调用其他区域内状态的状态变化。

并发区域与分叉和汇合基本相同，逻辑和处理规则相似。

在示例中：

- 当转换到 SalesProcess 时，每个区域同时激活
- 检查信用，订单总额和所需的货物打包
- 但是，如果选择失败，这会触发到 Sale Cancelled状态的转换；注记当这种情况发生时，整个父状态和所有拥有的区域都会立即退出，而不管它们的处理状态
- 如果Credit选择成功，则区域移动到最终状态，一旦其他区域都达到自己的最终状态，则可以退出父状态

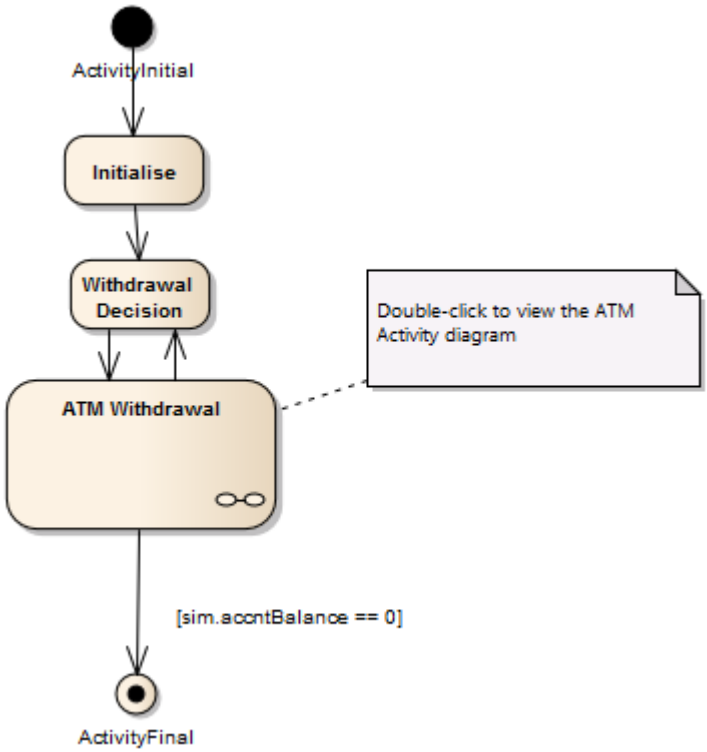


使用复合图表

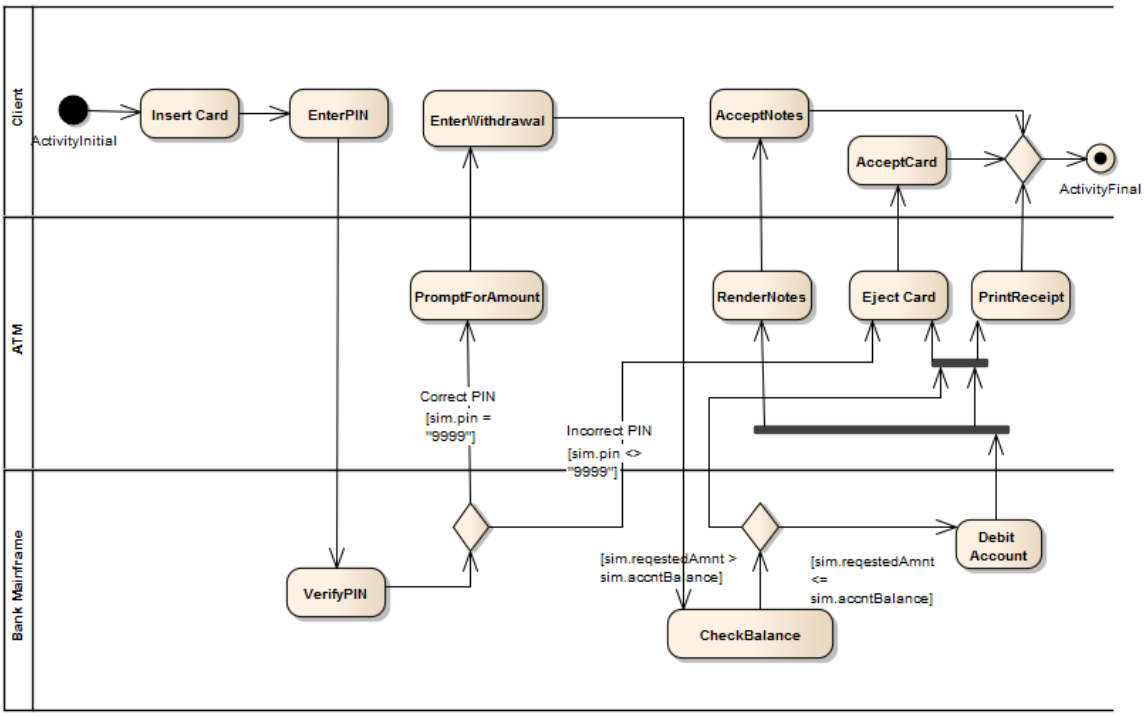
如果要模拟包含在不同图上表示的分支的处理（例如，为了降低主图上的复杂性，或者隐藏仅在异常下操作的处理区域），可以使用复合元素来表示并访问其子复合图上的分支。当您运行模拟并到达复合元素时，它会打开子图并对其进行处理，然后再返回（如果适用）到主处理路径。这是在复杂流程中遵循处理路径的绝佳方法，用复合活动元素表示流程的各个部分，这些元素在各自的子图中扩展了实际处理。您可以有多个复合元素来表示流程的不同阶段或分支。

需要注意的一个方面（这将通过模拟失败来揭示）是让多个线程在不同的图表上同时处理。如果模拟也跟随当前图表上的另一个线程，则模拟无法传递到新图表。

此图提供了 ATM 现金提取流程的概述：



ATM取款活动是一种复合元素。如果您双击它，您将打开并显示子图，这是提款过程的更详细的细分。同样，模拟将打开并处理子图。


Client
ATM
Bank Mainframe
ActivityInitial
Insert Card
EnterPIN
EnterWithdrawal
AcceptNotes
AcceptCard
ActivityFinal
PromptForAmount
RenderNotes
Eject Card
PrintReceipt
Correct PIN
[sim.pin =
"9999"]
Incorrect PIN
[sim.pin <>
"9999"]
VerifyPIN
[sim.reqestedAmnt >
sim.acontBalance]
[sim.reqestedAmnt
<=
sim.acontBalance]
Debit
Account
CheckBalance

Win32用户接口仿真

Enterprise Architect支持模拟使用 Win32 ®用户接口配置文件创建的对话框和屏幕，以将用户界面设计与定义的系统行为集成。可以使用行为模型（例如状态机）中的JavaScript命令以编程方式引用和调用对话框，从而提供完全可定制和完全交互式的行为模型执行。

按钮控件可用于广播信号，在单击按钮时触发触发器。信号参数可以从对话框输入字段中填充；例如，捕获并发送用户名和密码以进行评估。

使用 Win32用户接口配置文件设计的对话框（并且与正在执行的行为模型存在于同一包分支中）将在模拟开始时在后台创建为新窗口。可以在设计时通过 Win32用户接口配置文件提供的标记值自定义影响每个对话框和控件的外观和行为的各种属性。

要在模拟过程中通过JavaScript与对话框交互，使用 对话框"模拟级关键字，后跟句点和对话框名称，然后可以访问属性和方法；例如，显示对话框，或设置 编辑控件"的文本值：

对话框.登录.显示=真；

dialog.Login.Username.Text="admin";

例子

要查看 Win32 用户接口仿真示例，请打开用户模型并找到图表：

示例模型>模型仿真>状态机模型>顾客登录>顾客顾客登录

公共属性

这些常用属性和方法在大多数支持的 Win32用户界面控件类型上都可用。

属性/方法	描述
使能够	布尔值 启用或禁用用户交互。
移动到（x，y，宽度，高度）	将窗口移动到指定坐标并设置窗口的高度和宽度。
节目	布尔值 显示或隐藏对话框。当此属性设置为False时，对话框将移出屏幕。
文本	字符串 设置对话框或窗口的标题。

JavaScript函数

函数	描述
广播信号（ string信号）	向模拟事件队列发送信号。 参数：

	• 信号：字符串——要广播的信号的名称
信号（ string信号，数组参数）	将带有附加参数的信号发送到模拟事件队列。 参数： • 信号：字符串——要广播的信号的名称 • 参数： Array – 为这个信号提供的附加参数 示例： UIBroadcastSignal("登录",{名称:文本, Password:文本});
ShowInterface (string InterfaceName, boolean Show)	已弃用。请参阅 对话框"控件上的显示属性。例如： 对话框.HelloWorld.Show = true;
InterfaceOperation（ string接口名称， string控制名称， string操作名称，[string arg1]，[string arg2]）	已弃用。操作可以直接从控件中引用。例如： dialog.HelloWorld.ListControl.InsertItem("测试", 2);
GetInterfaceValue (string InterfaceName, string ControlName, string OperationName,[string arg1],[string arg2])	已弃用.属性可以直接从控件中引用。例如： 文本;

注记

- 控件必须在对话框中；对话框外的任何控件都不会被解释
- 对话框和控件必须在 Win32用户接口图上
- 简单 UI 控件和基本 UI 控件也可以在模拟中使用，但与 Win32 UI 控件相比，功能有限
- 对话框名称和控件名称必须是唯一的；如果存在多个同名控件，则模拟将无法区分它们
- 对话框名称和控件名称中的空格被视为下划线
- 对话框名称和控件名称区分大小写

支持的 Win32 UI 控件

该表标识了Enterprise Architect中用于用户界面设计和模拟的所有 Win32 UI 控件。

访问

功能区	设计>接口>工具箱:图表> 在 查找工具箱项"对话框中指定 用户 - Win32"
键盘快捷键	Ctrl+Shift+3 : > 在 查找工具箱项"对话框中指定 用户接口- Win32"

Win32 用户界面控件

控件	描述
按钮	按钮控件是在运行时允许用户交互的常用方式；例如，登录屏幕中的确定按钮。 Button A响应单击事件，通过添加 "OnClick"标记值来定义。 响应点击事件，可以使用按钮，例如，发送信号，在运行时触发启动。 可定制的设计属性： ● 客户端边缘 ● 默认按钮 ● 禁用 ● 平坦的 ● 水平对齐 ● 模态框架 ● 多行 ● 右对齐文本 ● 从右到左阅读顺序 ● 静态边缘 ● 制表位 ● 透明的 ● 垂直对齐 ● 可见的 标记值： ● OnClick – 指定响应此按钮上的单击事件而执行的JavaScript命令 属性： ● 使能够 ● 节目 ● 文本 操作： ● 搬去

选择框	可定制的设计属性： • 自动 • 客户端边缘 • 禁用 • 平坦的 • 水平对齐 • 左文本 • 模态框架 • 多行 • 右对齐文本 • 从右到左阅读顺序 • 静态边缘 • 制表位 • 垂直对齐 • 可见的 标记值： • OnCheck – 指定响应此复选框值的变化而执行的JavaScript命令 属性： • 检查器 - 整数值 [0| 1] • 使能够 • 节目 • 文本
组合框	可定制的设计属性： • 自动 • 客户端边缘 • 数据 – 以分号分隔的string ，用于在运行时填充组合框；例如， 是；否；也许” • 禁用 • 有字符串 • 小写 • 模态框架 • 右对齐文本 • 从右到左阅读顺序 • 种类 • 静态边缘 • 制表位 • 类型 • 大写 • 垂直滚动 • 可见的 操作 • 添加字符串（ string ） • 删除所有（ ） • DeleteItem (number) – 删除指定索引处的项目

	• DeleteString (string) – 删除所有匹配string的项目 • 获取计数 () • 获取字符串（数字） • 插入项（数字， string ） • 插入字符串（数字， string ） • SetString（数字， string ） 属性： • 使能够 • 选择 - 当前选定项目的索引 • 节目
对话框	可定制的设计属性： • 绝对对齐 • 应用程序窗口 • 边框 - 仅调整大小或对话框框 • 中心 • 客户端边缘 • 中心鼠标 • 剪辑兄弟姐妹 • 禁用 • 水平滚动条 • 左滚动条 • 本地编辑 • 最大化框 • 最小化框 • 没有激活 • 重叠窗口 • 调色板窗口 • 右对齐文本 • 从右到左阅读顺序 • 设置前景 • 系统菜单 • 系统模态 • 标题栏 • 工具窗口 • 最顶层 • 透明的 • 垂直滚动条 • 可见的 • 窗边 属性： • 使能够 • 节目 • 文本 操作：

	• 搬去
编辑控件/富编辑控件	可定制的设计属性 • 对齐文本 • 自动水平滚动 • 自动垂直滚动 • 边界 • 客户端边缘 • 禁用 • 控件（仅限编辑） • 模态框架 • 多行 • 数字 • 密码 • 只读 • 右对齐文本 • 从右到左阅读顺序 • 静态边缘 • 制表位 • 透明的 • 控件（仅限编辑） • 可见的 • 想要返回 属性： • 使能够 • 节目 • 文本
组框	可定制的设计属性： • 客户端边缘 • 禁用 • 平坦的 • 水平对齐 • 模态框架 • 右对齐文本 • 静态边缘 • 制表位 • 可见的 属性： • 使能够 • 节目 • 文本
列表框	可定制的设计属性： • 边界

	• 客户端边缘 • 禁用无滚动 • 禁用 • 左滚动条 • 模态框架 • 右对齐文本 • 选择 • 种类 • 静态边缘 • 制表位 • 垂直滚动 • 可见的 操作： • 添加字符串（ string ） • 删除所有 （ ） • DeleteItem (number) – 删除指定索引处的项目 • DeleteString (string) – 删除所有匹配string的项目 • 获取计数 () • 获取字符串（数字） • 插入项（数字， string ） • 插入字符串（数字， string ） • SetString（数字， string ） 属性： • 使能够 • 选择 - 当前选定项目的索引 • 节目
列表控件	可定制的设计属性： • 结盟 • 始终显示选择 • 边界 • 客户端边缘 • 禁用 • 编辑标签 • 左滚动条 • 模态框架 • 无列标题 • 无滚动 • 单选 • 种类 • 静态边缘 • 制表位 • 视图 • 可见的 标记值：

	• Column - 用于初始化此 List控件的列名称和大小的string ，以分号分隔：例如，“Column1;100;Column2;150;” 操作： • 添加字符串（ string ） • 删除所有 （ ） • DeleteItem (number) – 删除指定索引处的项目 • DeleteString (string) – 删除与string匹配的所有项目 • 获取计数 () • GetString（数字，数字） • 插入项（数字， string ） • 插入字符串（数字， string ） • SetString（数字，数字， string ） 属性： • 使能够 • 选择 - 当前选定项目的索引 • 节目
图片控件	初始图片控件标记值可以使用控件图像来图像。将该值设置为模拟可访问的文件名。可以在运行时使用JavaScript中的 ChangeImageFile 方法修改图像。这需要一个要加载的文件名的string参数。 将 图像类型”属性设置为文件的正确类型（位图、增强型元文件或图标）。此设置不能在运行时修改。 可定制的设计属性： • 边界 • 中心图像 • 客户端边缘 • 颜色（框架颜色） • 禁用 • 图像类型 • 模态框架 • Real Size图像 • 静态边缘 • 制表位 • 视图 • 可见的 操作： • ChangeImageFile (string) - 文件名 属性： • 节目
进度控件	可定制的设计属性： • 边界 • 客户端边缘 • 禁用 • 选框 • 模态框架

	• 光滑的 • 静态边缘 • 制表位 • 垂直的 • 可见的 标记值： • Range – 指定此控件的最小值和最大值的string ，用分号分隔：例如，“ 1 ;100” 属性： • 使能够 • 位置 • 范围 • 节目 • 节
单选按钮	可定制的设计属性： • 自动 • 客户端边缘 • 禁用 • 平坦的 • 团体 • 水平对齐 • 左文本 • 模态框架 • 多行 • 静态边缘 • 制表位 • 垂直对齐 • 可见的 标记值： • OnChangeSelection – 指定响应单选按钮选择变化而执行的JavaScript命令 属性： • 检查器 - 整数值 [0| 1] • 使能够 • 选择 - 整数值 • 节目
滑块控件	可定制的设计属性： • 自动打勾 • 边界 • 客户端边缘 • 禁用 • 启用选择范围 • 模态框架 • 方向

	• 观点 • 静态边缘 • 制表位 • 刻度线 • 透明的 • 透明背景 • 工具提示 • 可见的 标记值： • Range – 指定此控件的最小值和最大值的string ，用分号分隔：例如，“ 1 ;100” 属性： • 使能够 • 页面大小 • 位置 • 范围 • 节目
旋转控件	可定制的设计属性： • 结盟 • 方向键 • 自动好友 • 客户端边缘 • 禁用 • 模态框架 • 没有数千 • 方向 • 设置好友整数 • 静态边缘 • 制表位 • 可见的 • 裹 标记值： • Range – 指定此控件的最小值和最大值的string ，用分号分隔：例如，“ 1 ;100” 属性： • 使能够 • 位置 • 范围 • 节目
静态文本/标签	可定制的设计属性： • 对齐文本 • 边界 • 客户端边缘 • 禁用

	● 结束省略号 ● 模态框架 ● 路径省略号 ● 无包装 ● 通知 ● 路径省略号 ● 右对齐文本 ● 简单的 ● 静态边缘 ● 凹陷 ● 制表位 ● 可见的 ● 单词省略号 属性： ● 使能够 ● 节目 ● 文本
选项卡控件	可定制的设计属性： ● 边界 ● 底部 ● 纽扣 ● 客户端边缘 ● 禁用 ● 扁平按钮 ● 重点 ● 热门曲目 ● 模型框架 ● 多行 ● 右对齐文本 ● 静态边缘 ● 风格 ● 制表位 ● 工具提示 ● 可见的 标记值： ● Tabs – 指定此控件的每个选项卡的名称的string ，以分号分隔：例如，“Tab 1 ;Tab 2;Tab 3;” 属性： ● 使能够 ● 节目
树控件	可定制的设计属性： ● 始终显示选择 ● 边界

	• 选择框 • 客户端边缘 • 禁用拖放 • 禁用 • 编辑标签 • 全行选择 • 有按钮 • 有线条 • 水平滚动 • 左滚动条 • 根线 • 模态框架 • 右对齐文本 • 从右到左阅读顺序 • 滚动 • 单 展开 • 静态边缘 • 制表位 • 工具提示 • 曲目选择 • 可见的 操作： • 删除() - 删除指定的TreeItem • InsertItem (string) - 要插入的新树项的虚线路径；将自动创建此虚线路径中尚不存在的任何父项 • InsertString (string) - 见*InsertItem* • TreeItem (string) - 要访问的树项的虚线路径；使用文本属性为该树项设置文本，或使用删除操作从树中删除该项 属性： • 使能够 • 选择 - 包含所选树项的虚线路径的string • 节目 • 文本– 获取或设置指定 TreeItem 的文本 例子： dialog.MyDialog.MyTreeControl.InsertItem("根.Parent.子"); dialog.MyDialog.MyTreeControl.TreeItem("根文本.子").Text="Modified"; dialog.MyDialog.MyTreeControl.Selection = "根.Parent"; dialog.MyDialog.MyTreeControl.TreeItem("根.Parent.Modified").删除();

Win32控件的标记值

可以在设计时通过 Win32用户接口配置文件提供的标记值自定义影响每个 Win32 对话框和控件的外观和行为的各种属性。

标记值

一些控件类型支持添加特殊标记值来修改它们的行为。

按钮、选择框和单选按钮等控件可以对 GUI 事件做出反应并执行JavaScript命令。要允许控件响应事件，请创建一个具有适当名称的新标记值；例如，“OnClick”，然后在值中键入JavaScript命令。

选项卡控件可以使用 选项卡″标记值来定义该控件在模拟时将出现的选项卡。

滑块控件、旋转控件和进度控件可以使用 范围″标记值来定义控件在模拟过程中接受的默认最小值和最大值。

标签	描述
列	**适用于**：列表控件 **使用**：初始化列表控件的列名和宽度。每列名称和宽度用分号隔开；例如，列 1；100；列 2；150；″。
点击	**适用于**：按钮 **使用**：标识为响应 Button 控件上的单击事件而执行的JavaScript命令。
安检	**适用于**：选择框 **使用**：标识为响应选择框控件值的变化而执行的JavaScript命令。
OnChangeSelection	**适用于**：单选按钮 **使用**：标识为响应单选按钮控件值的变化而执行的JavaScript命令。
范围	**适用于**： Slider控件、Spin控件、Progress控件 **使用**：指定控件的默认最小值和最大值，用分号分隔：例如，“ 1 ;100″。
标签	**适用于**：选项卡控件 **使用**指定要为选项卡创建的每个选项卡的名称，以控件分隔：例如，“Tab 1 ;Tab 2;Tab 3;″。

BPMN仿真

BPMN 模拟是一种可视化和验证 BPMN业务流程流程图行为的方法。通过所有当前正在执行的活动以及接下来可以执行的可能活动的可视化指示，您将能够轻松地识别和解决您已建模的流程的潜在问题。

模拟仿真模型类似于模拟标准UML行为模型，除了BPMN：

- 使用一些不同的元素类型（例如网关而不是决策）和
- 对放置在与属性和元素相关联的 标记值"字段中的脚本进行操作，而不是在 ""字段中（如果您愿意，而不是在 执行分析器编译"对话框中）;脚本是用JavaScript的

与BPMN仿真合作

活动	细节
创建BPMN仿真模型	当您创建适合模拟的 BPMN模型时，您会考虑如何表示起点、流程和要测试的条件。
将UML活动与 BPMN 流程进行比较	BPMN业务流程模型的模拟与UML活动图的模拟有许多不同之处。

注记

- BPMN 模拟在Enterprise Architect的统一版和终极版中可用

创建BPMN仿真模型

作为开发仿真模型过程的一部分，请考虑您更喜欢应用执行仿真的三个选项中的哪一个：

- 执行一个模拟脚本来初始化图表的变量——选择'BPMN'作为平台，执行模拟'作为脚本'并选择脚本；然后，您可以在开始模拟之前或在模拟过程中，在元素和连接器的标记值内定义条件和决策作为JavaScript声明
- 不要使用脚本，而是在第一个活动中初始化变量，然后再次修改元素和连接器的标记值中的条件和决策，然后以“解释”执行模拟；然后您可以在模拟期间重新初始化变量以及条件
- 以“手动”方式执行模拟，并在每一步手动管理流程和条件

创建适合模拟的 BPMN 图

节	行动
1	从 BPMN 2.0 技术创建业务流程或 BPEL 图。如果您创建 BPEL 图， Enterprise Architect会显示专门的对话框来简化兼容模型的创建。
2	我们建议您创建一个开始事件以清楚地显示您的模拟从哪里开始。事件类型有多种选择；该选择不会影响您的模型的模拟。如果没有定义任何开始事件，则模拟将从没有传入序列流的活动开始。
3	添加正在建模的进程中涉及的所有活动。任务类型有多种选择；该选择不会影响您的模型的模拟。通过指定 Sub-Process 的活动类型并选择 Embedded 或 CallActivity 可以进一步分解活动的行为。还支持标准循环。
4	在您的活动之间添加序列流。在“BPEL属性”对话框中，您可以输入在遵循序列流之前必须满足的条件 (True)。您还可以将 conditionType 设置为“Default”，以确保如果所有其他分支都未能满足指定的条件，则将采用此流程。 如果您不使用 BPEL 图，则使用 conditionExpression 和 conditionType标记值。
5	为将导致进程或活动执行路径结束的任何条件添加事件。事件类型有多种选择；其中只有终止类型会影响执行。在具有多个活动节点的模拟中，它会导致整个进程终止，而不仅仅是到达该节点的线程。

注记

- 要包括正在模拟的包之外的包中的活动，请绘制：
  - 从包含包包导入连接器
  被模拟到每个外部包，或
  - 包含图表的包中的依赖连接器
  模拟到外部包中的每个活动

初始化变量和条件

对于 BPMN 模拟模型，您可以在执行分析器脚本中初始化变量。您还可以在流程的第一个活动元素的标记值中初始化这些变量，这使您在模拟进行时可以更灵活地添加和更改变量。同样，您可以在序列连接器的标记值中定义适用于流程中各个决策点（网关）的条件和值。

如果你想在你的模拟过程中加入一个用户界面，使用 Win32，你再次使用标记值来识别对话框或提示来显示，在活动元素中，就在处理值或决定的点之前。

对于UML图的模拟，“sim” object和“this” object内的变量显示在“局部变量”窗口中。

访问

使用此处概述的方法之一显示属性窗口的“标签”选项卡。

功能区	探索>门户>窗口>属性>属性>标签
键盘快捷键	Ctrl+2 >属性窗口的“标签”选项卡

初始化变量

1. 在图表上，单击过程中的第一个活动元素。
2. 在属性窗口的“Tags”选项卡中，点击脚本“value”字段的下拉箭头，选择“脚本”。
3. 在脚本“值”字段中，输入适当的JavaScript代码，例如：

   sim.loan=真； sim.status="未定义';

定义条件

1. 在图表上，单击从网关元素发出的序列流连接器。
2. 在属性窗口的“标签”选项卡中，单击条件类型“值”字段的下拉箭头，然后选择“表达式”。
3. 在conditionExpression'Value'字段（<memo>*）点击[...]按钮，显示标记值注记窗口。类型在相应的JavaScript代码中，例如：
   sim.status=="保持"
4. 点击确定按钮。语句文本显示为连接器的标签。

整合Win32用户接口

5. 在图表上，单击代表做出决定的活动元素。
6. 在属性窗口的“标签”选项卡中，点击“任务类型值”字段的下拉箭头，选择“脚本”。
7. 在“脚本值”字段中，输入适当的JavaScript代码，例如：
   对话框.屏幕1.显示=真；
   （此语句显示对话框 Screen1。您可以通过将 'Show' 更改为False来临时隐藏对话框。）

UML活动和 BPMN 流程的比较

BPMN模型的执行和模拟与UML活动图的执行和模拟有许多不同之处。这里介绍了相似概念的映射，以及表达系统行为的两种方法之间的差异。

UML活动和 BPMN 流程的比较

UML活动	BPMN业务流程
起点由 Initial节点定义。无法指定活动开始的原因。	起点由开始事件定义。这意味着活动开始的特定原因，尽管它可能是未指定的。
一个活动的基本行为单元是行动元素。 UML提供了许多不同形式的行动，尽管模拟使用了其中的一小部分。	活动活动元素A许多不同的任务类型可用。这些通常描述不同的执行方法（例如手动），而不是发生的情况。
控件用于连接活动图表中A元素。 A区别特征是，除了显式分叉节点外，任何节点都只能遵循单个控件控件。要在 Flow控件上限制流量，请添加一个守卫条件。	A用于连接序列业务流程中的元素。这些与UML活动图的不同之处在于默认采用所有有效的序列流。要限制序列流上的流，请将条件标记值设置为 表达式″，并在条件表达式标记标记值中创建脚本。
决策节点A显式地对正在做出的决策进行模型。当潜在流重新组合为一个时，将使用使用相同语法 A Merge 节点。	当必须选择单个路径时，使用设置为 独占″ A网关节点。它还用于再次组合潜在流。可以A方向指定为 会聚″或 发散″以 确选择两种模式。
A分叉节点用于并发执行多个节点，而一个汇合节点，使用相同的语法，用于等待所有传入的流变为可用，并以单个流离开。	设置为 并行″ A网关节点用于显式模型多个节点的并发执行。它还用于等待所有传入流变为可用并以单个流离开。可以A方向指定为 会聚″或 发散″以确选择两种模式。
不允许同时执行来自一个节点的UML活动的一些输出。如果您需要这个，您可以在以后添加带有适当 Guard 的控件。	设置为 Inclusive A Gateway 节点用于显式模型，即所有条件为 true 的传出流同时执行的情况。
A需要通过引用外部活动来进一步分解行动时，使用调用行为动作。	活动元素在需要通过引用外部活动进一步分解行为时设置为CallActivity子流程。
活动行动调用行动行为	活动元素被设置为嵌入式子流程，当行为需要进一步分解而不参考外部活动时。