



**ENTERPRISE ARCHITECT**

用户指南系列

# BPSim业务模拟

Author: Sparx Systems

Date: 13/11/2024

Version: 17.0

创建于  **ENTERPRISE  
ARCHITECT**

# 目录

BPSim业务模拟	3
安装 BPSim	6
配置	7
配置- 配置页面	9
BPSim - 执行页面	19
BPSim - 节页	21
BPSim - 审查页面	26
使用参数值对话框	28
使用 BPSim 执行引擎	31
BPSim执行引擎-仿真语言	35
跟踪属性参数Values	38
跟踪属性参数Values - 示例	40
比较 BPSim 配置	48
BPSim 图表	51
BPSim 示例	54
订餐协作版本1	55
订餐协作版本2	59
服务台电话支持仿真	62
基于日历的服务台电话支持仿真	69
汽车维修进程	73
BPMN2.0事件示例	80
错误事件	81
升级事件	84
事件子流程	87
带有链接事件的斐波那契数生成器	92
信息事件	95
信号事件	99
边界事件	107
事件独立中间事件	110
画墙进程仿真(调用活动)	113
BPSim 成本参数	119
在活动上设置成本参数	120
在资源上设置成本参数	123
导出一个导出配置	126

## BPSim业务模拟

开放的 BPSim 规范提供了一组丰富的材料，涉及如何为活动或任务配置和分配资源、如何引发事件、决策制定和其他现实世界的能力。一旦根据 BPSim 规范进行配置，业务流程模型（在 BPMN 中构建）可以传递给合适的 BPSim 模拟引擎，并使用运行信息中附加的配置数据，按照 BPMN 模型中定义的流程运行。

BPSim 规范非常详细，为感兴趣的建模师和业务战略家提供了前所未有的灵活性，可以将操作信息分配给模型，然后根据从仿真引擎返回的信息评估解决方案的质量。本节详细介绍了配置模型以执行细节时可用的各种屏幕和选项。

Sparx Systems 提供支持 BPSim 的模拟器 - **BPSim 执行引擎**。这个插件

与 Enterprise Architect 中定义的运行和 BPMN 模型集成，提供运行和存储多个模拟结果的能力，并在每个配置的结果集之间进行方便的比较。

**BPSim 执行引擎**是访问和使用 BPSim 配置功能的先决条件。执行引擎集成了 Enterprise Architect 的统一和终极版本；在企业版中使用，可以单独购买和安装。

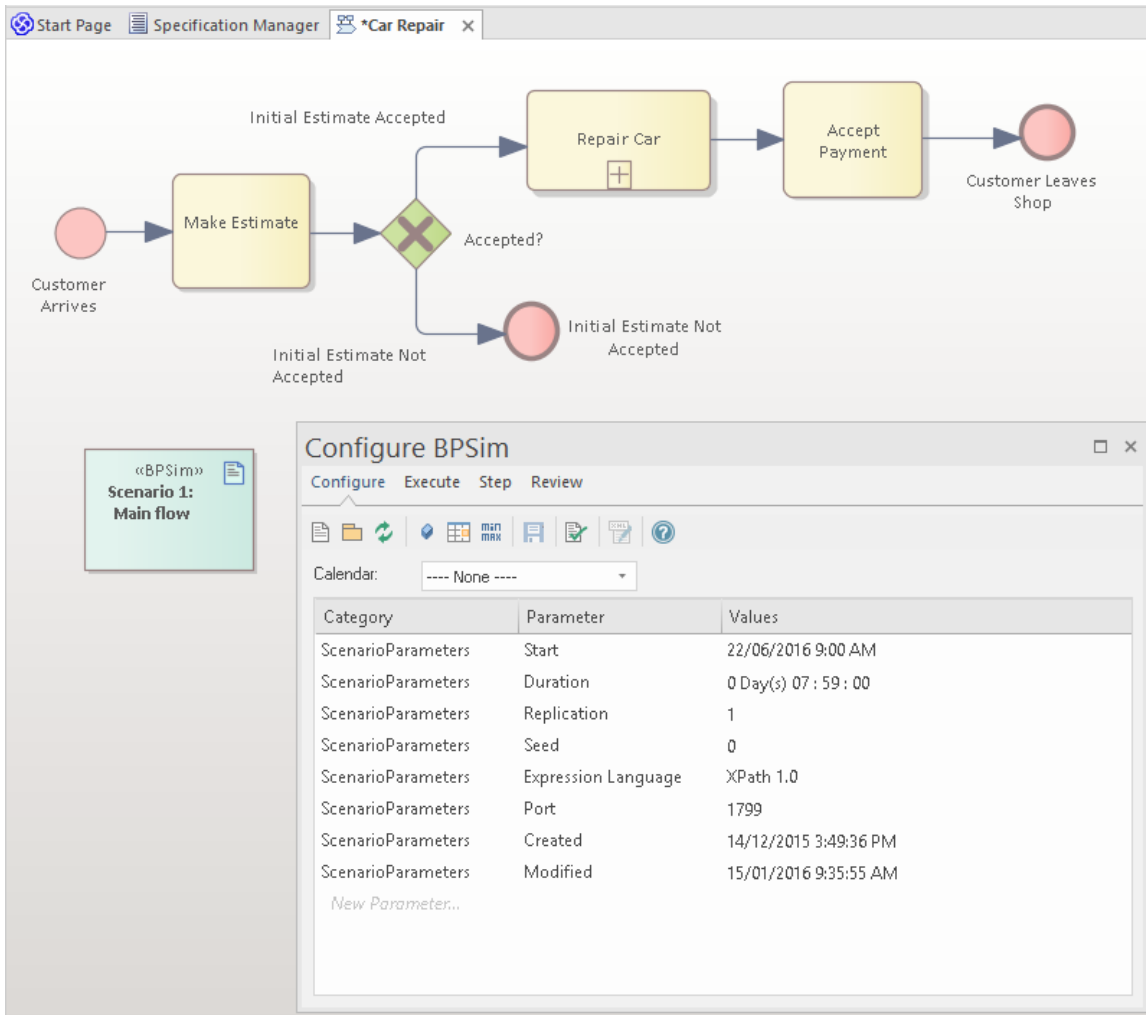
设置好 BPSim 配置后，模拟执行过程会以标准形式导出 BPMN 模型及其 BPSim 数据。这确保了对模型的更改始终合并到模拟中。类似地，模型导出过程可以导入另一个模型并由 Sparx Systems BPSim 执行引擎或任何其他符合标准的 BPSim 引擎使用的形式捕获 BPMN 模型及其 BPSim 数据。

### 安装 BPSim

虽然 BP 与 Enterprise Architect 的统一和终极模拟版集成，但它与企业版分开，并且 - 购买后 - 必须安装在您的系统上。

对于所有三个版本，您必须确保在您的系统上也安装了正确版本的 Java Runtime 环境 (JRE) 和 Java Development Kit (JDK)。

### BPMN 模型与 BPMN 仿真



配置 BPSim窗口可帮助您定义多个类别的仿真参数，每个类别侧重于仿真配置的一个方面。例如，您将定义：

- ScenarioParameters，定义仿真本身应该如何进行
- 控件参数，它检查活动如何在业务流程中流动，由控件事件的可能性和某些事件的优先级序列
- 时间（时间）参数，检查活动处理中一个或多个阶段的持续时间如何影响业务流程
- 资源参数，检查工人和其他资源的类型和角色的参与、所需数量、成本和可用性

您可以维护多个版本的配置（单独的工件配置也可以维护多个版本）并轻松比较每个配置的差异，以了解如何改变仿真或流程执行的流程例如，您可以建立一个基线配置，然后创建多个“假设”？改变一个或多个参数的配置。通过仿真引擎运行这些配置后，您可以检查每个结果并确定每个配置的优点。此处应用的一个有用原则是一个配置中常见的、未更改的数据由仅包含变化数据的另一种配置简单继承 - 因此，您可以在当前变量集上运行模拟，该模拟利用了标准数据配置同时。

用户可以结合图表和 Charting功能快速改变、模拟和比较业务流程模型的各个方面，并以多种图表格式之一显示模拟之间的差异。

如果您正在跨多个项目工作，您可以在它们之间导出和导入 BPSim 配置。该配置自动携带它所基于的 BPMN 2.0模型。

Enterprise Architect业务流程仿真配置工具基于工作流管理联盟 ( WfMC ) 开发的BPSim框架。

### 注记

- 如果单击图表或浏览器窗口中的业务流程元素或连接器，它将在配置 BPSim窗口中突出显示并选中
- 您模拟的业务流程可以包含多个包中的元素；要在模拟中包含外部元素，您必须创建一个包含“父”包和包含

- 外部元素的 外部”包或外部元素本身的包图；创建一个：
- 将包连接器从父包导入到每个外部包，或者
  - 从父包到每个外部元素的依赖连接器

# 安装 BPSim

虽然 BP 与Enterprise Architect的统一和终极模拟集成，但它必须安装在您的系统上。对于这些版本，您必须确保您的系统上也安装了正确版本的Java运行时环境(JRE) 和Java开发工具包 (JDK)。

## 安装 JDE 和 JDK

要使用Sparx Systems BPSim 执行引擎，您的系统上必须具有Java运行时环境(JRE) 版本1.7 或更高版本，并且如果您的配置包含任何属性参数，您还必须具有Java开发工具包 (JDK) 版本1.7 或更高。

除非您的系统上有多个版本的 JRE/JDK，并且您想指定执行引擎应该使用哪个版本，否则您不需要对引擎进行任何进一步的配置。在这种情况下，应用这些环境变量，如下所示：

1. 单击窗口 开始”图标并选择 电脑”选项。
2. 从横幅菜单中，选择 系统属性”选项。
3. 从侧面板中，选择 高级系统设置”选项。
4. 在 系统”属性的 高级”选项卡上，单击 环境变量”按钮。
5. 在 环境变量”对话框的 系统变量”面板中，单击 新建”按钮。
6. 在 系统”对话框中，完成具有显示值的字段：

对于 JRE：变量名称：MDG\_BPSIM\_JRE\_HOME  
变量值：C:\Program Files\Java\jre7

对于 JDK：变量名：MDG\_BPSIM\_JDK\_HOME  
变量值：C:\Program Files\Java\jdk1.7.0\_51

7. 点击确定按钮。
8. 您必须重新启动计算机才能使新变量生效。

# 配置

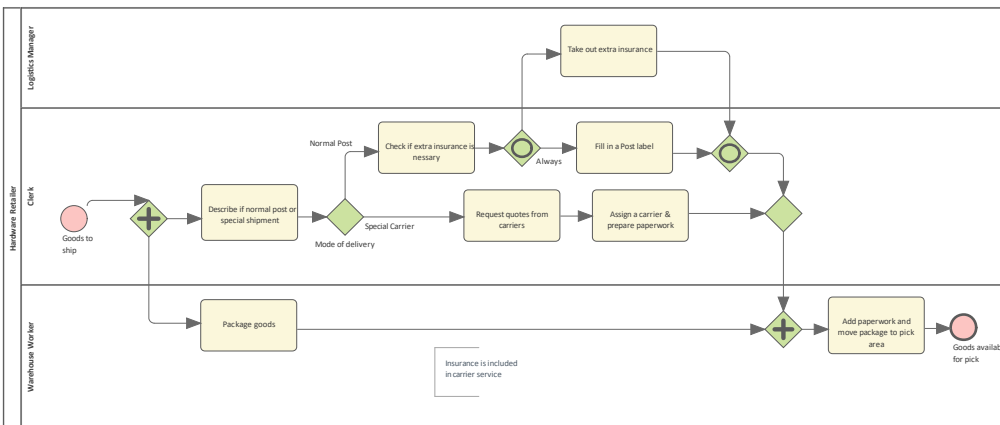
业务流程仿真模型中A元素流程工件（业务流程仿真）中的代表，您可以在与您正在使用的BPMN 相同的项目中的任何包中创建图表。

## 创建业务流程模型

每个 BPSim 配置都是专门为现有业务流程创建的，并根据 BPMN 中定义的现有业务流程创建。因此，在您使用业务流程仿真管理模型之前，您需要创建或导入该配置所基于的工件模型。

可以在 EAExample模型中找到并处理此示例图，位于：

分析与业务建模> BPMN流程2.0示例> 硬件零售商的进程图表>进程



## 打造业务流程仿真工件

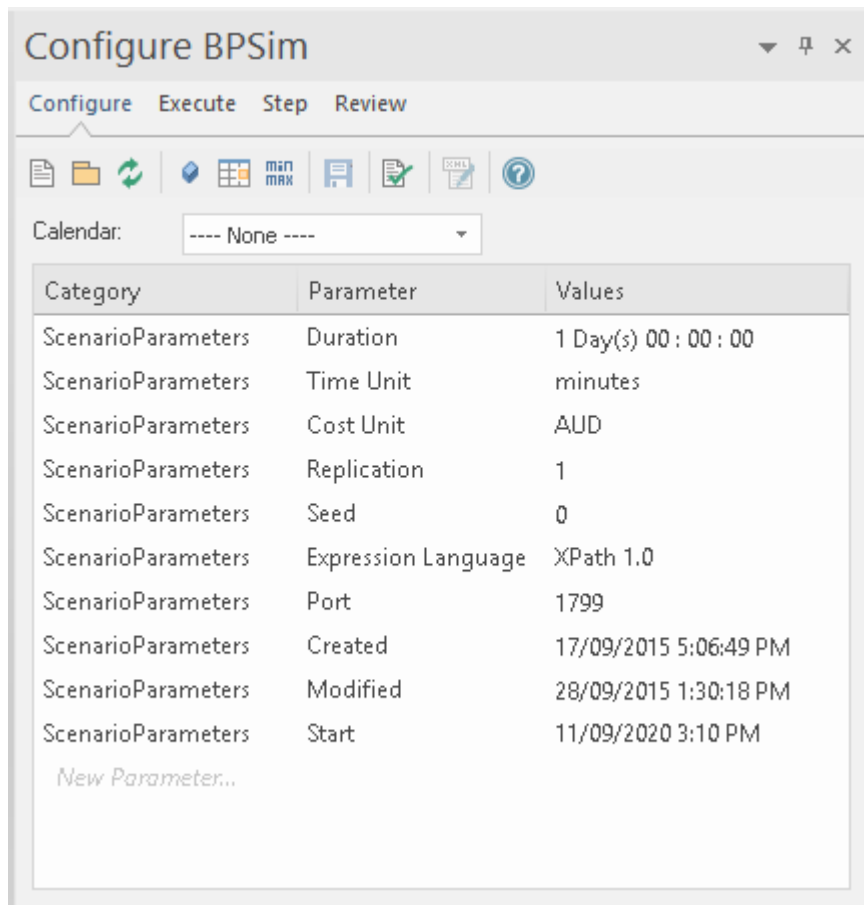
打开图表，在其中创建一个箱子，然后显示图表工具箱工件。展开常用的“仿真”页面，将“业务流程仿真”图标拖到图表上。

当您设置工件时，请考虑是否可以创建一个设备来定义其他工件以定义基本配置的某些方面或其他工件。您将在“变体”工件之间使用概括器和连接器，这样变体将继续继承您在“基础”工件中定义的数据。这样，您不必在每次工件时都重新定义整个配置。

双击元素并给它一个适当的名称，例如（例如）'Base配置'。

## 配置 BPSim窗口概览

右键单击设备元素（在图表或浏览器窗口中）并选择“配置 BPSim工件”选项。配置 BPSim的窗口工件。



此窗口包含四个选项卡：配置、执行、节和审阅。

- 配置：为每个BPMN元素配置配置参数；定义属性参数、日历和场景参数
- 执行：使用 BPSim 配置执行 BPMN模型
- 节：step over/step in 提供对执行过程的洞察，包括令牌状态、属性值和每个时间/步骤的资源分配
- 审阅/审阅工件 (s) ，生成标准的自定义模拟结果报告



## 配置- 配置页面

BP工件包将被配置为将加载此包或其子包下的所有 BPMN 元素。默认情况下，包含此工件的包会在加载到此窗口时进行配置。

这个窗口是上下文的。当在图表或浏览器窗口中选择一个元素时，列表将显示该元素的当前配置；此外，值的下拉列表将仅显示元素的可用参数。

当工件是上下文的元素时，列表会显示 ScenarioParameters。

### 访问

功能区	仿真>流程分析>进程> 打开BPSIM Manager >配置
-----	---------------------------------

### 工具栏选项

选项	描述
	单击此按钮可选择或创建一个 BPSimConfiguration元素。
	单击此按钮为工件按钮设置一个包。本包或其子包下的所有 BPMN 元素都将包括在内。
	单击此按钮可从已配置的包中重新加载 BPMN 元素。例如，当修改了一些 BPMN 元素时，运行此命令重新加载包，以便运行仿真将更改考虑在内。
	单击此按钮可定义属性，该属性可用作 BPMN 元素的属性参数。
	单击此按钮可定义日历，可用于配置元素参数。
	单击此按钮可显示或隐藏“结果请求”列。自定义模拟需要结果请求配置。执行报告将仅包含请求的结果。
	单击此按钮可将配置信息保存到配置 BPSim工件元素。
	单击此按钮以验证 BPMN模型和 BPSim 配置。如果生成错误或警告消息，它们可能会显示在系统输出窗口中。
	单击此按钮以导出具有 BPSim 配置的 BPMN模型。此导出的 BPMN 文件符合 BPMN 和 BPSim 规范，可供第三方 BPSim 执行引擎使用。

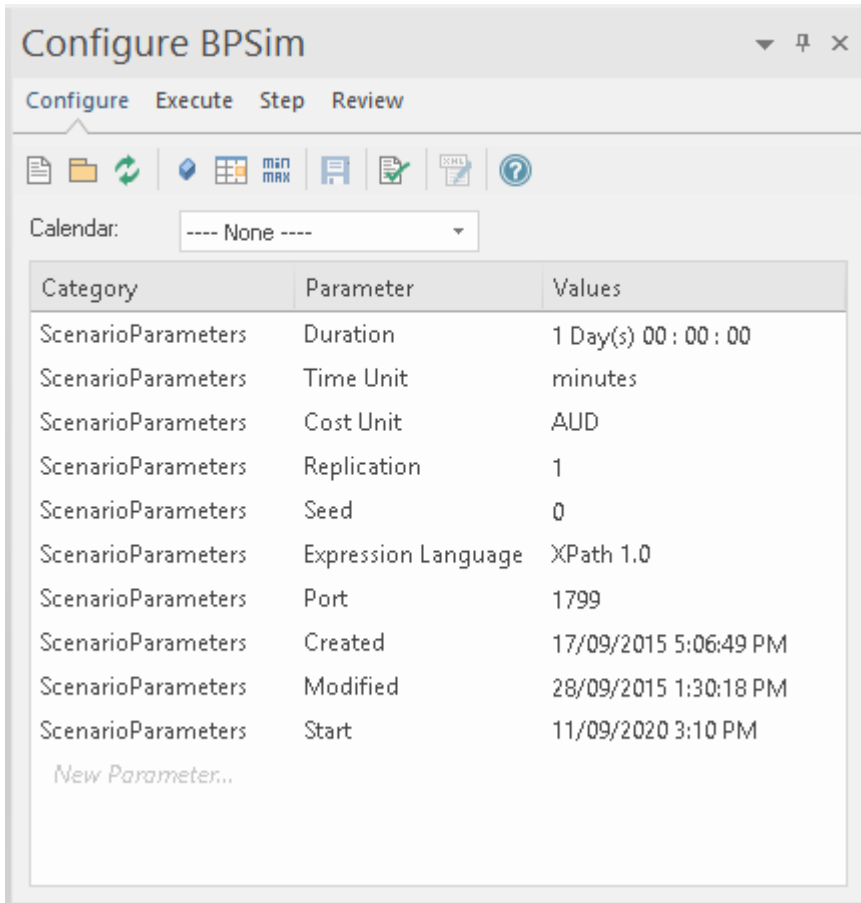
### 场景参数

A场景是由一组元素参数组成的。场景本身将所有元素使用的参数定义为全局设置。并非元素的所有参数都会

显示，但您可以通过以下方式将它们添加到列表中：

1. 点击新参数 文本，单击下拉箭头并选择 “场景参数”。
2. 单击 “参数” 字段中的下拉箭头，然后从列表中选择参数类型。

注意一旦您为场景添加了所有可能的参数，配置 BPSim窗口将不允许您尝试添加更多参数。



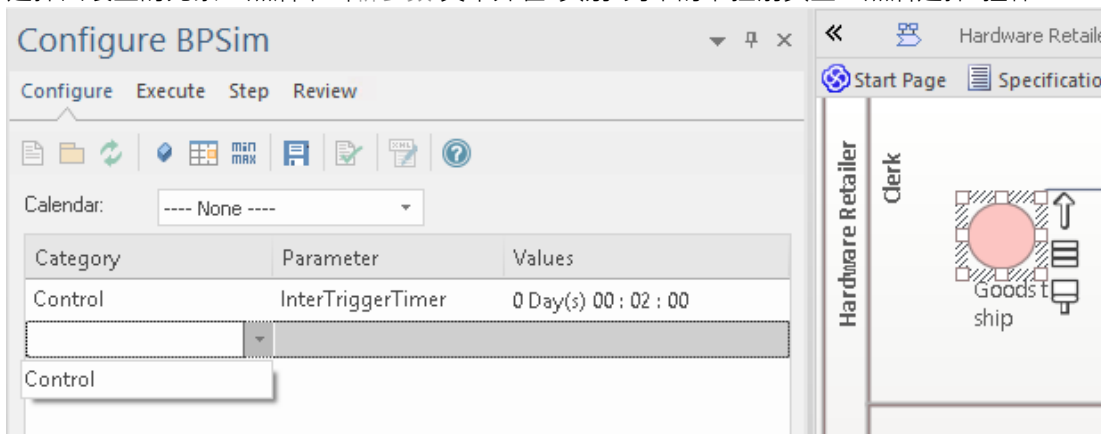
名称	描述
开始	<p>流程开始生效的日期和时间。</p> <p>您可以通过改写值来编辑它，或者对于日期，通过从下拉日历中进行选择来编辑它。</p>
期间	<p>该过程花费的时间长度。</p> <p>“持续时间”参数是必需值。它必须足够长以容纳完成模拟；例如，如果一个过程（以及它的模拟）需要三个小时才能完成，则“持续时间”参数必须设置为大于三个小时的值。</p> <p>您可以通过以“天:时:分:秒”格式改写适当的段来编辑它。</p>
时间单位	<p>在此方案中表示时间段的基本单位。除非在本地被覆盖，否则所有表示时间的数值和浮点值都应视为以该单位表示。</p> <p>您可以通过单击下拉箭头并选择单位来编辑它。</p>
成本单位	<p>流程中记录的任何成本的货币单位。除非在本地被覆盖，否则所有表示成本的数字和浮点值都应视为以该货币代码表示。</p> <p>您可以通过单击下拉箭头并选择单位缩写来编辑它。</p>
复制	<p>要执行的场景的复制次数。默认为1。</p>

	您可以通过简单地输入一个值来编辑它。
种子	用于初始化伪随机数生成器A随机种子。 您可以通过简单地输入一个值来编辑它。
表达语言	XPath 1.0 和Java - XPath 1.0 是默认语言。如果将Java指定为表达式语言，则必须设置 JDK Home。 您可以通过单击下拉箭头并选择语言来编辑它。
DMN模块	在 BPMN模型中使用业务规则任务时，您可以将这些任务作为 DMN模型来实现。 您可以先创建一个 DMN模型并在Java中生成一个 DMN服务器，然后单击  按钮指定生成的 DMN服务器文件。
JRE主页	Enterprise Architect BPSim 执行引擎在Java环境中运行，因此必须指定 JRE Home。单击  按钮选择一个目录；例如，C:\Program Files\Java\jre7。 您可以通过再次单击  按钮来浏览目录来编辑它。
JDK 主页	当表达式语言为Java时，Enterprise Architect BPSim 执行引擎将生成Java代码并使用 javac 作为供应商扩展进行编译。所以必须指定一个 JDK Home。使用  按钮选择一个目录（例如 C:\Program Files\Java\使用）。 您可以通过再次单击  按钮来浏览目录来编辑它。
端口	Enterprise Architect用来与端口执行引擎通信的端口号。默认端口号为 1799。
已创建	只读字段。工件创建时的时间戳。
修改的	只读字段。工件上次修改时间的时间戳。

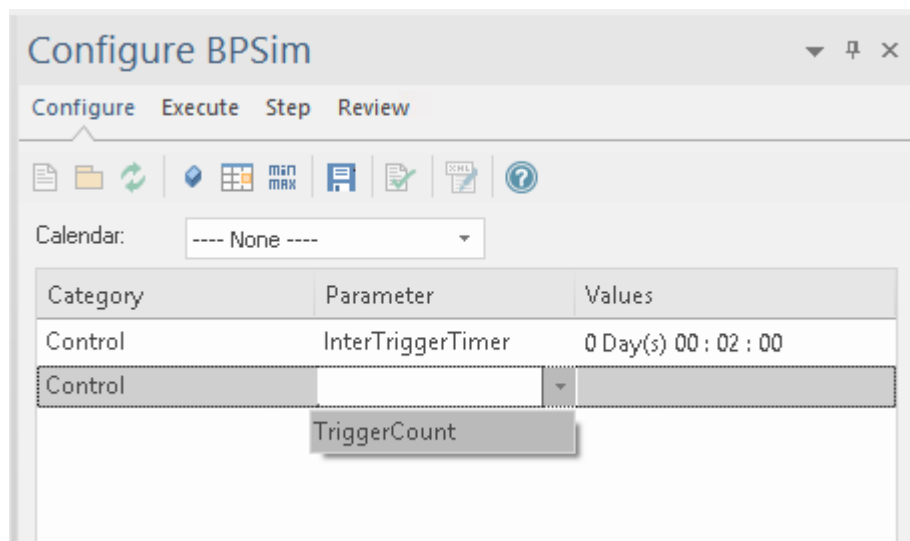
## 控件参数


要开始为适当的元素控件（例如事件或网关）定义参数：

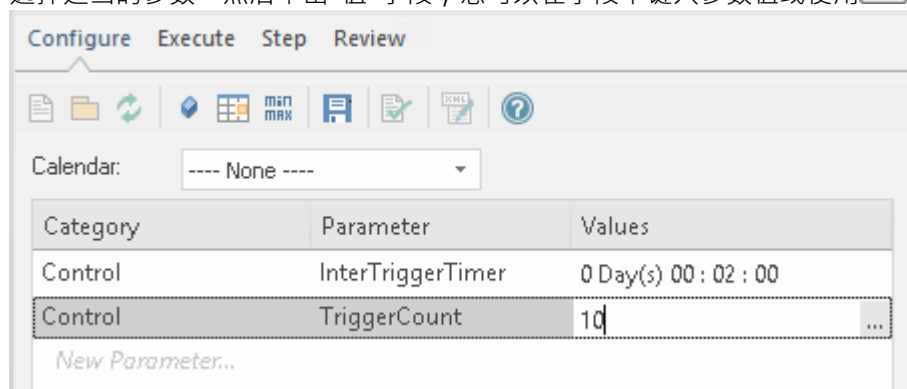
1. 选择图表上的元素，然后单击新参数 文本并在 类别”列中的下拉箭头上，然后选择 控件”。



2. 单击 参数”字段中的下拉箭头，它将显示所选元素的可用未分配参数。




3. 选择适当的参数，然后单击“值”字段；您可以在字段中键入参数值或使用  按钮打开“参数值”对话框。

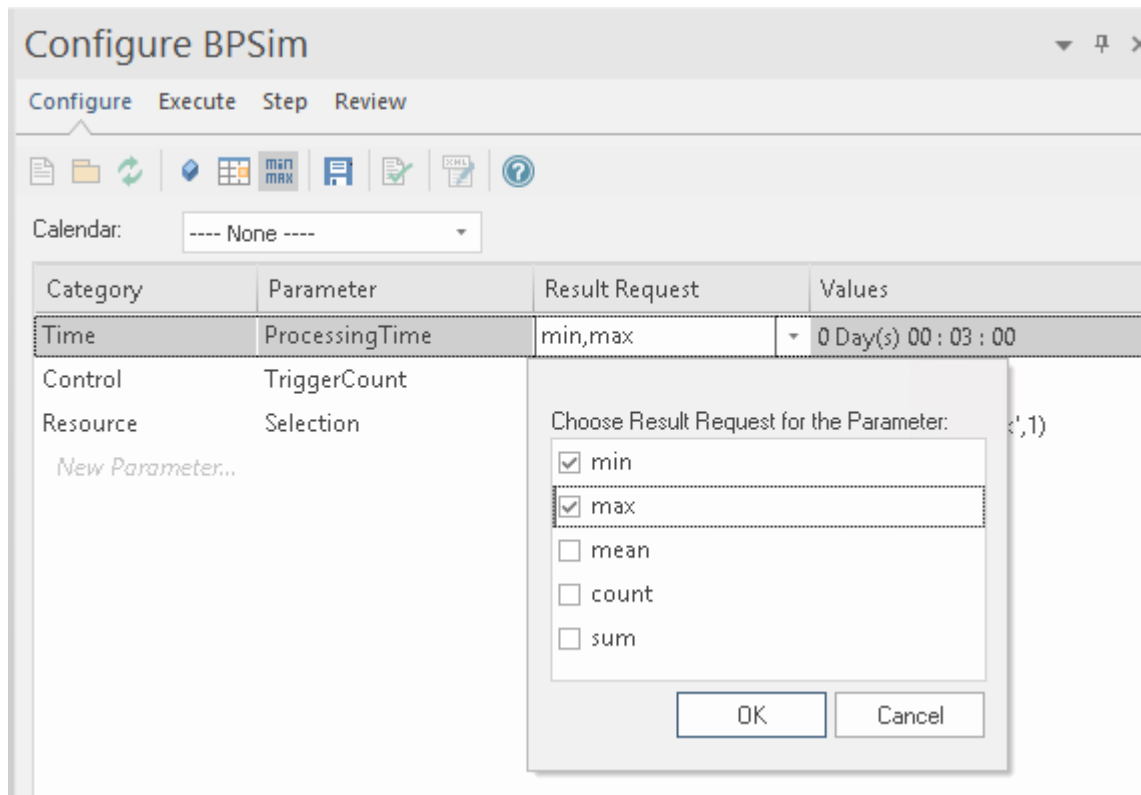


注意该页面只允许您为元素提供适当的参数。一旦您指定了这些参数，这些字段就不允许进一步输入或选择。

## 时间参数


要开始为适当的元素（例如 BPMN 任务）定义时间参数：

1. 选择图表上的元素。
2. 单击新参数文本和下拉箭头，然后从列表中选择“时间”。
3. 选择“时间”后，单击“参数”字段中的下拉箭头，然后从元素的可用参数中进行选择。
4. 在“值”字段中，输入值或单击  按钮打开“参数值”对话框。
5. 您可以通过单击工具栏上的最小/最大按钮来切换“结果请求”列，以通过要求某些结果来自定义模拟输出。



## 资源参数

要开始为适当的元素（例如 BPMN 任务）定义资源参数：

1. 选择图表上的元素。
2. 单击新参数文本和下拉箭头，然后从列表中选择“资源”。
3. 在“参数”字段中，单击下拉箭头并单击列表中的“选择”。
4. 在“值”字段中，单击  按钮以打开“编辑资源选择”对话框。

Resource or Role	Quantity Required
Clerk	1
Warehouse Worker	1

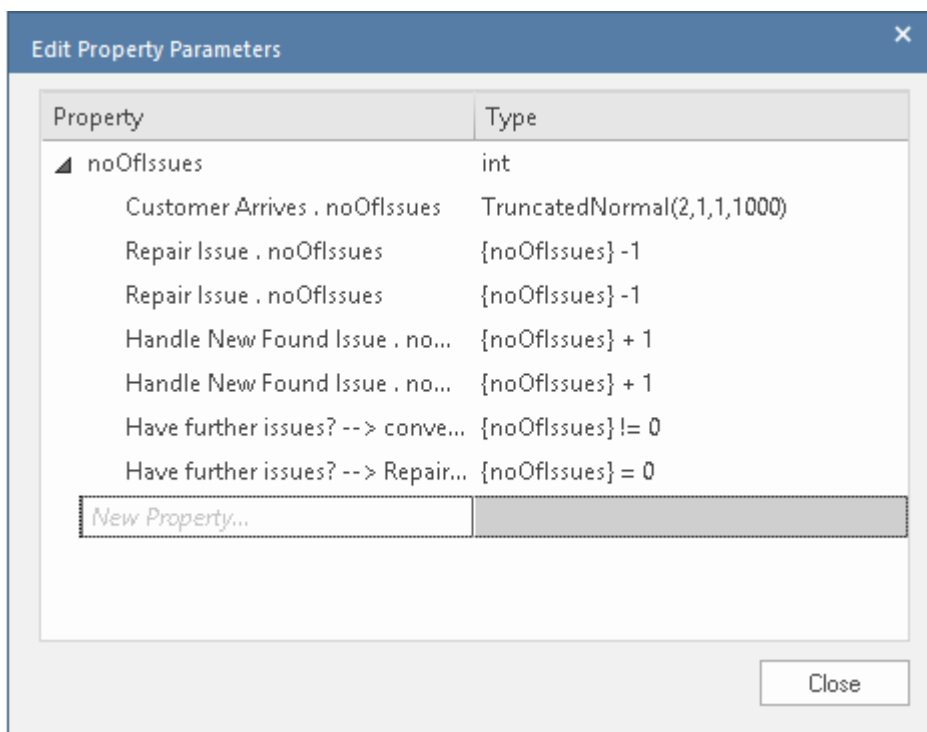
AND     OR

getResource('Clerk',1) and getResource('Warehouse Worker',1)

- 左上面板列出了已定义的 Resource 元素；单击要分配的资源，然后单击“按资源添加选择”按钮将选择移动到“资源或角色”面板
- 右上角的面板列出了为资源元素定义的角色（如果有的话）；单击所需的角色，然后单击“按角色添加选择”按钮将选择移动到“资源或角色”面板
- 每个资源/角色的“需要数量”列默认为1；如果需要更大的数量，请用适当的数字改写此值
- 单击相应的单选按钮以将选择的逻辑关系设置为 AND 或 OR
- 资源选择的最终表达式组成并显示在文本字段中
- 单击确定按钮返回到配置 BPSim 确定，其中表达式显示在“值”字段中

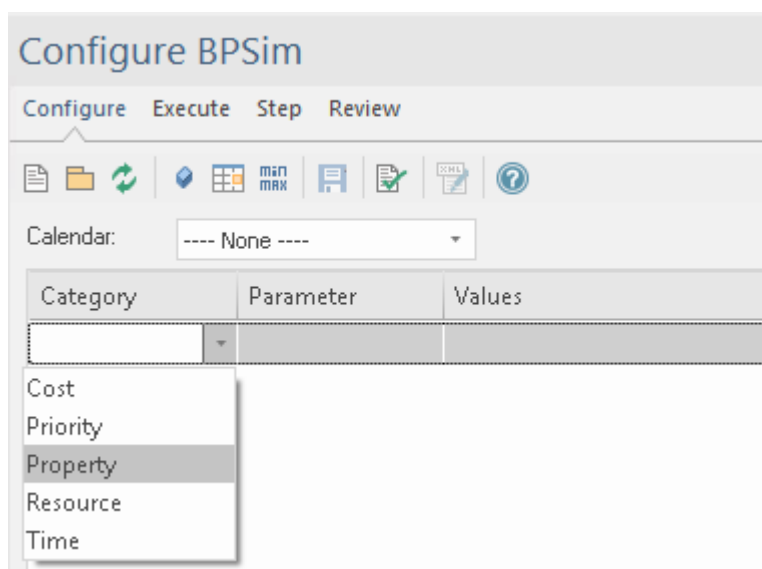
## 属性Parameters

要开始定义属性参数，请单击工具栏上的  按钮。将显示“编辑属性参数”对话框。




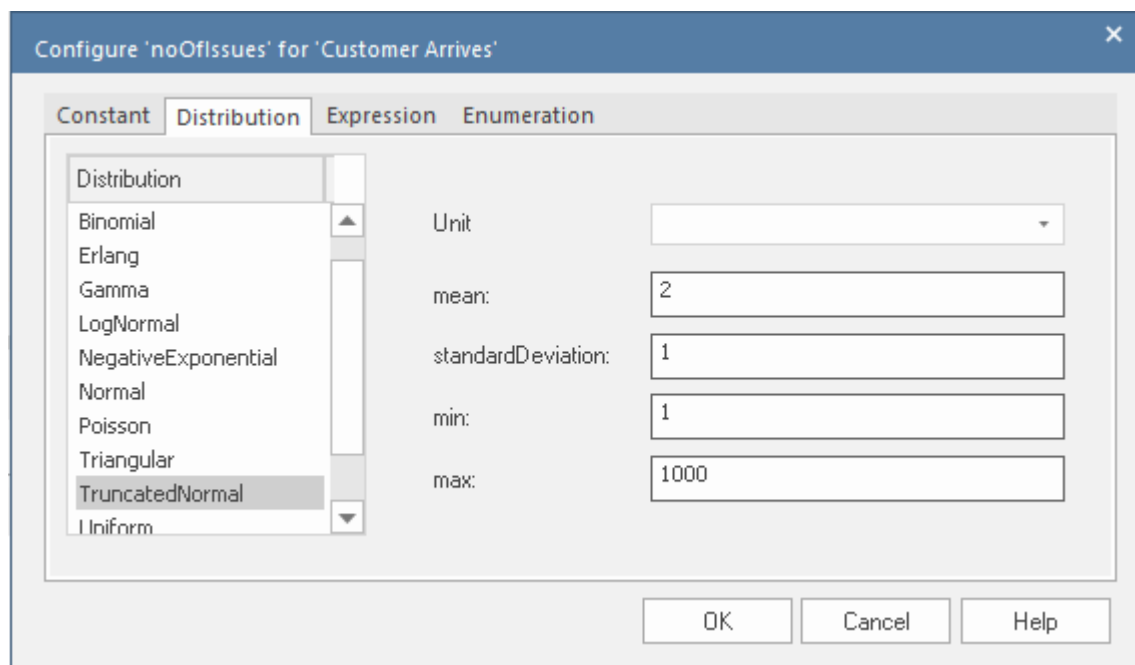
列出了定义的属性及其引用。

您可以添加新属性、删除选定属性（使用上下文菜单选项）、改写属性名称或为属性选择不同类型。检查定义的属性后，您可以在 BPMN 元素上设置属性参数。



选择“属性”作为类别，然后单击“参数”字段中的下拉箭头并选择一个属性。

单击“值”列上的  按钮以显示参数值对话框（从属性和父元素命名）。



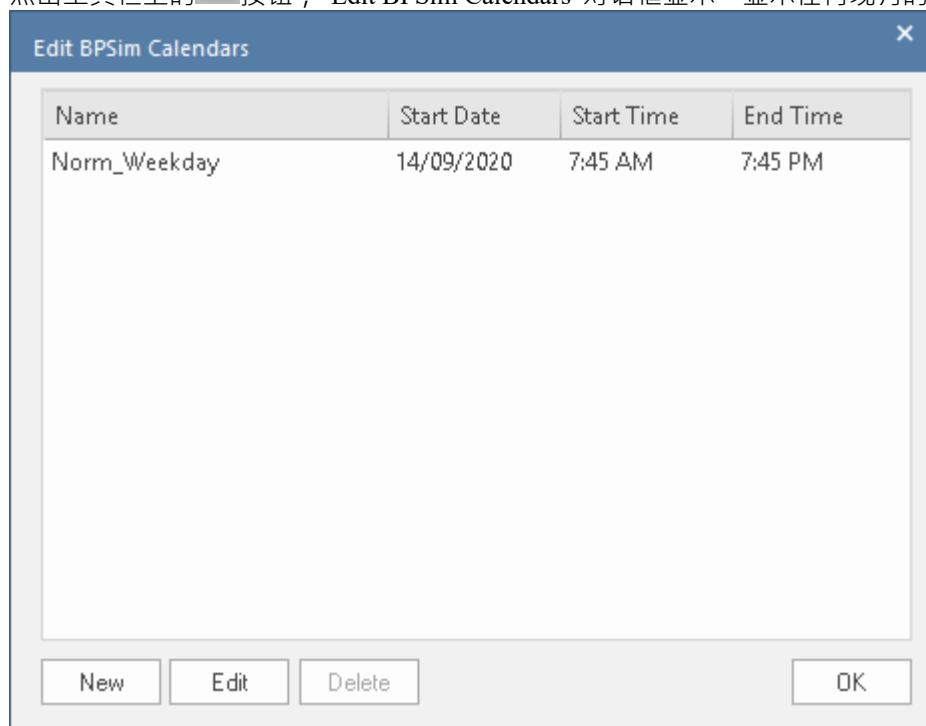
导航到相应的选项卡以选择并定义值和实际值的类型，然后单击确定按钮。该值显示在“值”字段中。

## 日历

日历可帮助您定义可能影响流程的任意数量的特殊时间段，例如工作日、轮班、假期或定期事件（例如盘点、盘点或审计）。

要开始定义日历：

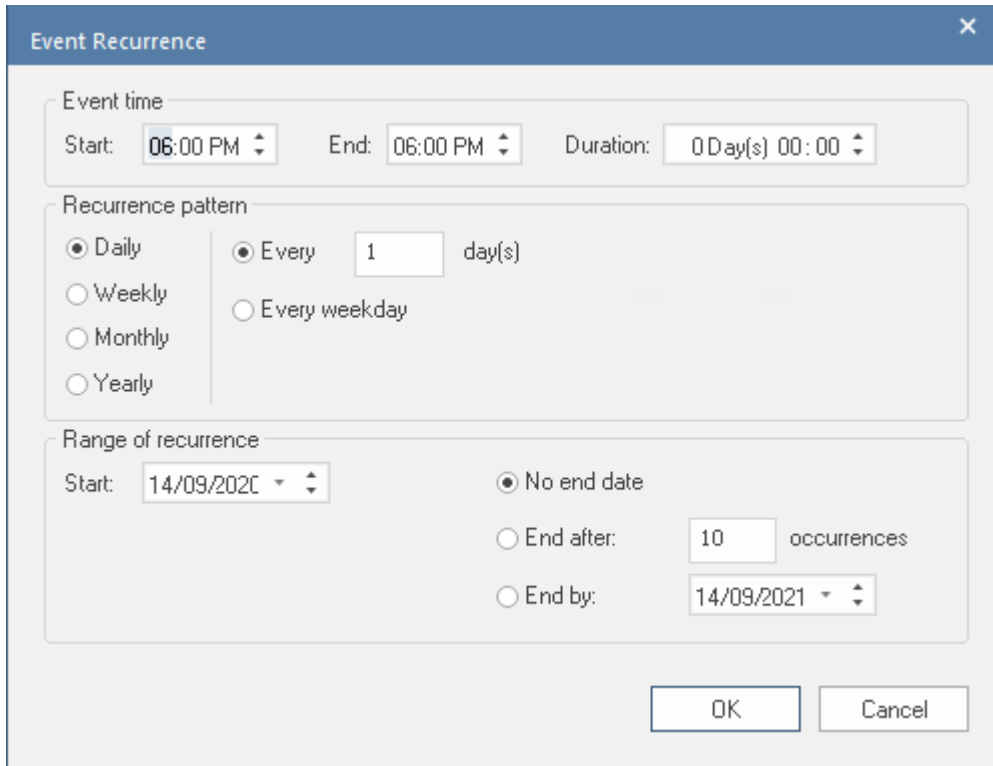
1. 单击工具栏上的  按钮；'Edit BPSim Calendars' 对话框显示，显示任何现有的日历。



您可以添加新日历，或者编辑或删除选定的日历。

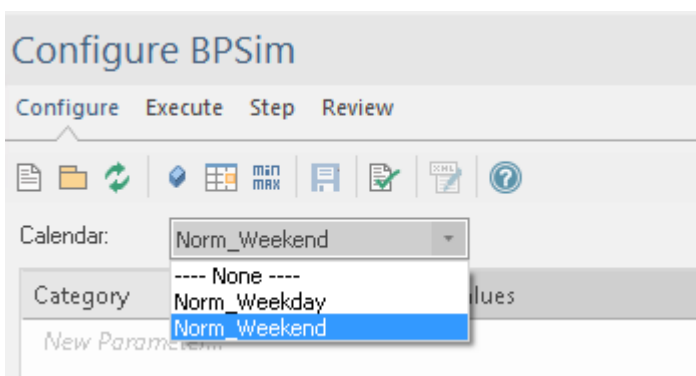
2. 要添加新的日历周期，请单击“新建”按钮以显示“事件重复”对话框。






3. 在“事件时间”面板中，开始和结束”字段均默认为当前时间。开始”字段是锚点；参考开始”字段，对结束”字段或持续时间”字段的更改会自动更新另一个字段。分别单击每个字段的时和分钟段（对于持续时间”字段，天”段），然后使用旋转”箭头设置开始时间和结束时间或持续时间时期。
4. 在“重复模式”面板中，选择日历周期重复间隔的单选按钮。每个选项在面板右侧显示一组适当的字段，用于将该时间间隔细化为每天/每周/每月或每两/三/四天/每周/每月、一周中的特定日期或日期或日期一年中的月份、日期或日期。根据需要选择下拉列表中的复选框或值。
5. 在“重复范围”面板中，选择日历周期生效的日期，然后选择适当的单选按钮来定义周期何时停止适用 - 从不、在设定的出现次数之后或在特定日期。您可以从下拉日历中选择结束日期，也可以使用日期每个部分上的旋转”箭头。
6. 点击确定按钮设置日历周期。

当您定义日历期间时，它们会按照开始日期和/或时间的顺序列出，最早在前。



使用已定义的日历，您可以在选定的日历上配置参数。

## 验证

为某些 BPMN 元素配置 BPSim 参数后，单击  按钮以运行模拟验证。任何 BPMN 或 BPSim 错误/警告都将显示在系统输出窗口中。根据消息修复问题。

完成此操作后 · 继续下一个帮助主题：帮助*Execute Page* °

## BPSim - 执行页面

定义配置后，您可以选择进行标准模拟或自定义模拟。执行将生成结果报告和用于重放（逐步执行）模拟的记录列表。

### 访问

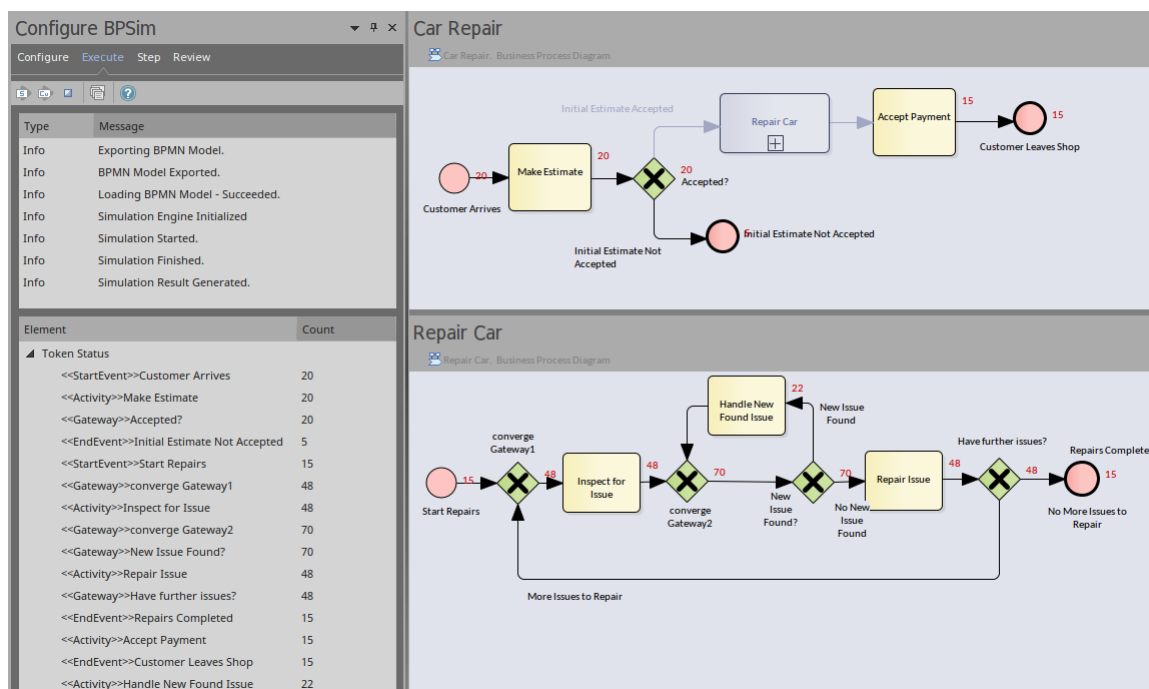
功能区	仿真>过程分析>进程>打开BPSIM Manager>执行页面
-----	---------------------------------

### 工具栏选项

选项	描述
	单击此按钮可使用 BPSim 配置执行 BPMN模型并生成标准报告。
	单击此按钮可使用模型配置执行 BPMN，并根据“配置”页面上的“结果请求”设置生成自定义报告。
	单击此按钮可停止执行并退出模拟。
	单击此按钮在“审阅”页面中打开生成的报告。

### 执行

当您点击运行仿真按钮或运行自定义仿真按钮时，带有运行配置的BPMN模型将被导出并加载到执行引擎中。



在模拟过程中：

- 令牌状态列表将闪烁运行时值
- 该图将与运行令牌计数一起闪烁

但是，模拟可能执行得太快而无法看到这一点。如果您使用“节”页面运行模拟，您可以看到这些变化。在此示例中，流程“汽车修理”和子流程“修理汽车”下的BPMN元素在新客户定期到达时被触发。

# BPSim - 节页

执行成功后，系统会生成一个执行报告，告诉你一般的流程状态，比如（以修车为例）一个任务的平均时间，客户的总等待时间，修复了多少问题。

此外，您可以从各个角度检查过程。例如：

- 从时间戳来看 - 这个过程在上午 9:30 的状态是什么？
- 从代币来看——第三位顾客在店里做了什么？
- 从属性来看属性号车的问题数量是如何减少和增加的？
- 从多个线程 - I 可以看到客户走进并在图表上自动模拟吗？
- 从资源中 - 支持人员何时忙或闲？为什么客户要等 40 分钟？

所有这些问题都可以在“节”页面上得到解答。

## 访问

功能区	仿真>流程分析>进程> 打开BPSIM Manager >节页
-----	---------------------------------

## 工具栏选项

选项	描述
	点击此按钮，根据执行结果自动模拟流程。 单击下拉箭头和菜单选项“设置回放速度”，然后将模拟速度调整为正常的倍数。例如，键入“60”使模拟比实际活动快 60 倍；现实生活中的1分钟将在1秒内模拟出来。
	单击此按钮可暂停自动重播模拟。
	单击此按钮可停止模拟。
	单击此按钮可将“节结束”到下一个时间戳。每个“节结束”可以包含多个“步骤”。
	单击此按钮可播放单步。这表示过程中令牌的单次移动。
	单击此按钮为仿真生成一个图表。 您可以从菜单中选择“为每个生成单个时间线”或“为每个令牌生成多个时间线”。请参阅本主题“面的生成图表”。
	单击此按钮可将此步骤页面上过滤的记录导出到 CSV 文件。您可以选择从哪个选项卡（“令牌”、“属性参数”或“资源”）导出数据。

### 令牌选项卡

运行执行后，这个页面会填充模拟过程中的token信息；序列的顺序是按照触发时间的顺序。

The screenshot shows the BPSim interface. The top window displays a BPMN diagram for 'Car Repair' with tasks: 'Make Estimate', 'Accepted?', 'Repair Car', 'Accept Payment', and 'Customer Leaves Shop'. The 'Configure BPSim' window is open, showing the 'Tokens' tab with the following table:

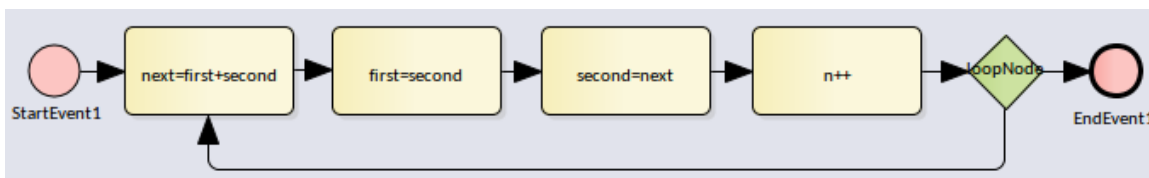
Token ID	Element	Action	Relative Time	Absolute Time
03				
03	Customer Arrives	Enter	072	22/06/2016 10:12:00 AM
03	Customer Arrives	Leave	072	22/06/2016 10:12:00 AM
03	Make Estimate	Enter	072	22/06/2016 10:12:00 AM
03	Make Estimate	Leave	074	22/06/2016 10:14:00 AM
03	Accepted?	Enter	074	22/06/2016 10:14:00 AM
03	Accepted?	Leave	074	22/06/2016 10:14:00 AM
03	Initial Estimate Not Accepted	Enter	074	22/06/2016 10:14:00 AM
03	Initial Estimate Not Accepted	Leave	074	22/06/2016 10:14:00 AM

- 使用标题带中的过滤器栏（右键单击列标题并选择“切换过滤器栏”），您可以过滤显示的结果；例如，在令牌 ID“列中键入 03 将仅显示令牌 03 的记录
- 如果您单击“节in”按钮一次，将播放列表中的一条记录
- 如果您双击记录，模拟将从头开始“节到”该记录
- 如果元素上设置了时间参数，点击节结束按钮将运行到下一个时间事件的最后一条记录
- 播放列表中的记录时，模拟快照将显示在图表上

### 属性参数选项卡

在播放“令牌”选项卡上的记录时，属性参数”选项卡将显示时间戳属性的运行时间值。

例如，计算斐波那契数的 BPMN 流程可能会以这种方式建模：



在定义好属性参数，为每个元素配置属性参数并执行模型之后，我们就可以进行步骤模拟了：

The screenshot shows the BPSim interface for a Fibonacci process. The flowchart at the top includes a StartEvent1, a task 'next=first+second', followed by 'first=second', 'second=next', and 'n++', leading to a decision diamond and finally EndEvent1. Below the flowchart is the 'Configure BPSim' section with tabs for 'Configure', 'Execute', 'Step', and 'Review'. The 'Tokens' tab is active, displaying a table with the following data:

Token ID	Attribute	Value	Message	Element	Relative Time	Absolute Time
0	N	10	Initialize	StartEvent1	0	0.0
0	first	1	Initialize	StartEvent1	0	0.0
0	n	0	Initialize	StartEvent1	0	0.0
0	second	1	Initialize	StartEvent1	0	0.0

信息'栏表示初始化了属性'N'、'first'、'n'和'second'。

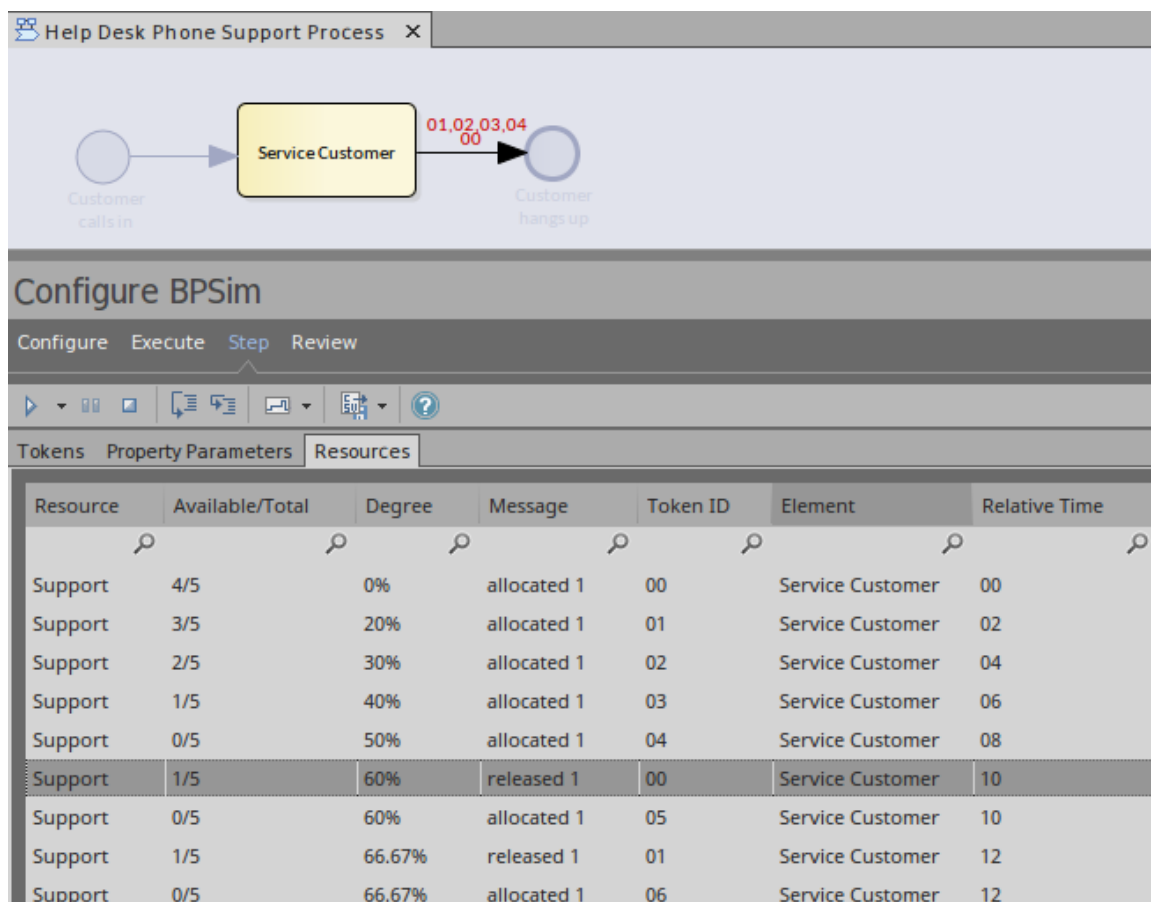
This screenshot shows the BPSim interface after the first task execution. The flowchart is identical to the previous one, but the 'next=first+second' task is highlighted in yellow. The 'Tokens' table below shows the updated values:

Token ID	Attribute	Value	Message	Element	Relative Time	Absolute Time
0	N	10	---	next=first+second	0	0.0
0	first	55	---	next=first+second	0	0.0
0	n	9	---	next=first+second	0	0.0
0	second	89	---	next=first+second	0	0.0
0	next	144.0	'89.0' --> '144.0'	next=first+second	0	0.0

如果你一直点击节在按钮，列表中的属性将改变它们的值。插图显示，在输入任务“next = first + second”时，属性“next”的值从 89 变为 144。

### 资源标签

在播放“令牌”选项卡上的记录时，资源”选项卡将显示可用的运行时资源、可用资源的数量以及时间戳的分配或释放事件。

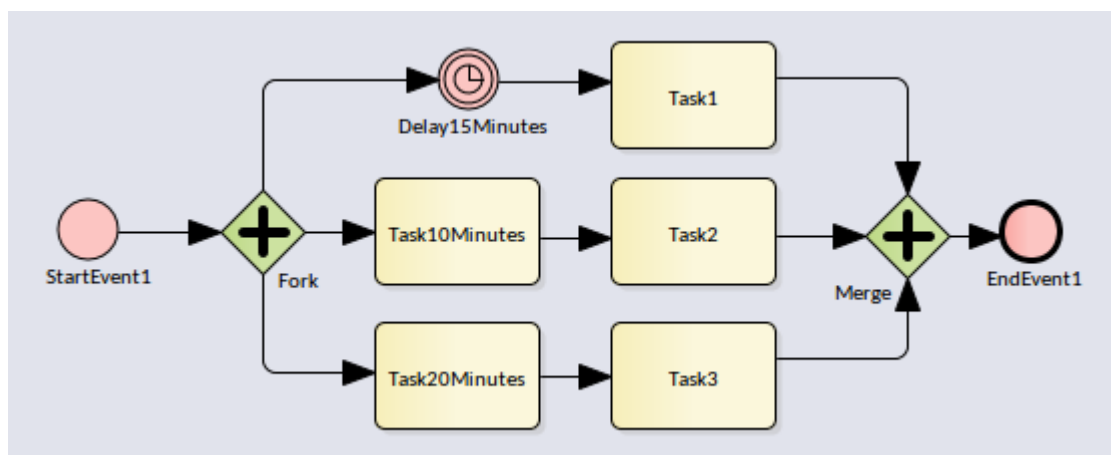


### 生成计时图表

当在 BPMN 元素上配置时间参数时，Enterprise Architect 可以为模拟过程生成时序图。

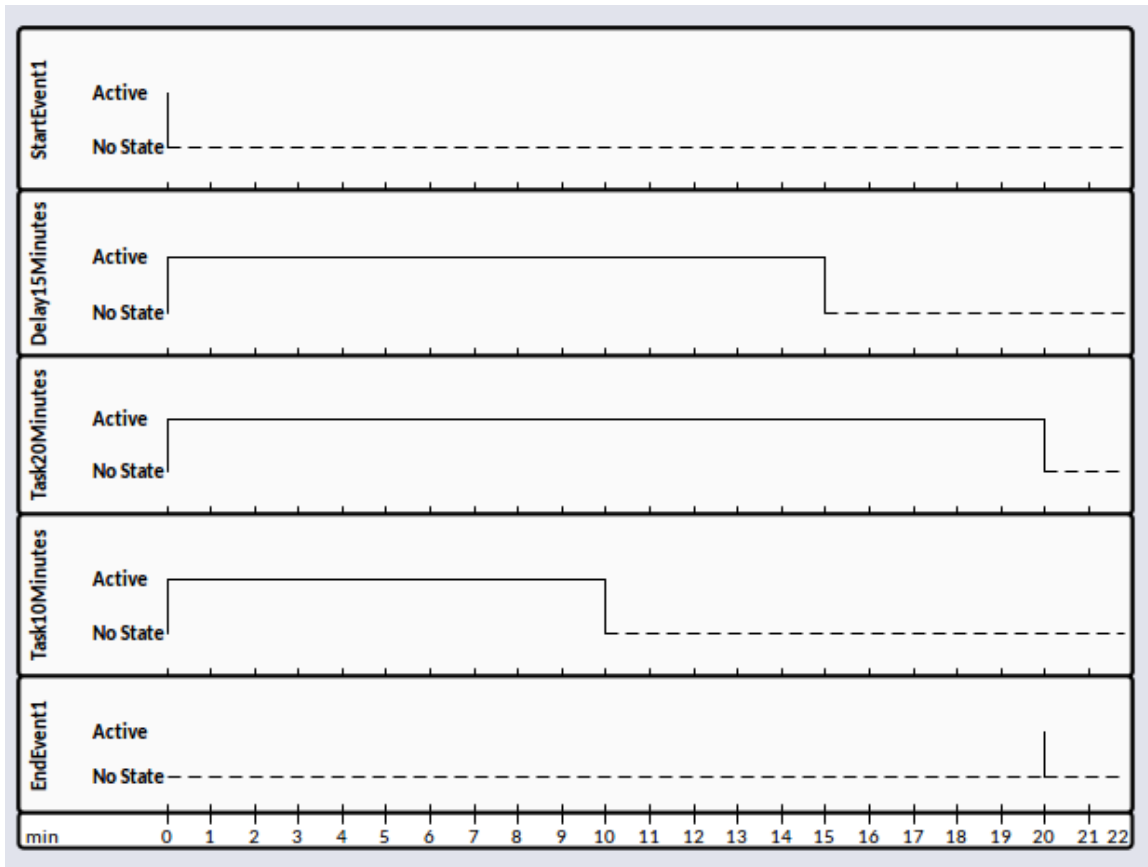
- 为每个令牌生成单个时间线 - 将此选项用于“单线程”流程；也就是说，没有并行网关或事件子进程
- 为每个令牌生成多个时间线 - 在“为每个令牌生成单个时间线”选项不适用的情况下使用此选项

例如：



执行此模型并单击“为每个令牌生成多个时间线”，生成的时序图如下所示：





## BPSim - 审查页面

此审阅页面包含三个选项卡：

- 配置总结
- 标准结果报告
- 自定义结果报告

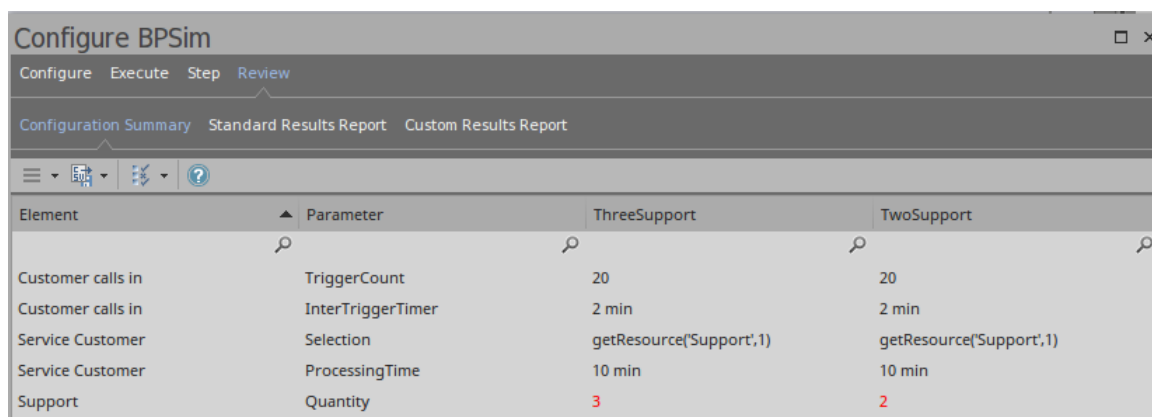
这些选项卡以类似的方式添加一个工件以审阅工件。这使您可以轻松进行假设分析。

### 访问

功能区	仿真>流程分析>进程>打开BPSIM Manager>审阅
-----	-------------------------------


### "如果"分析

在服务台支持示例中，我们可以比较两个工件及其相应的结果。



The screenshot shows the 'Configure BPSim' application window with the 'Review' tab selected. The 'Configuration Summary' sub-tab is active, displaying a table comparing parameters for 'ThreeSupport' and 'TwoSupport' configurations. The table has columns for 'Element', 'Parameter', 'ThreeSupport', and 'TwoSupport'. The 'Quantity' parameter for 'Support' is highlighted in red in both columns, with values 3 and 2 respectively.

Element	Parameter	ThreeSupport	TwoSupport
Customer calls in	TriggerCount	20	20
Customer calls in	InterTriggerTimer	2 min	2 min
Service Customer	Selection	getResource('Support',1)	getResource('Support',1)
Service Customer	ProcessingTime	10 min	10 min
Support	Quantity	3	2

在此图中，我们点击了工具栏中的  图标，并选择了“仅显示不同的项目”选项，以查看更改的参数值所导致的差异。

Configure BPSim			
Configure Execute Step Review			
Configuration Summary Standard Results Report Custom Results Report			
Element	Parameter	ThreeSupport- Result	TwoSupport- Result
Help Desk Phone Support Process	Maximum Time	34	64
Help Desk Phone Support Process	Standard Deviation Time	7.72	17.23
Help Desk Phone Support Process	Average Time	21.4	37
Service Customer	Average Number Of Tokens Waiting For Resource	3.17	5.29
Service Customer	Average Time In Task	21.4	37
Service Customer	Average Time Waiting For Resource	11.4	27
Service Customer	Maximum Number Of Tokens Waiting For Resource	8	12
Service Customer	Maximum Time In Task	34	64
Service Customer	Maximum Time Waiting For Resource	24	54
Service Customer	Total Time In Task	428	740
Service Customer	Total Time Waiting For Resource	228	540
Support	Degree Of Utilisation	92.59%	98.04%
Support	Number Started Immediately	3	2
Support	Total Time Available	216	204
Support	Total Time Idle	16	4
Support	Average Number Available	0.22	0.04


我们看到，当支持人员数量从 3 人减少到 2 人时，等待资源的平均时间从 11.4 分钟增加到 27 分钟。

# 使用参数值对话框

“参数值”对话框可帮助您为整个配置中的各种参数定义值。它支持定义简单的固定值，直至产生派生值的分布和表达式。并非所有类型的值或派生都适用于所有类型的参数。

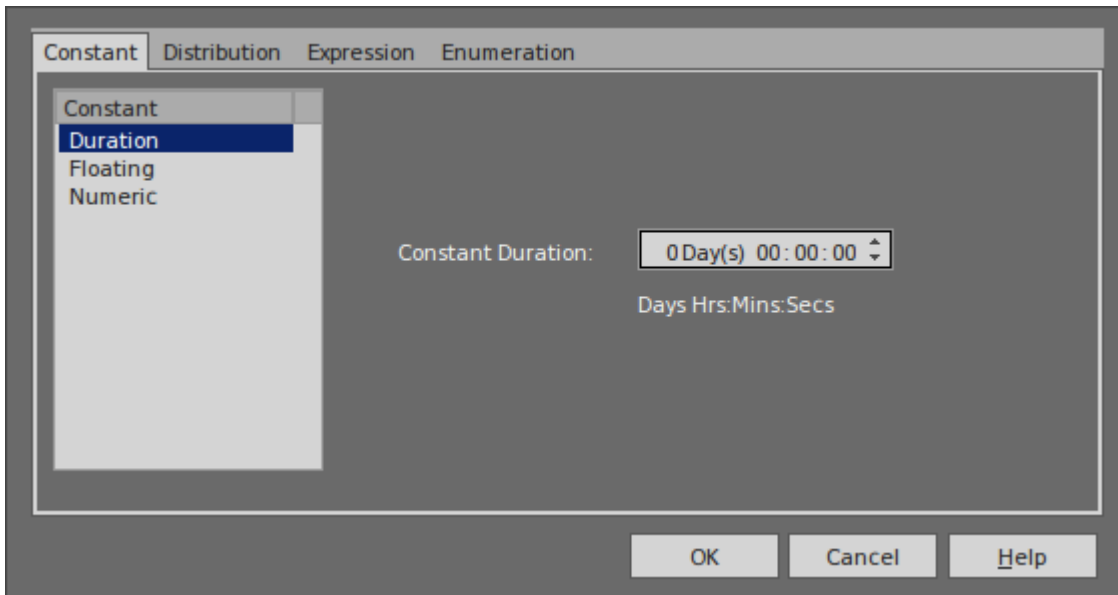
对话框名称取自object名称和正在定义的参数名称；例如，为 'Activity1'配置'Processing'。

## 访问

将工件加载到配置 BPSim窗口中，在图表或浏览器窗口中选择一个元素，然后在“值”字段中单击 。（如果尚未创建参数，请选择列表中的类别和参数以创建新参数。）

## 常量选项卡

使用此选项卡为参数定义特定值 - 例如，数字、文本string或时间。



在“常量”面板中，选择常量的类型：

- 漂浮的
- 数字
- 字符串
- 约会时间
- 布尔值，或
- 期间

适当的字段显示在面板的右侧；输入值，如果需要，输入表示值的单位（例如，时间单位或货币单位）。对于某些类型的参数，可以使用下拉列表从中选择一个值。

## 分布选项卡

在此选项卡上，您可以应用统计抽样方法来获取参数值；对于每种可用的分布类型，将显示相应的字段供您输入分布参数。所有发行版都要求您识别表达单位。

分布参数对于您正在开发的业务流程不是必需的，但（如果您从分布中获取值）是模拟所必需的。

您可以从以下分布类型中进行选择：

- **Beta** - 在短范围内提供“真实”值的连续概率分布，通常为 0 到 1
- **Weibull** - 提供“真实”值的连续概率分布，通常用于 object 寿命分析
- **Gamma** - 提供“真实”值的连续概率分布，可用于对指数分布的随机变量进行建模
- **二项式**- 整数“分布，根据试验次数和某个结果的概率提供值
- **Erlang** - 根据 Erlang  $K$  值和分布的平均值提供“真实”值
- **正态**- 根据分布的均值和标准差提供“真实”值
- **LogNormal** - 对数呈正态分布的“真实”随机变量的连续概率分布
- **泊松**- 离散（整数）概率分布，表示在固定时间或空间（体积、距离或面积）间隔内独立发生的给定数量事件的概率
- **NegativeExponential** - 根据分布的平均值提供“真实”值
- **三角形**- 根据分布模式和范围的最小值和最大值提供“真实”值
- **TruncatedNormal** - 根据范围的最小值和最大值内的点的平均值和标准差提供“真实”值
- **统一**- 提供范围内最小值和最大值之间的“真实”值

## 表达式选项卡

在此选项卡上，您键入 XPATH 1.0 表达式以组合要在运行时处理的显式值、运算符和函数以提供值。表达式的每个属性参数必须用大括号括起来 - {xxx}。

示例1：为了表示  $c = a + b + 10$ ，我们将此表达式分配给属性“c”：

{a} + {b} + 10

其中“a”和“b”是在属性模型中定义的属性。

示例2：为了表示  $c = t - p * (a - b)^2$ ，我们将这个表达式分配给属性“c”：

{t} - {p} \* 数学。pow({a} - {b}, 2.0)

注记：使用该表达式模拟模型时，请选择“Java”作为语言，以便使用 java 内置的函数数学。pow（）。

## 枚举选项卡

在“枚举”选项卡上，您可以定义一个枚举以提供一组常量值。您可以从现实世界的历史数据或模型的分析中获得这些值。每次评估参数时，都会返回下一个枚举值。

定义每个数值时，单击“保存”按钮将其添加到可能值列表中，然后单击“新建”按钮清除准备输入另一个值的数据字段。对于某些类型的枚举值，可能会要求您定义表示值的单位。您可以定义的枚举类型包括：

- 字符串
- 漂浮的
- 数字
- 期间
- 约会时间
- 布尔值




## 使用 BPSim 执行引擎

BPSim 执行引擎是一个插件

与Enterprise Architect的统一和终极版集成，执行您使用业务流程仿真仿真（功能）功能定义的模拟。Engine 是访问和使用功能的先决条件。



### 访问

使用此表中的任工件配置配置窗口打开，单击配置 BPSim并浏览业务流程仿真按钮。

功能区	仿真>过程分析>进程> Open BPSim 管理器 ( 或流程分析>仿真>进程配置流程> 查找工件配置流程 )
上下文菜单	右键单击业务流程仿真元素配置 BPSim工件
其它	用工件BPMN 右键单击业务流程仿真元素仿真

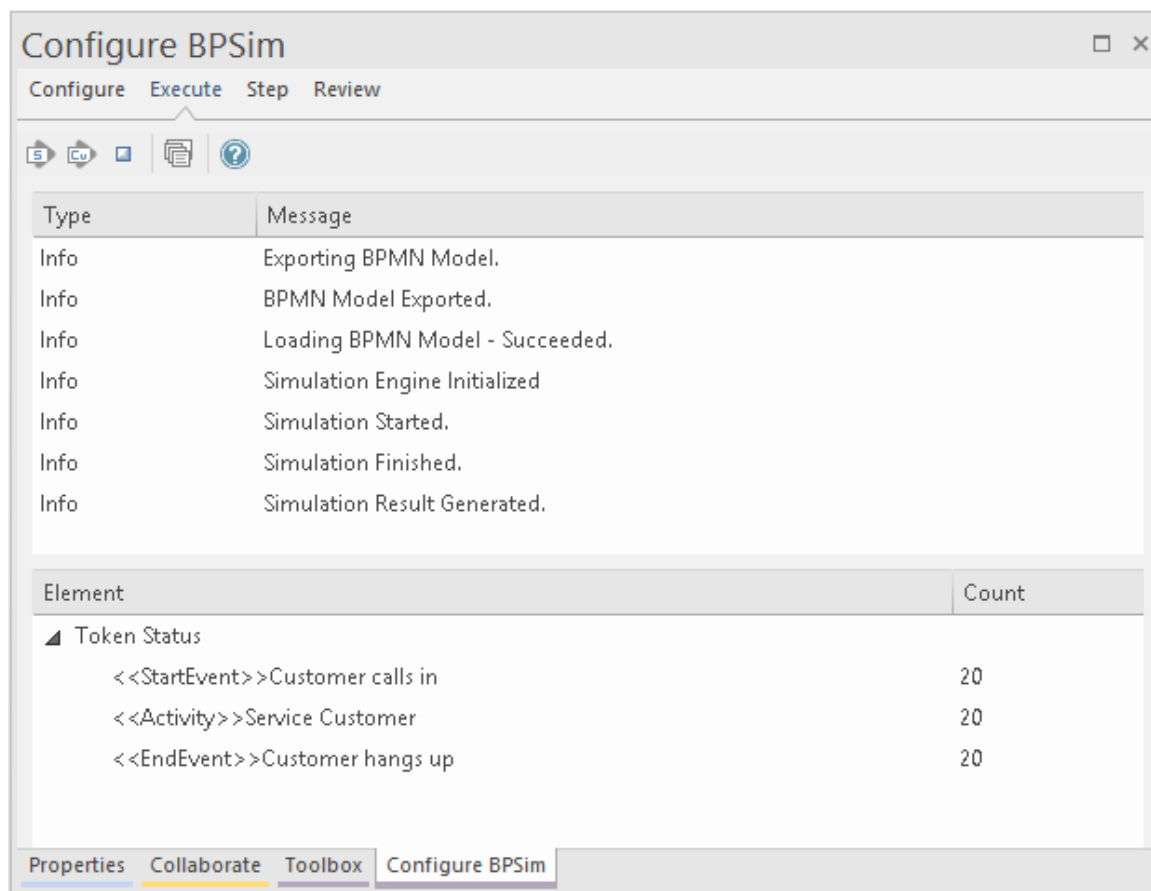
### 执行和仿真控件


单击 执行“选项卡，然后单击：

-  启动标准仿真或
-  启动定制仿真

这些选项触发相同的处理，除了标准仿真生成模拟中设置的所有内置参数的报告，而定制仿真仅提取您使用配置中的“结果请求”列特别标记的参数的结果。

模拟执行，在对话框顶部显示处理消息，并使用配置中使用的运行时值处理元素和参数。



在模拟过程中，您可以单击  图标取消模拟。

将结果写入业务流程元素的工件包。A <<自定义模拟>>-标准模拟写入<<BP 模拟报告>>-stereotyped工件 Report，而自定义模拟写入<<自定义报告>>-工件。

## 跟踪属性Values

除了内置参数外，您还可以在配置中定义自己的流程特定属性参数（属性）。模拟完成后，如果您已定义属性参数，则启用“属性”按钮。当您单击此按钮时，将显示“BPSim PropertyParameter Values”对话框，您可以通过该对话框跟踪所有属性参数的运行时值如何在业务流程中累积或更改。

## 审阅仿真

模拟完成处理后，单击“Open Result”按钮。“BPMN仿真报告视图”选项卡在主工作区打开，显示当前模拟中内置参数的结果（但不显示用户属性的参数）。如果您已经基于相同的业务流程运行了另一个配置的模拟，那么它也会作为附加列显示在报告中。否则，您可以单击工件元素的报表属性，并将其拖到报表配置选项卡上，以比较两个（或更多）下的内置参数。





为了方便查看报告中的数据，您可以将“BPMN仿真报告视图”选项卡拖出主视图，使其变为浮动窗口，并将窗口放大到合适的大小。

单击要检查的参数对应的扩展框。您还可以使用右键单击上下文菜单选项公开和过滤信息。

您可以将来自不同模拟的结果之间的特定差异表示为图表。模拟结果>> (<<工件的结果元素) 名称必须存在，然后才能设置图表工件。有一种用于标准图表的模板工件和一种用于标准模拟的模型。

## BPMN仿真报告选项

选项	描述

全部收缩	选择此选项可将参数层次结构折叠到仅父选项卡名称。																																	
展开全部	选择此选项可将参数层次结构扩展到最低值类型。																																	
仅显示不同的项目	( 当您显示两个或更多模拟时。 ) 选择此选项可将显示限制为模拟之间值不同的那些参数。再次单击该选项以取消选择它。																																	
突出显示不同项	<p>( 当您显示两个或更多模拟并且它们的某些参数值不同时。 ) 以红色显示不同的参数值。如果您选择 仅显示不同项”选项，此选项将被禁用。</p>  <table border="1" data-bbox="520 510 1417 813"> <thead> <tr> <th colspan="3">BPMN Simulation Report View</th> </tr> <tr> <th>Item</th> <th>Base BPSim Configuration - Result</th> <th>1 Extra Warehouse worker- Result</th> </tr> </thead> <tbody> <tr> <td>Time</td> <td></td> <td></td> </tr> <tr> <td>Logistics Manager</td> <td></td> <td></td> </tr> <tr> <td>    Total Time Available</td> <td>134</td> <td>91</td> </tr> <tr> <td>    Sum Of Wait Time</td> <td>104</td> <td>61</td> </tr> <tr> <td>Clerk</td> <td></td> <td></td> </tr> <tr> <td>    Total Time Available</td> <td>134</td> <td>91</td> </tr> <tr> <td>    Sum Of Wait Time</td> <td>50</td> <td>7</td> </tr> <tr> <td>&lt;&lt; Gateway &gt;&gt;</td> <td></td> <td></td> </tr> <tr> <td>    Total Token Wait Time</td> <td>181</td> <td>380</td> </tr> </tbody> </table>	BPMN Simulation Report View			Item	Base BPSim Configuration - Result	1 Extra Warehouse worker- Result	Time			Logistics Manager			Total Time Available	134	91	Sum Of Wait Time	104	61	Clerk			Total Time Available	134	91	Sum Of Wait Time	50	7	<< Gateway >>			Total Token Wait Time	181	380
BPMN Simulation Report View																																		
Item	Base BPSim Configuration - Result	1 Extra Warehouse worker- Result																																
Time																																		
Logistics Manager																																		
Total Time Available	134	91																																
Sum Of Wait Time	104	61																																
Clerk																																		
Total Time Available	134	91																																
Sum Of Wait Time	50	7																																
<< Gateway >>																																		
Total Token Wait Time	181	380																																
仅显示非空项	选择此选项可过滤显示以仅显示具有非 0 特定值的参数。																																	
移除模型	( 当您选择了特定的结果时，它会在报告中标识模拟。 ) 选择此选项可从报告中删除模拟列。																																	

## BPSim执行引擎-仿真语言

BPSim 执行引擎支持在 XPath 1.0 或Java上进行模拟，其中适当的语言被定义为模拟配置中的表达式语言。它还支持在属性Parameters 中使用流程实例数据，其中实际值仅在执行期间确定。

### XPath 1.0 运算符

这些运算符可用于 BPSim 表达式参数。

操作员	描述
	Union算子，用于资源获取。 示例： <code>getResource('w1', 1)   获取资源 ( 'w2' · 1 )</code>
+	添加。 示例： <code>4 + 6</code>
-	减法。 示例： <code>6 - 4</code>
*	乘法。 示例： <code>6 * 4</code>
div	分配。 示例： <code>8 div 4</code>
=	平等。 示例： <code>4 = 4 ( True )</code>
!=	不相等。 示例： <code>5 != 3</code>
<	少于。 示例： <code>6 &lt; 9</code>
<=	小于或等于。 示例： <code>x &lt;= 6</code>
>	比...更棒。 示例： <code>9 &gt; 6</code>
>=	大于或等于。 示例： <code>n &gt;= 7</code>
或者	选择。 示例： <code>n = 6 或 n &lt;= 6</code>
和	组合。

	示例：n = 5 和 m < 8
模组	模数除法。 示例：5 模 2
获取属性	获取一个属性值。 示例：getProperty ( "金额" )
获取资源	获取资源分配。 示例：getResource ( 'w1', 1 )

## 注记

表达式语言可以在配置 BPSim窗口中的“配置”选项卡上进行设置；'XPath 1.0' 和 'Java' 两个选项可用作 'Expression' 参数的值。

如果选择“Java”，则必须将属性“JDK Home”设置为有效的 JDK 目录。

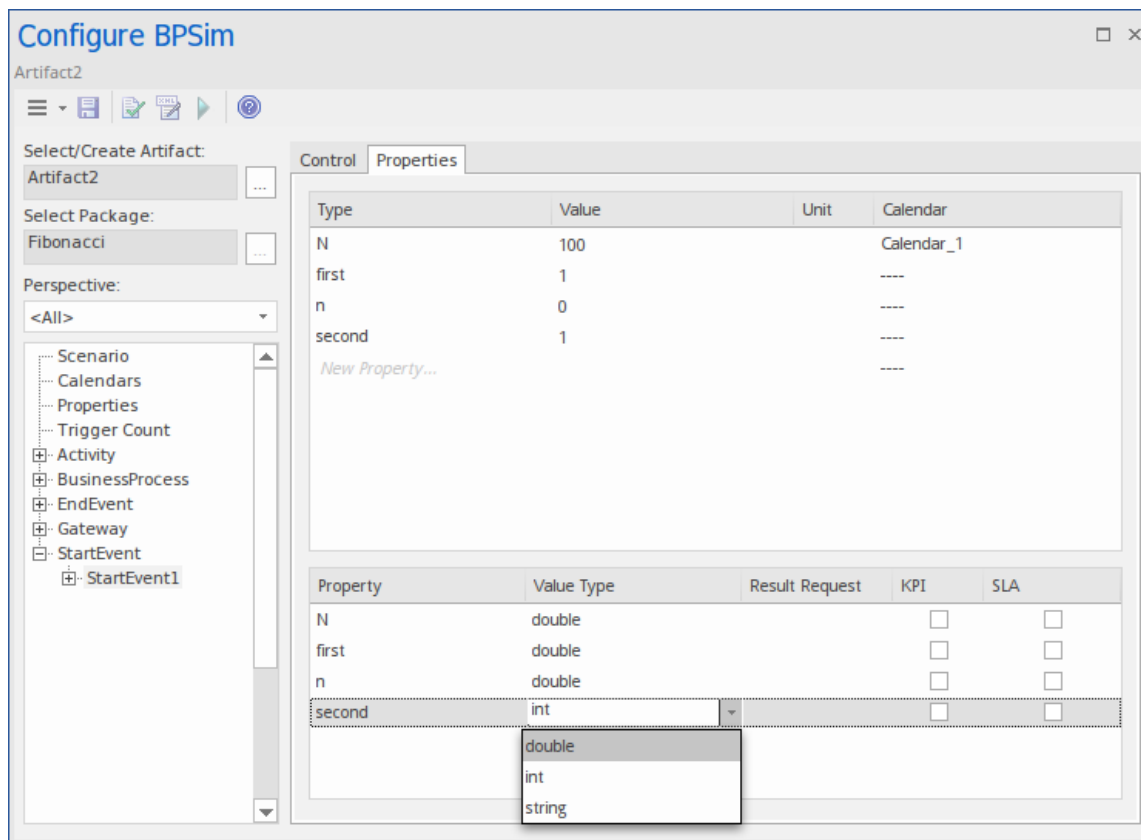
**提示：**您可以使用 {PropertyParameterName} 作为 getProperty('PropertyParameterName') 的简写形式，这在为表达式编写值时很有用；例如：

{n} < { N } 而不是 getProperty('n') < getProperty(' N ')

getProperty 运算符的缩写形式可用于 XPath 1.0 和 Java。

## 属性Parameters

从 Enterprise Architect 13.0 版开始，属性参数可以保存流程实例数据，直到运行时间才被赋值。您可以在配置 BPSim 属性的“属性”页面定义属性参数类型；支持的类型是“int”、“double”和“string”。



# 跟踪属性参数Values

业务流程模拟器 (模型) 可帮助您对业务流程的操作细节进行建模和测试，例如活动和任务的资源分配、事件的干预以及决策点的影响以及在在这些点做出的决策。您将这些特定于流程的属性参数或属性添加到属性配置中，当您根据配置在模型上运行模拟时，BPSim 引擎会帮助您捕获每次迭代的属性参数的运行时值模拟，并过滤结果以检查特定的路径或决策点。这使您可以非常详细地了解在特定条件或条件组合下您的业务流程中可能实际发生的情况，以生成结果或显示产生该结果的处理路径。

## 访问

上下文菜单	<p>右键单击定义的业务流程仿真- 仿真运行仿真配置 工件模拟类型... : 属性 (如果模拟配置不包含任何属性参数，则属性按钮不可用)</p> 
-------	--

## BPSim PropertyParameter Values 对话框字段

选项	行动
属性	<p>此表列出了为进程定义的属性，并显示了整个进程中每个属性的最小和最大可能值。</p> <p>如果单击某个属性的扩展框，则该表会显示该过程中每个活动或事件 (元素) 的属性的最小值和最大值。</p>
令牌编号	<p>类型要检查的“令牌”的数量；此数字必须在字段右侧显示的范围之内。A令牌“是一个独立的触发器，例如客户或订单进入业务并在审阅下启动业务流程。可以有任意数量的客户或订单，每个客户或订单都与任何其他客户或订单没有任何关系，并且每个客户或订单都可能在业务流程中遵循不同的路线。</p> <p>如果一个可能的触发事件只有一个可能的实例，例如抛出一个开/关开关，则该令牌被视为0。</p>
查询	<p>单击此按钮以启动对令牌模拟的查询，以填充“按元素分组”和“按属性分组”选项卡。</p>
按元素分组	<p>从属性参数的值如何在选定元素中变化的角度显示结果。该选项卡显示过程中的元素列表，并且对于每个元素，在模拟的每次迭代中应用在元素中的每</p>

	个属性的值。使用标题栏上的“切换过滤器栏”选项，您可以优化显示以仅显示特定属性并查看元素使用它的频率以及使用什么值。
按属性分组	从各个属性的值在整个过程中如何变化的角度来展示结果。该选项卡显示了在流程中应用的属性列表，并且对于每个属性，在流程的每次迭代中每个活动（元素）中的值。

## 例子

在 EAExample 模型中，您可以研究从 BPMN 业务流程模型的模拟生成属性参数信息的两个示例。这些将演示如何根据模型在配置中定义属性参数。您最初可以只对每个示例运行一个模拟，然后按照此处所述检查输出。然后，您可以检查业务流程和配置本身，并更改或添加提供的属性参数。

这些示例在跟踪属性 *ParameterValues* - 示例主题中进行了描述。简而言之，它们是：

- “斐波那契”——一个非常简单的递归业务流程，通过十次迭代计算一系列斐波那契数；您可以通过过程的元素看到属性参数在每次迭代中如何递增（在示例模型>模型仿真>示例模型>斐波那契）
- “汽车维修” - 一个更复杂和现实的过程，表示当一系列单独的“步入式”客户将车辆带入汽车维修店进行估计和维修时可能发生的情况（在示例模型>模型仿真>示例模型>汽车修复进程）

还有一个时间参数行为的小例子（在示例模型>仿真模型>示例模型>时间参数）。

## 注记

- 如果 BPSim 配置包含结果请求并对其执行自定义模拟，则“BPMN 仿真报告”仅显示配置中请求的内置参数；相反，“BPSim PropertyParameter Values”对话框列出所有属性参数，而不管任何结果请求设置或模拟类型如何

# 跟踪属性参数Values - 示例

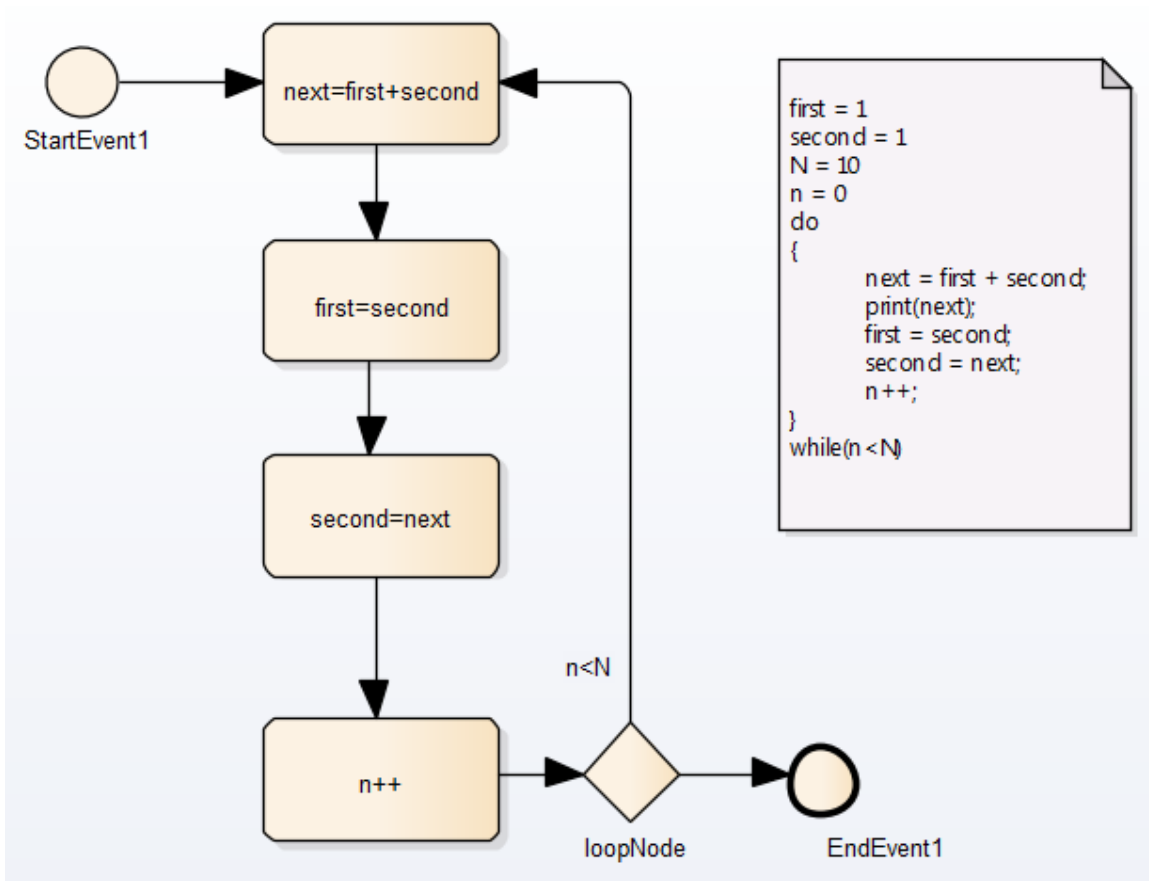
为了帮助您了解从 BPMN 业务流程模型的模拟中生成属性参数信息的功能，Sparx Systems提供了两个示例，您可以在 EAExample模型中探索。这些是：

- 斐波那契过程 - 一个帮助您熟悉参数跟踪功能的非常简单的示例
- 汽车维修过程 - 一个更复杂的示例，您可以操作它来了解如何调查现实生活中的过程

在本主题的结尾部分简要讨论了如何使用包含由“真实”分布初始化的参数的基于整数的过程，以及描述时间参数行为示例的部分。

## 斐波那契示例

这是一个非常简单的递归业务流程，通过十次迭代计算出一系列斐波那契数；您可以通过流程的元素看到属性参数如何在每次迭代中递增。打开示例模型>模型仿真>示例模型 > 斐波那契。



该过程的伪代码如图上的注记元素所示。语句 'print(next)' 将输出数字系列 2、3、5、8、13、21、34、55、89、144。

此流程的 BPSim 配置按此处所述进行设置。

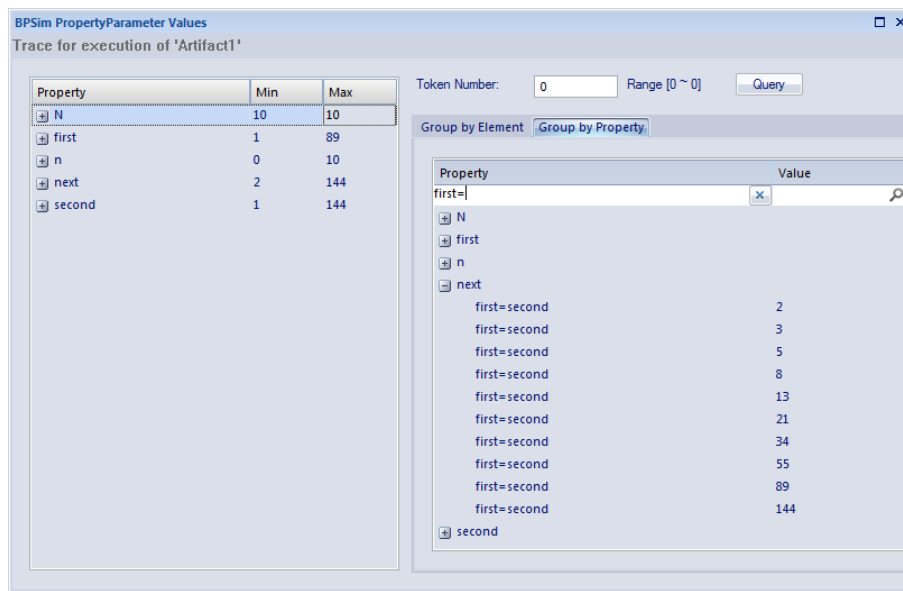
节	行动
1	



	 <p>在 StartEvent 的 元素控件“选项卡上，将 控件”设置为 ‘1’，并在 属性”选项卡上创建并初始化属性：</p> <ul style="list-style-type: none"> <li>• 'N' 作为 '10'</li> <li>• '第一'作为' 1 '</li> <li>• '第二'作为' 1 '</li> <li>• 'n' 作为 '0'</li> </ul> 
<p>2</p>	<p>现在在 属性”选项卡上为流程中的每个活动定义属性。笔记这些属性的值是从<b>Expressions</b>派生的，其组件必须用大括号括起来 - {xxx}。对于活动：</p> <ul style="list-style-type: none"> <li>• next=first+second - 设置属性'next'并将值定义为表达式 {first} + {second}</li> </ul>

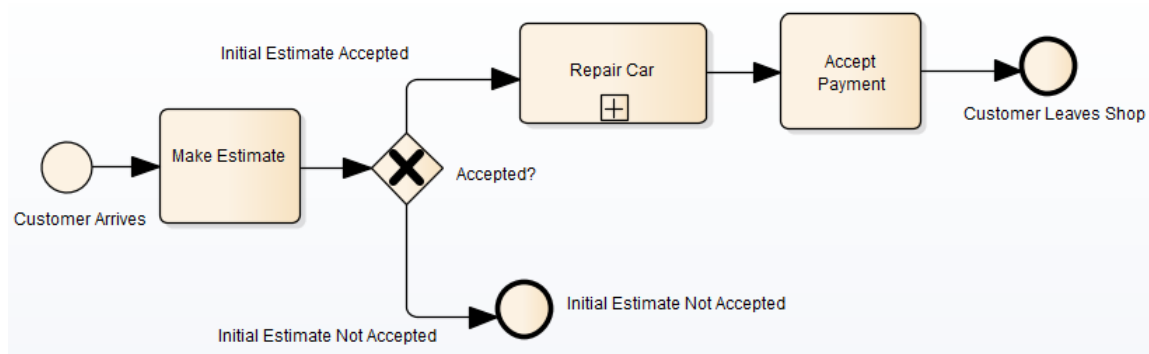
	 <ul style="list-style-type: none"> <li>• first=second - 设置属性 'first' 并将值定义为表达式 {second}</li> <li>• second=next - 设置属性 'second' 并将值定义为表达式 {next}</li> <li>• n++ - 设置属性 'n' 并将值定义为表达式 {n} + 1</li> </ul>
3	<p>在 条件控件条件元素发出的两个 Sequenceflow 连接器设置 属性参数”。 扩展网关   loopNode元素和链接到：</p> <ul style="list-style-type: none"> <li>• next=first+控件- 将参数设置为 条件”并将值定义为表达式 {n}&lt;{ N }</li> </ul>  <ul style="list-style-type: none"> <li>• 控件- 将控件参数设置为 条件”并将值定义为表达式 {n}&gt;{ N }</li> </ul>
4	<p>完成配置后，单击 仿真仿真模拟控制器”对话框中的配置 BPSim上的运行按钮，选择一个标准模拟。 模拟完成后，单击 属性”按钮。 在 BPSim PropertyParameter Values”对话框中，将 Token number”字段设置为“0”，然后单击查询按钮。</p>
5	<p>现在检查在模拟的每次迭代中输入 first=second活动的 下一个”属性的值。单击 按属性分组”选项卡并展开 下一个”项。</p>

值列表很长，因此右键单击列标题并选择“切换过滤器栏”选项。在“属性”列标题下，键入 `first=`。这会过滤列表以仅在输入 `first=` 活动时显示属性参数值。

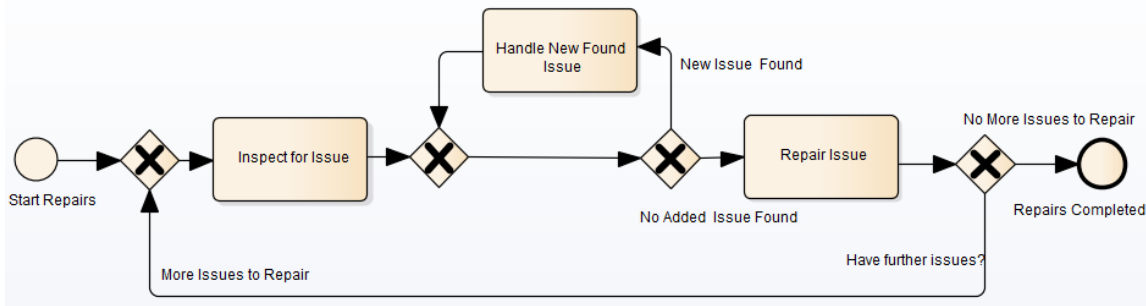


### 汽车维修示例

这个更复杂的示例基于汽车维修过程的真实模型，其中许多个人客户要求维修估算，然后继续维修或拒绝继续；您可以看到属性参数如何随着过程中做出的不同决策而变化。打开示例>模型仿真>汽车进程示例模型。整个流程如下图所示：



*Repair Car* 活动是一个复合元素，包含以下子流程图：



节	行动
1	在浏览器窗口中，展开 汽车维修进程“包下的 浏览器工件”子包，双击 场景1：主流程”。将显示配置 BPSim窗口。在 场景”选项卡上查看 持续时间”字段；这已设置为 2 天 12 小时（即 60 小时）。
2	在窗口左侧的进程层次结构中，展开 开始事件”类别并单击 顾客到达”。选择 控件”选项卡，查看 控件”参数，其值为24分钟；也就是说，每 24 分钟就有一位顾客到达（所以在 60 小时的时间里，有 150 位顾客经过维修店）。 每个客户带着一个或多个需要评估和维修的问题进入维修店。每个客户提出的问题数量可以使用 BPSim 支持的发行版之一随机生成。由于问题编号以离散单位计算（而不是按连续规模测量），我们将使用 整数”分布。如果您为 到达”开始事件选择 顾客属性”选项卡，您将看到 属性”属性值是从平均值为 3 的泊松分布初始化的。
3	现在扩展决定网关 接受？”及其连接器，在流程层次结构中。'In Estimate概率'有一个控件参数'，设置为0.67。替代的 初始概率连接器未接受”设置为控件。也就是说，我们预计平均有三分之一的问题会被客户撤回或不追究。
4	在此过程中，当评估车辆上的问题时，可能会发现另一个问题。 在网关元素列表中，最后一个 朱命名元素”有两个路径： 新问题发现”和 未添加问题”。单击每个这些并查看 控件”选项卡；'New QuestionFound'的问题'参数设置为0.25，对于'问题概率'，设置为 0.75。因此，平均而言，每报告和评估四个问题，就会发现一个新问题。 新问题发现”路径将流程带到 处理新发现问题”活动，这会将要为当前客户处理的问题数量加1。展开活动组，然后单击 处理新发现的问题”元素和 属性”选项卡。您将看到此处的属性 “noOfIssues”具有表达式 值 {noOfIssues} + 1。
5	当车辆出现问题时， 维修”问题活动会从当前客户需要维修的问题数量中扣除1。单击活动组中的 修复问题”元素和 属性”选项卡。您将看到此处的属性 “noOfIssues”具有表达式 值 {noOfIssues} - 1。
6	来自 修复问题”活动的值在 还有其他问题？”中测试。网关。 单击 要修复的更多问题”连接器和 控件”选项卡；遵循此路径的条件参数设置为表达式值 {条件} > 0；流程在 检查问题”活动之前传递到网关。 同样，如果您单击 No More Issues to Repair”连接器并在 控件”选项卡上，遵循该路径的条件参数设置为表达式值 {条件} =< 0，并且流程传递到 修复”完成'结束事件。 现在您已经检查了流程和配置设置，您可以运行模拟并审阅结果。
7	在配置 BPSim窗口，点击运行按钮，然后在 运行仿真仿真控制器”对话框中，再次点击 “Standard”模拟（但模拟类型对属性参数没有影响）。 在 仿真仿真控制器”对话框中，您可以审阅令牌牌状态（并看到另一个客户在最后一分钟设法进入商店），但很难确切地看到这些汇总数据是如何产生的。单击 属性”按钮可在 属性 PropertyParameter Values”对话框中获取详细的属性参数值信息。
8	对话框的左侧是进程中每个元素的属性参数（属性）的最小值和最大值的摘要。例如，对于 顾客

	<p>到达“元素· 顾客”参数的最 值为 0· 最大值为 8· 由泊松 (3) 分布生成。</p> <p>在“令牌编号”字段中· 输入一个介于 0 和 150 之间的数字 (N)· 为进入维修店的第 N 个客户选择。单击查询按钮以获取该客户在流程中使用的属性参数值。在两个选项卡中的每个选项卡上审阅结果：</p> <ul style="list-style-type: none"> <li>在“按元素分组”选项卡上· 查看每个元素中属性的值如何变化；例如· 对于客户 24· “noOfIssues”参数由 机分布初始化为 4· 并且“检查问题”活动被调用六次· 其中三个调用的参数值在循环前被调整为 3到1· 并且“处理新发现问题”活动被调用两次· 参数值都为 3</li> <li>在“按属性分组”选项卡上· 查看参数值如何随着流程循环完成活动而变化· 从 4 开始· 在 3 和 4 之间多次调整· 然后在结束时递减到 0</li> </ul>
9	<p>根据需要进行探索结果· 选择不同的客户 (代币)。您还可以返回到 BPSim 配置并更改参数初始化并添加新参数· 或更改决策点· 以试验该过程。</p>

## 在基于整数的过程的模拟中响应实数

在某些情况下· 当流程中的活动使用整数运行时· 或者当您想查看强制整数值对流程的影响时· 您可能需要使用返回“实数”的分布来生成属性参数值。

在这种情况下应用的一种机制是设置条件以避免绝对数字。因此· 例如· 您可能有一个递减1的计数器· 它被初始化为一个“实数”。如果将条件设置为 'value==0' (等于 0) 或 'value !=0' (不等于 0)· 这两个条件可能永远不会分别为 True 或始终为 True· 从而导致无限循环。为避免这种情况· 您将使用以下运算符：

'值 > 0'

'值 < 0'

'值 >= 0'

'值 <= 0'

另一种机制是编辑 BPSim 引擎使用的代码模板· 截取提供给特定参数的实数并将其替换为整数· 如图所示：

- 选择 开发 > 源代码 > 选项 > 编辑代码模板”功能区选项。
- 在代码模板编辑器的“语言”字段中· 单击下拉箭头并选择“MDGBPSimExecutionEngineExtension”。
- 在 (Java) 模板列表中· 单击“MDGBPSimExecutionEngineExtension Compute Value”。模板内容显示在“模板”面板中。
- 找到这一行：

```
double %bpsimPropertyParameterName% = (double) distribution.next();
```

将其更改为：

```
%如果bpsimPropertyParameterName == "noOfIssues" 或 bpsimPropertyParameterName == "noOfVisitors"%
双 %bpsimPropertyParameterName% = ( int ) distribution.next();
//double %bpsimPropertyParameterName% = Math. ceil (distribution.next());
//double %bpsimPropertyParameterName% = Math. floor ( distribution.next ( ) ) ;
//double %bpsimPropertyParameterName% = Math. round ( distribution.next() ) ;
% else %
double %bpsimPropertyParameterName% = (double) distribution.next();
%结束如果%
```

- 将属性参数名称替换为您自己的属性参数。
- 单击保存按钮· 关闭代码模板编辑器并重新加载项目。

如前所述· 对于每个指定的参数· 代码模板将简单地将任何由分布初始化的“实数”替换为整数。如果您愿意· 可以改用注释行之一：

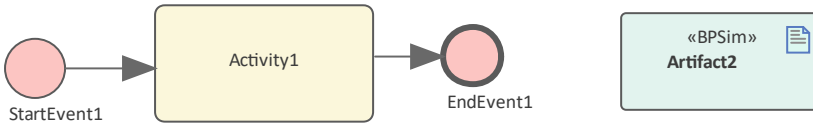
- 数学· `ceil ()` 将获取“真实”数字并将其转换为下一个最大整数

- 数学。 floor () 将采用 实数”并将其转换为下一个最小整数
- 数学。 round () 将取 实数”并根据它是否大于或小于 n.5 向上或向下round入

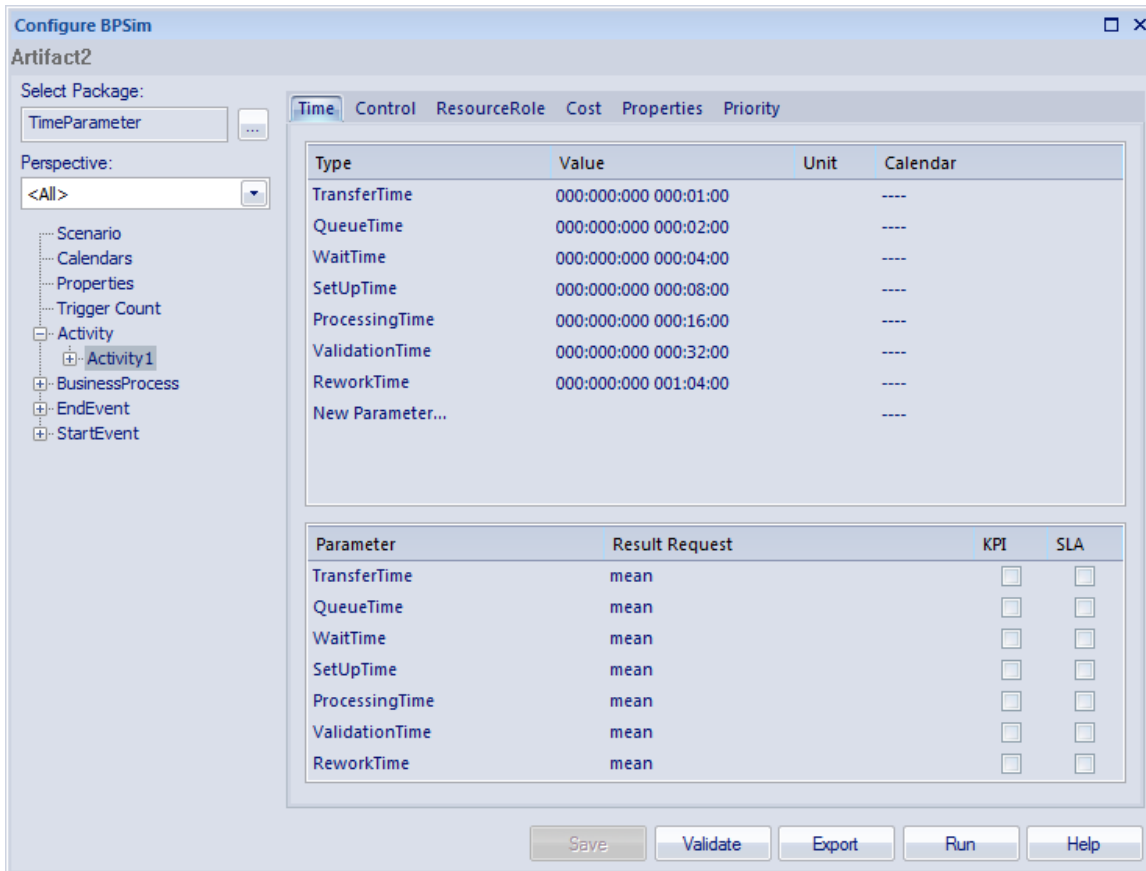
### 时间参数行为

在 BPSim 配置中，您可以为一个活动设置多个时间参数，例如队列时间和等待时间。您还可以为每个自定义模拟设置结果请求。然而，BPSim 模拟引擎将这些参数组合成一个单一的 处理时间”量。

考虑示例模型中的简单模型时间参数（示例模型>模型仿真>示例模型>时间参数），如下图所示：




如果双击 Artifact2元素，将显示配置 BPSim窗口。单击图中的 Activity1元素以展开活动组，在对话框左侧的层次结构中选择 Activity1，并显示配置中元素的第一个选项卡“时间”，如图所示。



注记在上面的面板中有七个系统提供的“时间”参数，它们的初始值依次为 - 1和 64 分钟（64 分钟为1小时）和 4 分钟）。注记在下面的面板中，每一个都有一个参数的平均运行时间值的结果请求。

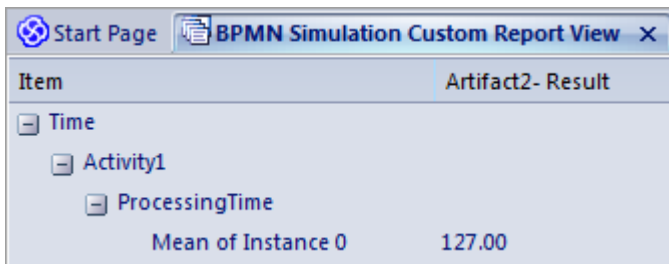
点击运行按钮，在“运行仿真控制器”对话框中点击运行按钮，选择“标准仿真”。模拟被配置为循环通过该过程一次。模拟完成后，点击“打开结果”按钮，在“BPMN仿真报告视图”上单击鼠标右键，选择“仅显示非空项”选项。对于设置了参数的 Activity1元素，这将为为您提供以下结果：



Item	Artifact2- Result
Time	
Activity1	
Mean Of Processing Time	127
Average Time In Task	127
Max Of Processing Time	127
Maximum Time In Task	127
Min Of Processing Time	127
Minimum Time In Task	127
Sum Of Processing Time	127
Total Time In Task	127

所有这些导出的结果都是 127 分钟，即原始七个“时间”参数的初始值之和。各个参数不单独处理。

如果您返回“仿真仿真控制器运行”对话框并单击“仿真仿真运行”按钮，这次选择“仿真”，“打开结果”按钮将显示“BPMN仿真报告视图”。在配置中，结果请求是针对七个参数的平均值。在模拟的报告视图中，您只能看到单个聚合参数 ProcessingTime 的平均值为 127 分钟。



Item	Artifact2- Result
Time	
Activity1	
ProcessingTime	
Mean of Instance 0	127.00

# 比较 BPSim 配置

当您开发 BPSim 配置时，您可以在运行模拟并观察这些设置和所选设置更改的影响之前定义要设置的范围广泛的参数。为了便于管理多个“假设”场景，建议您创建原始配置的副本（作为工件元素和设置更改）并在副本中进行设置。

创建配置变体的有用功能是应用继承，即您不打算改变的数据和参数保存在一个配置中，而只有您更改的那些参数保存在另一个配置中。“变量”配置使用（继承）基本配置中保存的公共数据，因此您不必在“变量”配置中重新创建该公共数据。

然后，您可以在更改的配置和原始“运行”上运行模拟，并比较模拟报告以查看运行时变量发生了什么差异，然后运行并显示配置的比较以查看参数设置的哪些变化导致了这些运行时差异。

通过在原始配置和副本配置下运行模拟，比较结果和导致结果的更改，并相应地修改模型，您可以在简化您正在开发的业务流程方面实现高度控制。

## 访问


上下文菜单	在图表或浏览器窗口中 右键点击业务流程仿真工件 显示配置>审阅选项卡>配置摘要选项卡
-------	--

## 配置摘要选项卡

此视图最初显示所选配置中具有值的参数，并在配置名称下的列中显示为这些参数设置的值。如果窗口停靠，您可以通过将选项卡拖离工作区以成为浮动显示，并将显示扩展到全屏大小，从而更轻松地审阅结果。

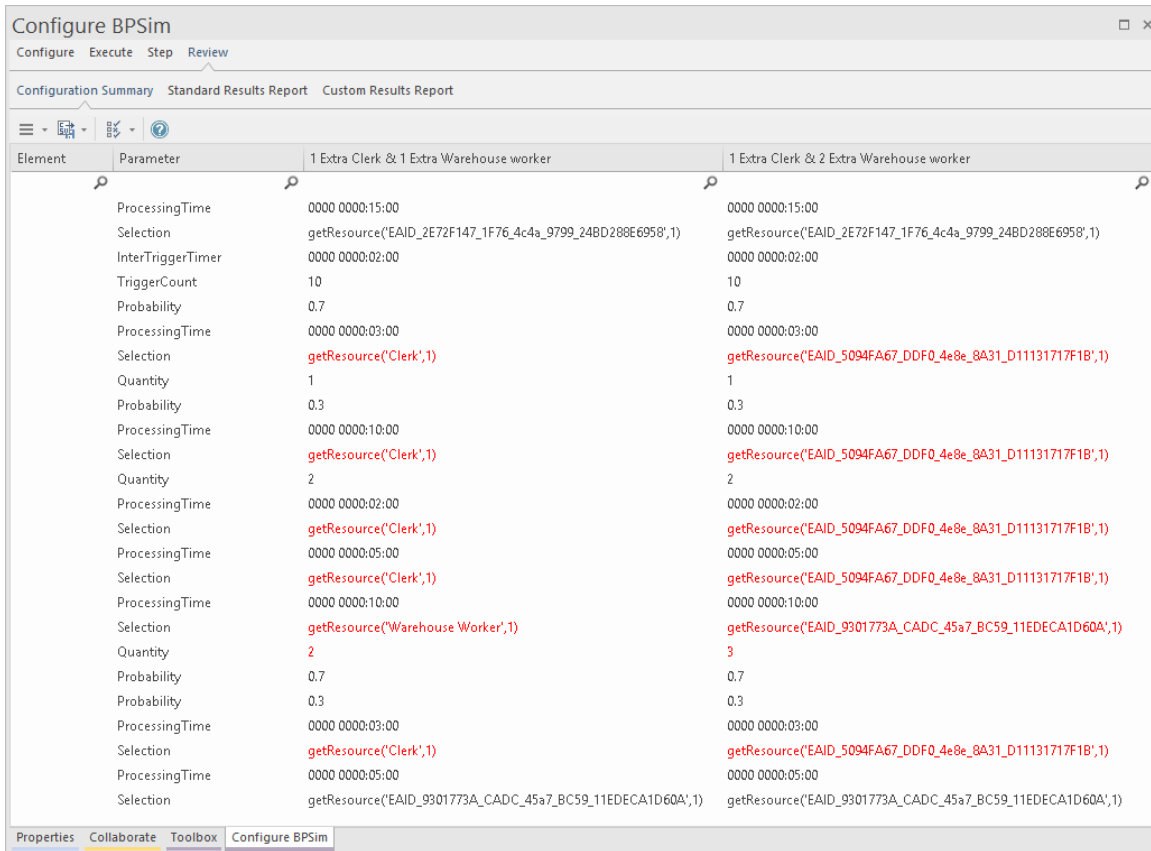
要比较两个或更多配置，请单击另一个配置工件（或单击另一个浏览器元素中的一个或一个）

- 将其拖到报告视图或
- 选择“显示配置”选项

您还可以在“配置摘要”选项卡上的  图标，单击“添加配置配置”选项并浏览并选择“工件元素”对话框上的“元素”。




参数层次结构现在包含任何附加配置，并且其参数值显示在原始配置的列中，参数值的名称位于该工件标题的左侧。





您可以使用工具栏和右键单击上下文菜单中的可用选项来审阅和操作报告上的信息。

### BPMN仿真配置汇总选项

选项	描述
	选择此工具栏选项以显示选项的简短菜单： <ul style="list-style-type: none"> <li>• 添加配置- 显示“无素选择”对话框，您可以通过该对话框浏览另一个工件并选择那些已经在页面旁边显示其参数的 BPSim 配置</li> <li>• 重新加载 - 刷新显示并将列更改返回到原始宽度和位置</li> <li>• 清除列表 - 从显示中清除选定的工件及其参数</li> <li>• 删除报告名称；从页面中清除报告名称工件参数为页面上显示的每个工件提供的选项之一</li> </ul>
	单击此工具栏图标可将此页面上的记录导出为 CSV 文件或 XML 文件。将显示“另存为”浏览器，您可以从中选择保存文件的位置。
	显示两个选项，用于过滤页面上显示的信息；当显示两个或更多配置时，将启用这两个选项： <ul style="list-style-type: none"> <li>• 仅显示不同项- 选择此选项将显示限制为仅显示配置之间值不同的参数；再次单击该选项可取消选择</li> <li>• 突出显示不同项- (当两个或多个配置具有不同的参数值时) 以红色显示不同的参数值；如果您已选择“仅显示不同项”选项，则此选项将被禁用</li> </ul>

The screenshot shows the 'Configure BPSim' application window. It has a menu bar with 'Configure', 'Execute', 'Step', and 'Review'. Below the menu bar are tabs for 'Configuration Summary', 'Standard Results Report', and 'Custom Results Report'. The main area contains a table with the following data:

Element	Parameter	1 Extra Clerk & 2 Extra Warehouse worker	1 Extra Clerk & 1 Extra Warehouse worker
Selection		<code>getResource('EAID_5094FA67_DDF0_4e8e_8A31_D11131717F1B',1)</code>	<code>getResource('Clerk',1)</code>
Selection		<code>getResource('EAID_5094FA67_DDF0_4e8e_8A31_D11131717F1B',1)</code>	<code>getResource('Clerk',1)</code>
Selection		<code>getResource('EAID_5094FA67_DDF0_4e8e_8A31_D11131717F1B',1)</code>	<code>getResource('Clerk',1)</code>
Selection		<code>getResource('EAID_5094FA67_DDF0_4e8e_8A31_D11131717F1B',1)</code>	<code>getResource('Clerk',1)</code>
Selection		<code>getResource('EAID_9301773A_CADC_45a7_BC59_11EDECA1D60A',1)</code>	<code>getResource('Warehouse Worker',1)</code>
Quantity		3	2
Selection		<code>getResource('EAID_5094FA67_DDF0_4e8e_8A31_D11131717F1B',1)</code>	<code>getResource('Clerk',1)</code>

# BPSim 图表

图表工具箱的“图表”页面提供了两个图标，专门用于生成反映从图表模拟中选择的结果的图表。这些是：

- 图表结果图表 - 生成反映一系列标准图表模拟的选定结果的图表
- 图表自定义结果图表 - 生成反映一系列自定义图表模拟结果的图表

至于其他维度的工件，图表图表都可以快速配置为以时间折线图或 3 维图表图表变化形式显示模拟结果。

## 先决条件

要填充从业务流程仿真工件的工件中创建的结果，您可以选择在模拟过程中显示的每个结果创建配置。因此，必须首先执行初始模拟，工件生成报告。

## 访问

使用本表中工具箱图表“图表”页面。

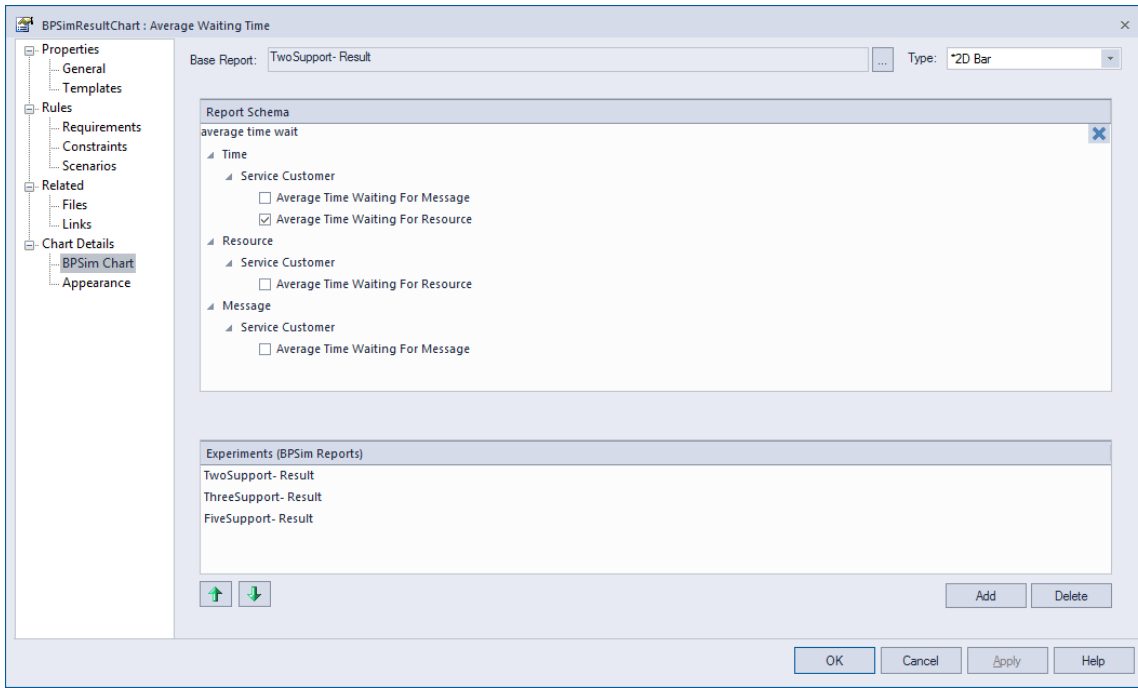
然后，将新的图表<type>工件图标拖到图表上——创建了一个新图表元素图标。

双击新的图形元素打开“属性”对话框，显示“图表图表”页面。

功能区	设计>图表>工具箱> 图表
键盘快捷键	Ctrl+Shift+3  图表
其它	您可以通过单击 >> << 图表工具箱显示或隐藏图形图表视图。

## 选择要在图表中显示的结果

填写“属性”对话框的“完全图表”页面上的图表。




选项	行动
基础报告	单击 在<工件>>报表 [...] 上选择“按钮显示 选择报表工件”对话框，并选择其他模拟结果将与之比较的模拟结果的报表。该结果作为“实验”工件中的第一个添加到要比较的报告结果列表中
类型	单击下拉箭头并选择要显示结果的图表的格式类型 - 2D Bar、3D Bar 或 Line。指定要比较的报告参数后，您可以选择对话框的“外观”页面并定义图表或时间线图的外观。
报告架构	根据需要展开层次结构并选中每个属性的复选框以显示在图表上。每个属性将由图表上的单独一行或一组图表表示。通常你会为每个object选择相似的对象（比如不同的资源）和相同的单个属性（比如资源的利用率）。你有大量的属性可以检查和比较，但是在同一个图表上的任何一个以上都会使图表难以阅读。
实验（BPSim 报告）	此模拟报告结果（如工件Result 面板将列出使用您选择的要比较的图表的模拟报告结果），在图表中显示它们选择的图表的序列通常，基础报告在列表中保持在第一位，其参数的结果显示在图表的左侧。如果要更改序列，请单击结果/工件名称并单击相应的向上向下绿色箭头按钮。 要添加结果名称，请单击“将结果名称添加到列表”工件从 <工件>> 选择工件”对话框中选择更多按钮。

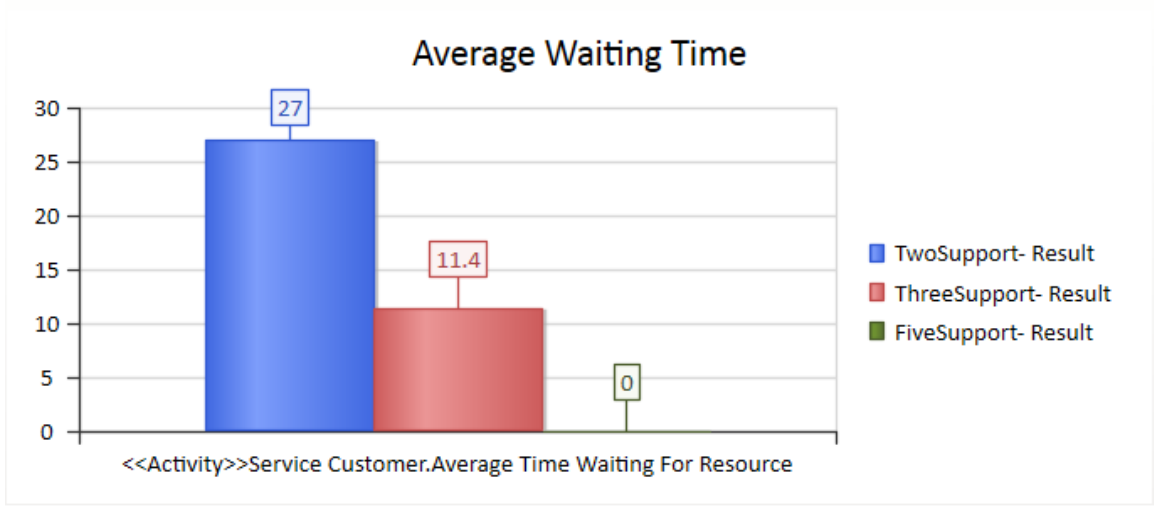
### A图表示例

在电话支持示例中，我们创建了三个服务台工件来分析“我们需要多少支持资源才能以经济的方式通过电话回答客户的问题？”


我们从两个支持资源开始，然后尝试了三个和五个资源。模拟后，我们有一个基于不同配置的 BPSim 报告：TwoSupport-Result、ThreeSupport-Result 和 FiveSupport-Result。

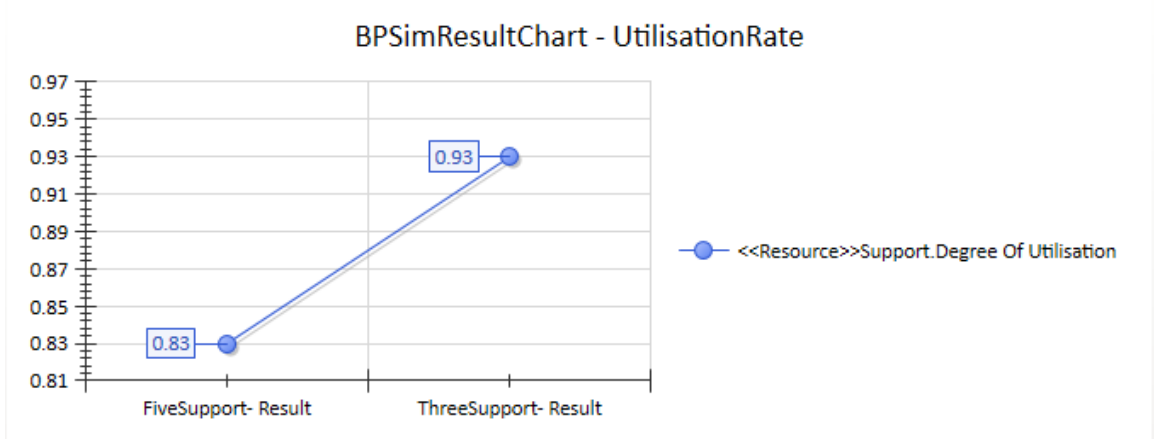
以下是创建图表以比较客户等待支持的平均时间的步骤：

1. 在图表上创建一个图表结果图表，并将其命名为平均等待时间。
2. 双击 Figure 打开属性窗口，然后打开 图表图表”选项卡。
3. 单击  按钮选择一个基本报告，我们从中定义要在图表中使用的模式（图例）。选择 “双支持结果”。
4. 选择此架构：  
-时间|服务顾客|平均时间等待资源
5. 单击 Add 按钮添加另外两个 BPSim 报告：“ThreeSupport-Result”和 FiveSupport-Result”。
6. 单击确定按钮，然后调整图表元素的大小。这个图表给了我们直接的信息。



这些是创建一个图表来比较支持的使用程度的步骤：

1. 在图表上创建一个图表Result Figure，并将其命名为Utilization Rate。
2. 双击 Figure 打开属性窗口，然后打开 图表图表”选项卡。
3. 单击  按钮并选择一个 Base Report，从中定义要在图表中使用的模式（图例）。选择 “双支持结果”。
4. 选择此架构：  
-资源 |支持 |利用度
5. 单击 Add 按钮添加另外两个 BPSim 报告：“ThreeSupport-Result”和 FiveSupport-Result”。
6. 单击确定按钮并调整图表元素的大小。该图表提供了具体信息。



## BPSim 示例

本节提供了几个 BPMN 建模、配置和仿真结果分析的示例。

这些示例都可以从 EAExample模型中访问。

要运行示例模拟，您必须安装运行引擎和Java运行时环境(JRE) 1.7 或更高版本。要使用属性参数，您还必须安装Java开发工具包 (JDK) 版本1.7 或更高版本。

# 订餐协作版本1

在此示例中，我们创建了一个非常简单的模型来模拟客户与餐厅之间就餐单进行的通信。

对于客户的流程：

1. A向餐厅发送消息以订餐。
2. 客户将等待交货。  
如果60分钟内没有送达，他们会打电话给餐厅，然后继续等待。
3. 交货后，客户将享用晚餐。

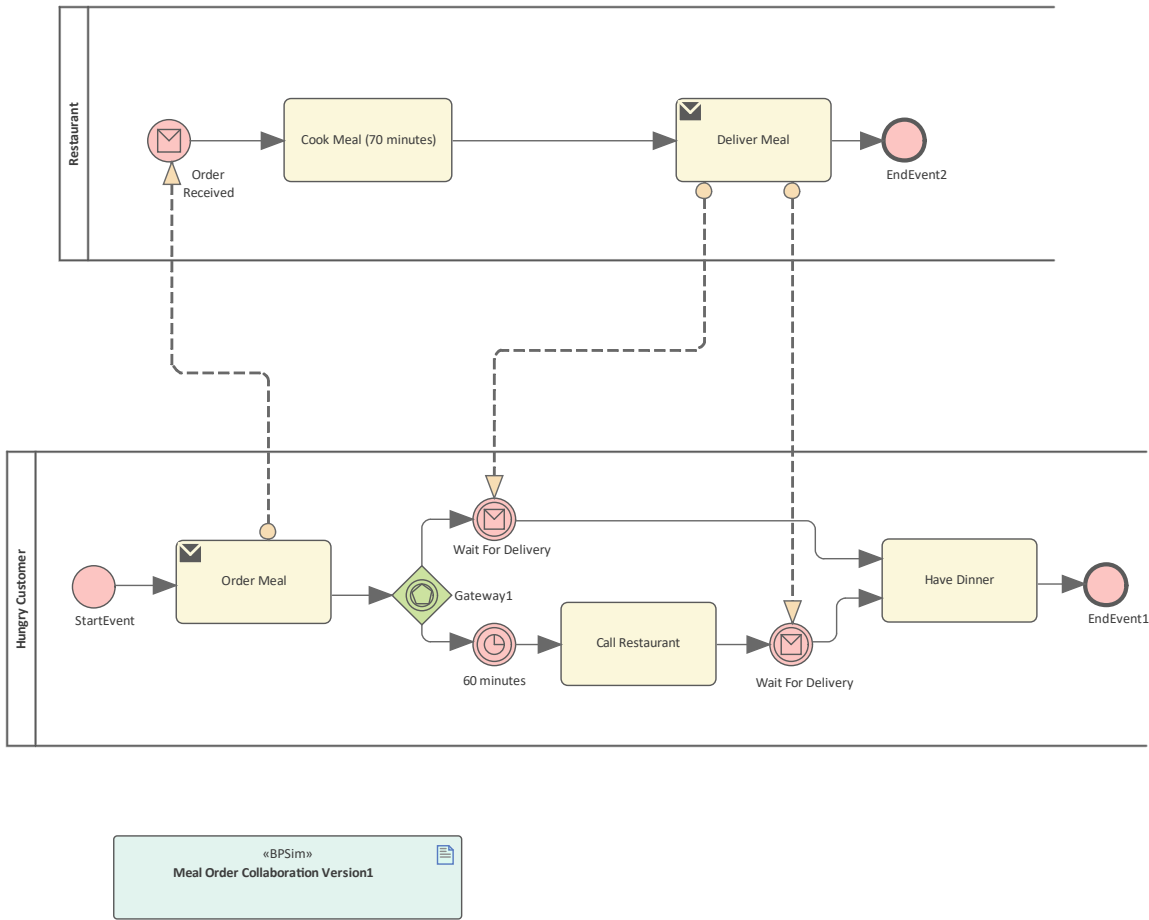
对于餐厅的流程：

1. 该过程从收到客户的订餐单开始。
2. 烹饪时间可由用户设定。这允许用不同的事件时间进行实验，例如 30 分钟、70 分钟
3. 餐厅送餐并完成流程。

## 创建 BPMN模型

要设置可用于此 BPSim 模拟的 BPMN模型，您：

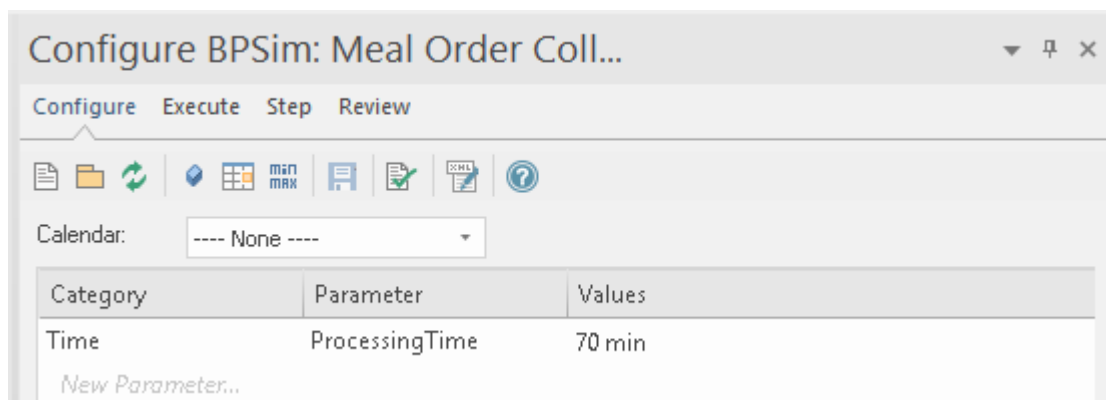
- 创建一个带有 2 个池的协作模型
- 在每个泳池中为每个过程创建一个元素
- 将元素与流程通信的信息流连接起来
- 包括用于设置模拟细节的 BPSim工件。



### 配置 BPSim

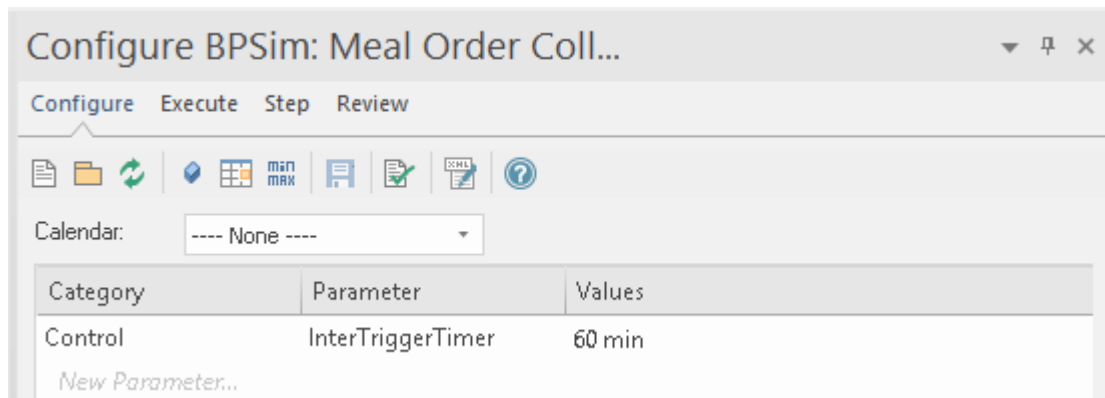
在此示例中，我们配置以下 BPSim 参数：

- 设置饿了么顾客中StartEvent的TriggerCount为1
- 将 Cook Meal 的 ProcessingTime 设置为 70 分钟



- 将中间事件的 InterTriggerTimer 设置为 60 分钟





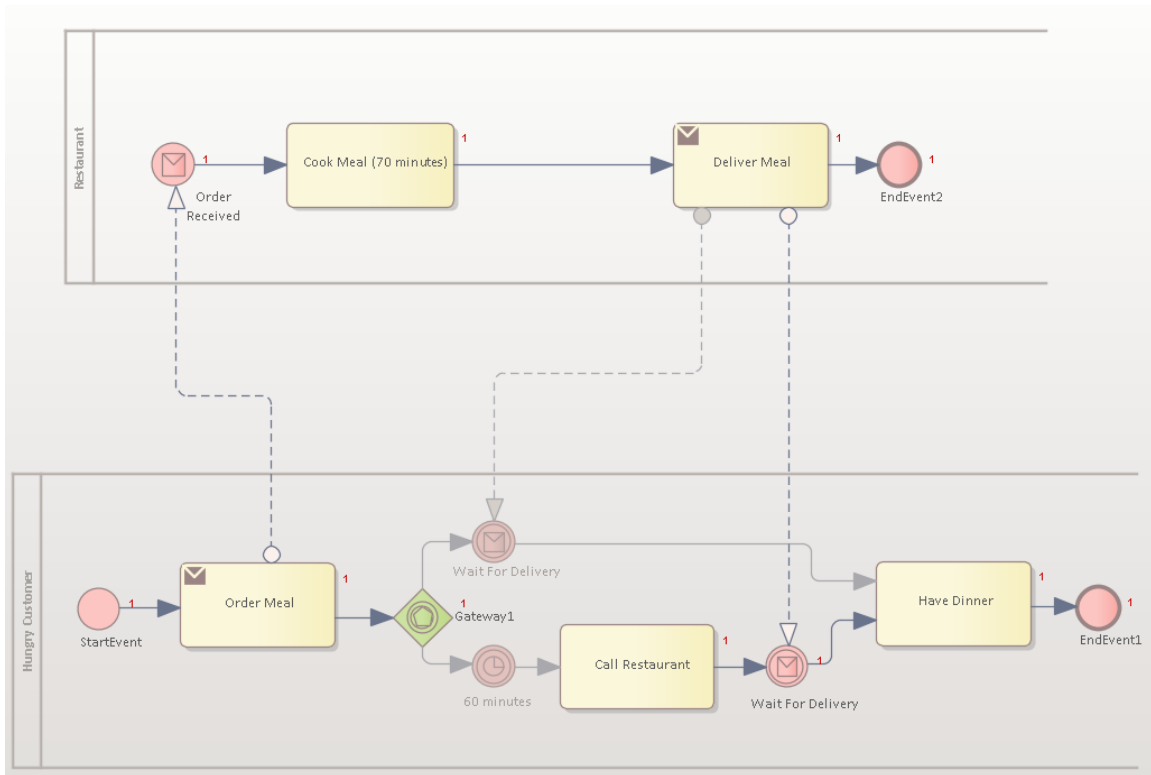
- 其他BPSim参数的默认设置，这里是一个配置列表，你可以查看through审阅>配置Summary



## 仿真

确保 Config BPSim 窗口已打开（仿真>进程分析> BPSIM > 打开 BPSIM 管理器）。

导航到执行选项卡并运行标准仿真：



当 Cooking Meal 任务耗时 70 分钟时，独家事件网关由 60 分钟计时器事件触发。

如果我们将 BPSim 设置更改为任务：烹饪膳食：处理时间从 70 分钟到 30 分钟，消息事件 *Wait For Delivery* 将触发 Exclusive 事件 Gateway，并且根本不会激活调用餐厅任务。

# 订餐协作版本2

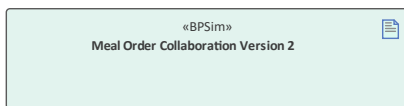
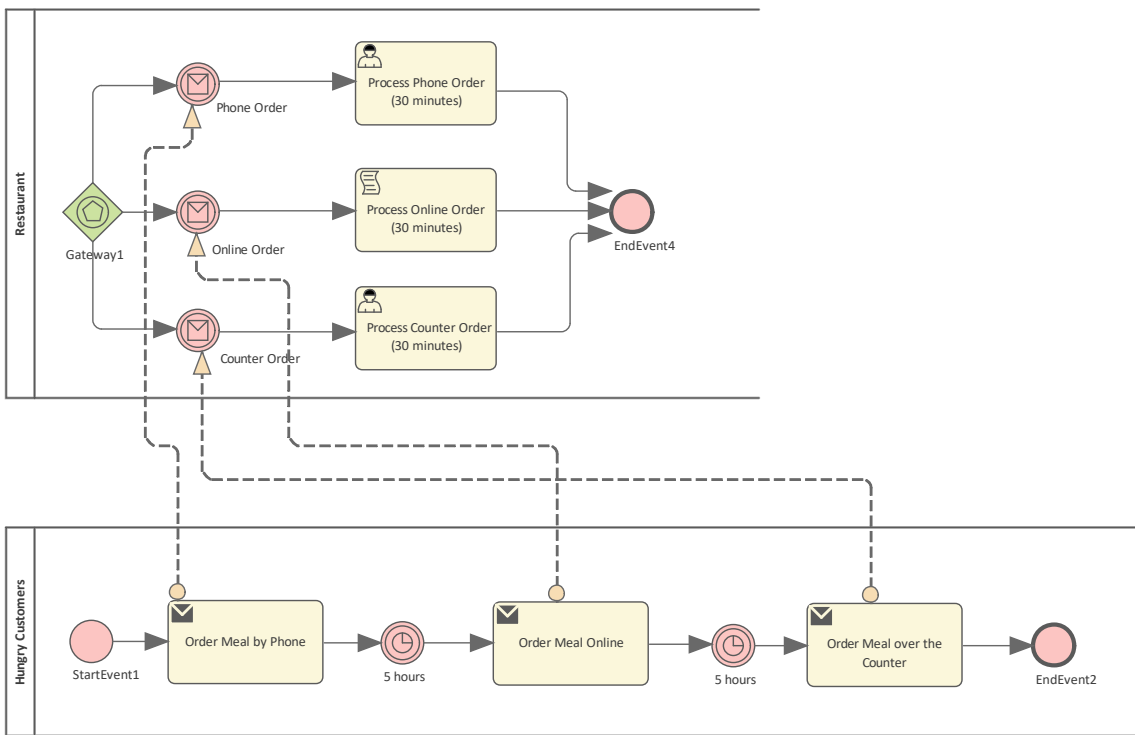
在此示例中，我们创建了一个模型来模拟客户与餐厅之间就餐单进行的通信。客户通过电话、在线和柜台下三种不同的订单。

## 创建 BPMN模型

要设置可用于此 BPSim 模拟的 BPMN模型，您：

- 创建一个带有 2 个池的协作模型
- 在每个池中为每个进程创建一个元素
- 创建描述进程通信的信息流
- 包括用于设置模拟细节的 BPSim工件。

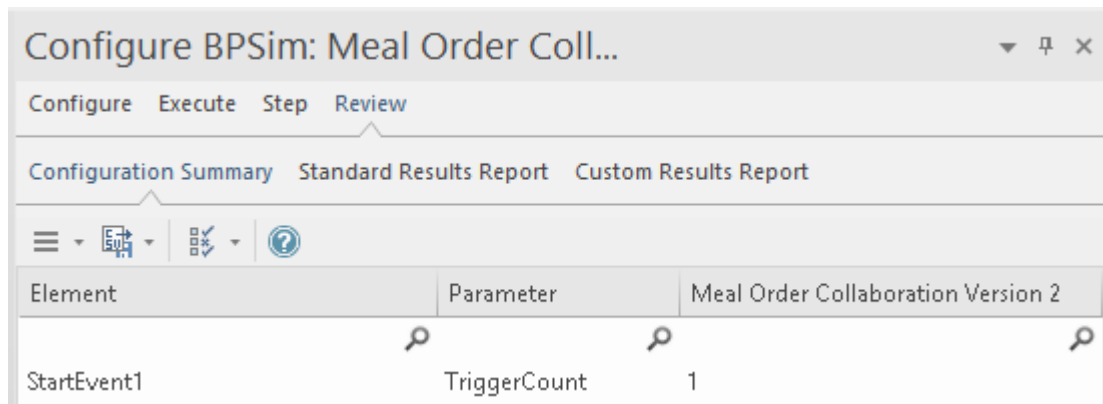
请记住：Event-Exclusive gateway 的 *instantiate* 属性被设置为 **true**，这意味着进程是由这个开始事件 *Gateway* 启动的。



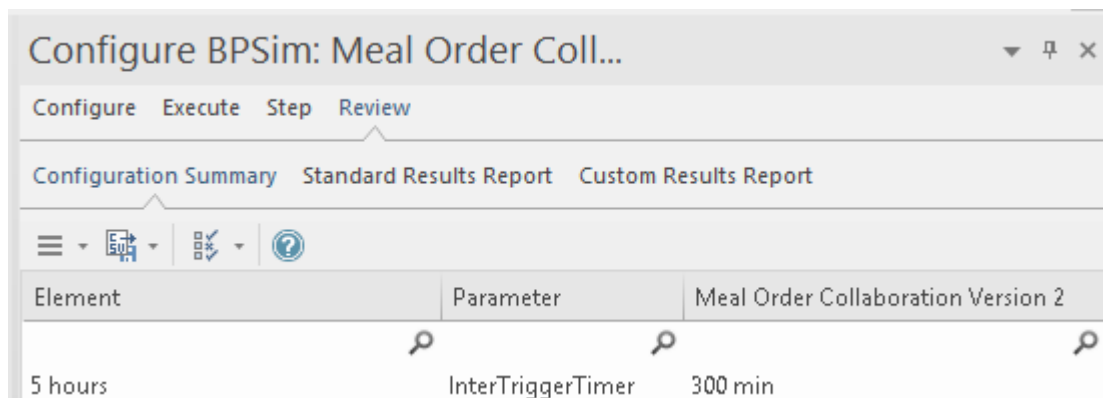
## 配置 BPSim

在此示例中，我们配置以下 BPSim 参数：

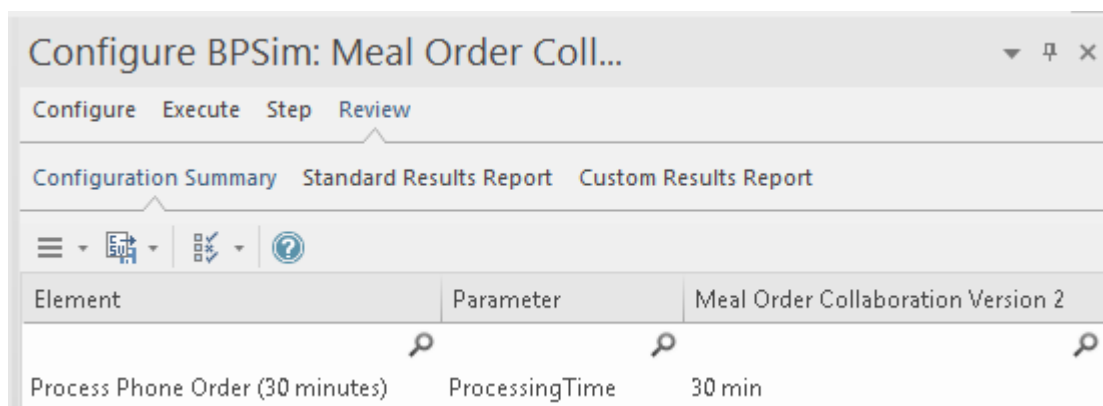
- 对于开始事件，将参数 *TriggerCount* 设置为值 1



- 对于中间事件中的两个 5 小时，将参数 InterTriggerTimer 设置为值 300 分钟



- 对于 Restaurant 中的三个任务，将参数 ProcessingTime 设置为值 30 分钟

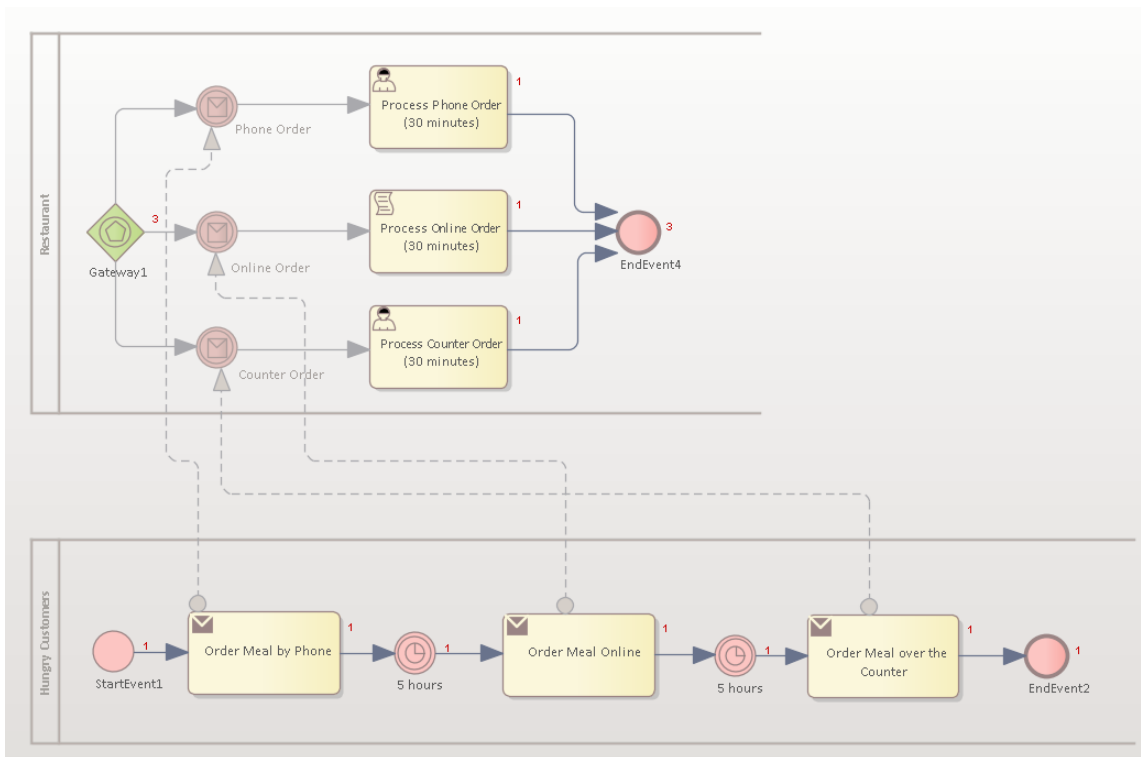


- 其他BPSim参数的默认设置，这里是一个配置列表，你可以查看[through审阅>配置Summary](#)

Configure BPSim: Meal Order Coll...		
Configure   Execute   Step   Review		
Configuration Summary   Standard Results Report   Custom Results Report		
Element	Parameter	Meal Order Collaboration Version 2
5 hours	InterTriggerTimer	300 min
5 hours	InterTriggerTimer	300 min
Process Counter Order (30 minutes)	ProcessingTime	30 min
Process Online Order (30 minutes)	ProcessingTime	30 min
Process Phone Order (30 minutes)	ProcessingTime	30 min
StartEvent1	TriggerCount	1

### 仿真

导航到执行选项卡并运行标准仿真：



Restaurant进程被 Event-Exxclusive 网关通过 3 条不同的消息实例化了 3 次。

# 服务台电话支持仿真

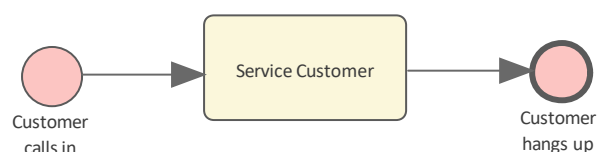
在本例中，我们创建了一个非常简单的模型来模拟服务台电话支持过程。

我们设置了一个资源有限的场景，并且必须将请求放入等待队列中以获取资源。然后我们尝试使用假设分析在客户的等待时间和资源数量之间寻求平衡点。

首先，我们一步一步地模拟这个过程，从一个简单的参数设置开始，而不是用笔和纸计算，然后用模型进行验证。之后，我们会进行假设分析，以帮助经理做出决定。

## 创建 BPMN 模型

模型本身很简单，由一个开始事件、一个任务和一个结束事件组成。



- 创建一个名为“顾客”的开始事件
- 将序列流添加到名为Service顾客的目标抽象任务
- 将一个序列流添加到一个名为“顾客挂断”的目标结束事件

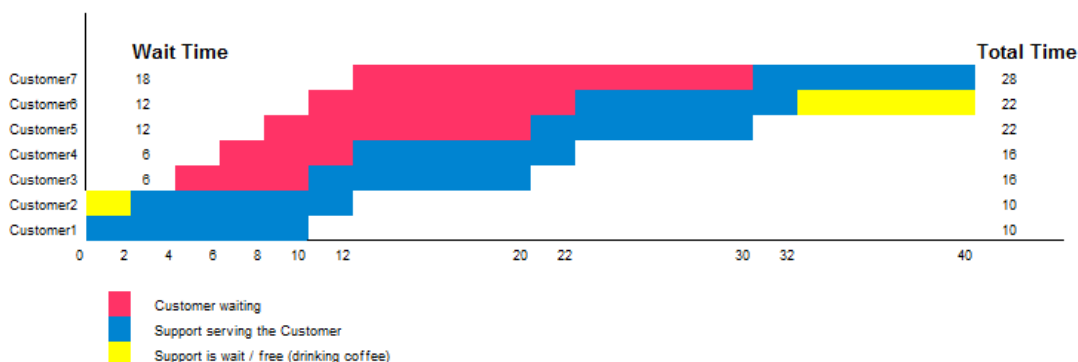
创建一个名为Support的BPMN2.0资源；该元素将在配置中使用。

## 笔和纸分析

我们将用纸笔来分析这个案例：

- 7位客户每2分钟打来一次电话
- 2个支持资源可用
- 每次服务需要10分钟

## Pen & Paper Analysis on a BPSim Example






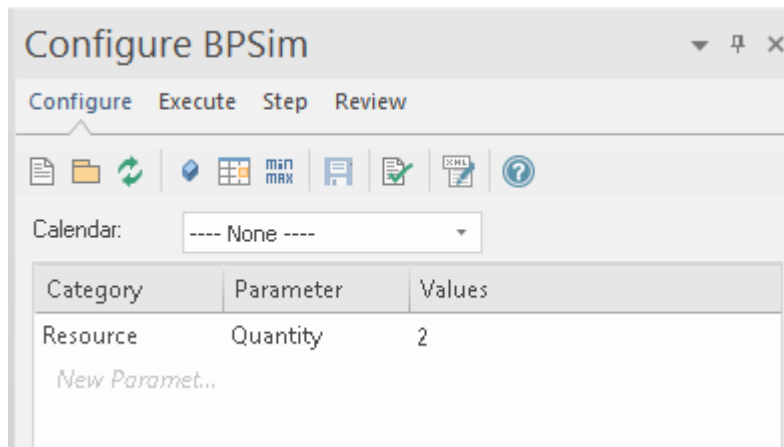
Now we can perform some calculations using a pen and paper based approach:



Total Time Waiting For Resource:	$(18+12+12+6+6+0+0) = 54$	number of <span style="color: red;">■</span>
Total Time In Task:	$(28+22+22+16+16+10+10) = 124$	
Average Time in Task:	$124/7 = 17.71$	
Average Time Waiting For Resource:	$54/7 = 7.71$	
Maximum Time In Task:	28	
Maximum Time Waiting For Resource:	18	
Sum of Support's wait time:	$2+8 = 10$	number of <span style="color: yellow;">■</span>
Sum of Support's time for Task:	$7*10 = 70$	number of <span style="color: blue;">■</span>
Degree of Utilisation:	$70/(10+70) * 100\% = 87.5\%$	

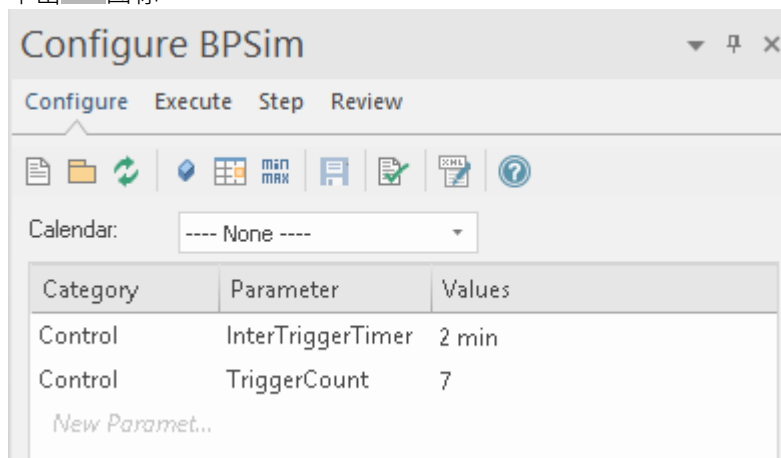
从这个结果可以看出，对于这样一个简单的模型，在应用资源约束的情况下，已经是一个非常复杂的计算了。当流程扩展并应用更多约束时，用笔和纸分析流程将很快变得不可能。我们将演示 BPSim 如何提供帮助。




## 配置

1. 打开配置 BPSim 窗口 ('仿真>过程分析>进程>打开BPSim 管理器')。
2. 单击  "图标和 加新业务流程" 按钮，并在命名为 巴命名的工件 & Paper 分析客户" 的 7 上创建业务流程仿真。
3. 单击  图标并浏览并选择包含相应 BPMN 2.0 模型的包。
4. 打开模型图并单击名为 "支持" 的资源元素。
5. 在窗口的 类别" 列中，单击 新参数" 下拉箭头并选择 资源"，然后单击 参数" 下拉箭头并选择 数量"，然后在 值" 中字段类型 2"。
6. 单击  图标。

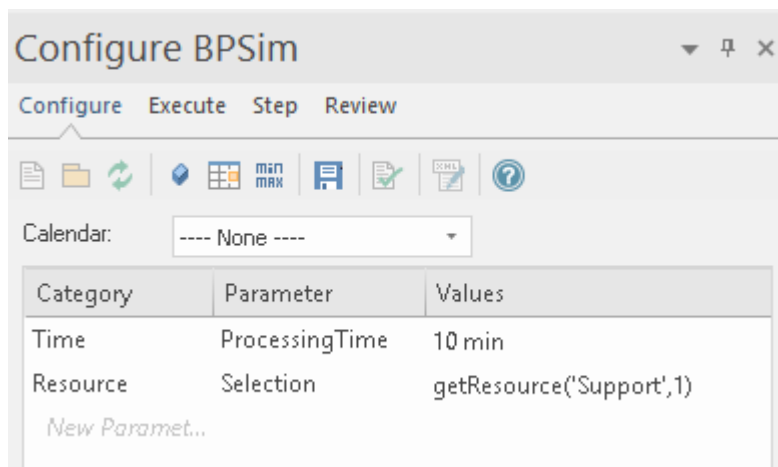


1. 在图中，单击 顾客呼入”开始事件元素。
2. 在窗口的 类别”列中单击 新参数”下拉箭头并选择 控件”。
3. 单击 参数”下拉箭头并选择 InterTriggerTimer”。
4. 在 Value”字段中，单击  按钮，选择 Constant”选项卡和 Numeric”，在 Constant Numeric”字段中输入 “2”并在 TimeUnit”字段中选择 minutes”，然后单击确定按钮。
5. 重复步骤 2 和 3，在 参数”字段中选择 TriggerCount”，并在 值”字段中键入 7”。
6. 单击  图标。





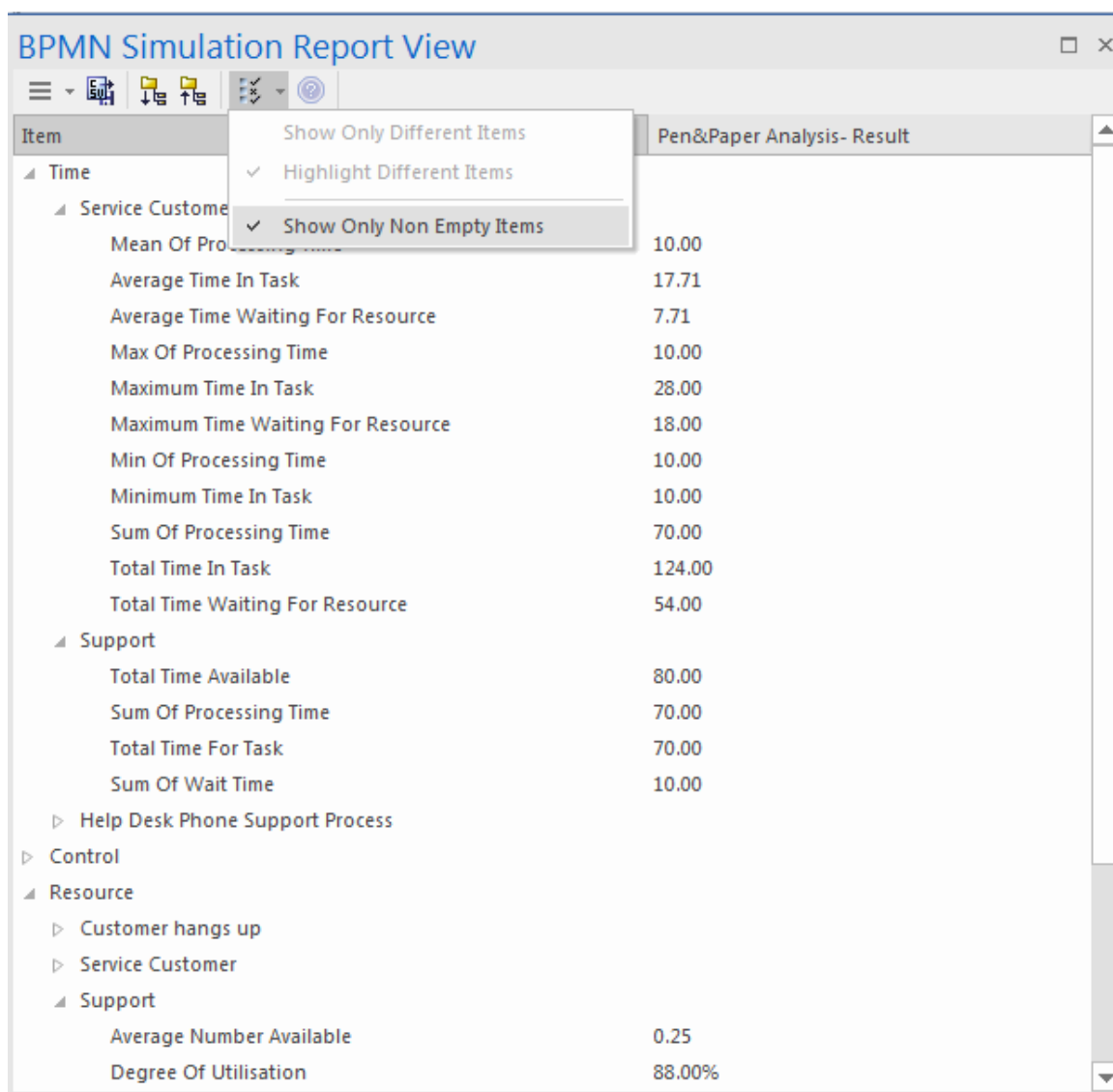
1. 在图中，单击 服务顾客”活动元素。
2. 在窗口的 类别”列中，单击 新参数”下拉箭头并选择 时间”。
3. 单击 参数”下拉箭头并选择 ProcessingTime”。
4. 在 Value”字段中，单击  按钮，选择 Constant”选项卡和 Numeric”，在 Constant Numeric”字段中输入 “10”并在 TimeUnit”字段中选择 minutes”，然后单击确定按钮。
5. 在窗口的 类别”列中，单击 新参数”下拉箭头并选择 资源”。
6. 单击 参数”下拉箭头并选择 选择”。
7. 在 值”字段中，单击  按钮，选择 支持”（您在元素中创建的资源的模型）并单击 按资源添加选择”按钮将 支持”移动到 “资源或角色”列。
8. 在 需要数量”栏中输入 1 。
9. 单击确定按钮。在 值”字段中，将显示自动生成的表达式 `bpsim:getResource('Support', 1)` 。
10. 单击  图标。





### 运行仿真


1. 在配置 BPSim窗口中，单击 执行”选项卡和工具栏中的  图标。
2. 模拟完成  完成以打开 结果”选项卡的 配置摘要”选项卡



结果与笔和纸分析相匹配。

## 仿真- 2个支持资源为20个客户

您可以通过复制现有的工件配置来创建新的业务流程仿真工程师。复制*Pen & 分析客户*元素并按 **Ctrl+Shift+V** 粘贴，将新元素命名为*TwoSupport*。

1. 双击*TwoSupport*打开配置 BPSim窗口；你可以看到所有的配置都保留在复制的源中
2. 在图中，单击“顾客呼入”开始事件元素。
3. 在窗口的“类别”列中，单击“控件”-“触发器计数”的“值”字段，然后将值更改为“20”。
4. 单击  图标。

### 运行仿真并分析结果

BPMN Simulation Report View		BPMN Simulation Report View	
Item	TwoSupport- Result	Item	TwoSupport- Result
wait		util	
Time		Resource	
Service Customer		Support	
Average Time Waiting For Resource	27.00	Degree Of Utilisation	98.00%
Maximum Time Waiting For Resource	54.00		
Total Time Waiting For Resource	540.00		

从报告中，您可以看到：

- “平均等待资源时间”为 27 分钟，“最大时间等待资源”为 54 分钟
- 两个支持资源 - 他们忙吗？如果他们不是，我们可能不得不改变流程以使用他们所有的时间并减少客户的等待时间；但是，“利用率”为 98%，这表明资源几乎没有空闲时间

## 假设“1有更多员工？比较 2 支持资源与 3 和 5 支持资源

1. 复制*TwoSupport*并按 **Ctrl+Shift+V** 进行粘贴，将新元素命名为*ThreeSupport*。
2. 双击*ThreeSupport*打开“配置 BPSim”对话框。
3. 在图中，单击“支持”资源元素。
4. 在配置 BPSim窗口中，在“资源”-“数量”的“值”字段中，键入“3”。
1. 再次复制*TwoSupport*并按 **Ctrl+Shift+V** 进行粘贴，将这个新元素命名为*FiveSupport*。
2. 双击*FiveSupport*以打开“配置 BPSim”对话框。
3. 在图中，单击“支持”资源元素。
4. 在配置 BPSim窗口中，在“资源”-“数量”的“值”字段中，键入“5”。

运行模拟并进行比较；在浏览器窗口中：

1. **Ctrl+单击***TwoSupport*、*ThreeSupport*和*FiveSupport*，然后右键单击并选择“显示 BPSim配置”选项。
2. **Ctrl+单击***TwoSupport-Result*、*ThreeSupport-Result*和*FiveSupport-Result*，然后右键单击并选择“显示 BPSim报告”选项。

The image contains two screenshots of the Enterprise Architect BPMN Simulation Report View. The top screenshot shows a comparison of resource quantities for 'FiveSupport', 'ThreeSupport', and 'TwoSupport'. The bottom screenshot shows a comparison of performance metrics like 'Average Time In Task', 'Degree Of Utilisation', and 'Sum Of Wait Time' for the same three support levels.

Item	FiveSupport	ThreeSupport	TwoSupport
Resource			
Support			
Quantity			
Default	5	3	2

Item	FiveSupport- Result	ThreeSupport- Result	TwoSupport- Result
Time			
Service Customer			
Average Time In Task	10.00	21.40	37.00
Average Time Waiting For Resource	0	11.40	27.00
Maximum Time In Task	10.00	34.00	64.00
Maximum Time Waiting For Resource	0	24.00	54.00
Total Time In Task	200.00	428.00	740.00
Total Time Waiting For Resource	0	228.00	540.00
Support			
Help Desk Phone Support Process			
Control			
Service Customer			
Resource			
Service Customer			
Support			
Degree Of Utilisation	83.00%	93.00%	98.00%
Sum Of Wait Time	40.00	16.00	4.00

提示：

- 单击 按钮和两个视图的“仅显示不同项”选项
- 您可以将视图停靠在一起，以便它们提供直接比较：这些是由配置中的这些差异引起的结果差异
- 切换过滤器栏以过滤您感兴趣的项目

分析


停靠的比较视图显示配置差异和相应的结果差异。

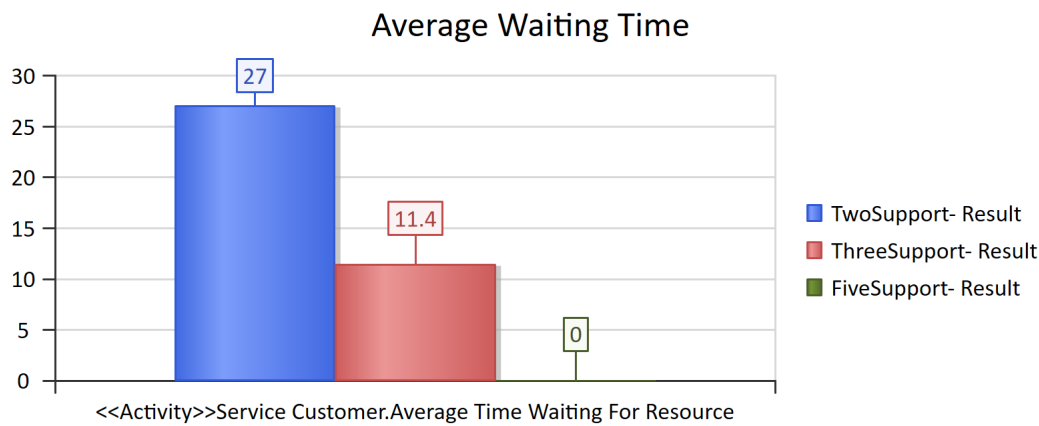
- 客户的等待时间从 27 分钟（2 个支持资源）下降到 11.4 分钟（3 个支持资源）并进一步下降到 0 分钟（5 个支持资源）
- 使用度”从 98%（2 个支持资源）下降到 93%（3 个支持资源）并进一步下降到 83%（5 个支持资源）


客户很可能对 5 个支持资源感到满意；但是，成本可能超出预算。因此，3 个或可能 4 个支持资源可能是这种情况下的平衡点。尝试复制业务流程仿真工件和模拟配置和运行支持资源之一。

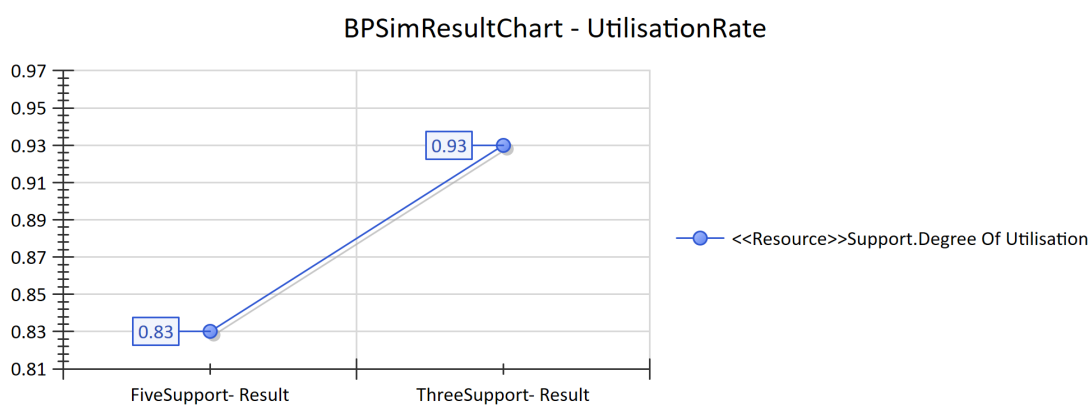
用图表显示结果

1. 将工具箱中的图表'Result Figure' 图标拖到工具箱中，然后在工具上创建图表'Result'工件；称之为平均等待时间。

2. 右键单击工件并选择 属性"选项以元素 属性"对话框；点击 图表图表"页面。
3. 单击  按钮并选择一个基本报告，从中定义要在图表中使用的架构（图例）；选择*TwoSupport-结果*。
4. 选择模式 时间" |'服务顾客'|'平均等待资源的时间'。
5. 单击添加按钮添加另外两个 BPSim 报告：*ThreeSupport-Result*和*FiveSupport-Result*
6. 点击确定按钮，调整图表元素的大小；这个图表给了我们非常直接的信息



1. 在工件上创建另一个图表结果显示，称为*Utilization Rate*。
2. 双击属性以显示元素对话框，然后单击 工件图表"选项卡。
3. 单击  按钮并选择一个 Base Report，从中定义要在图表中使用的模式（图例）；选择*TwoSupport-结果*。
4. 选择架构 资源" |'支持'| 利用程度"。
5. 单击添加按钮添加另外两个 BPSim 报告：*ThreeSupport-Result*和*FiveSupport-Result*。
6. 单击确定按钮并调整图表元素的大小。



# 基于日历的服务台电话支持仿真

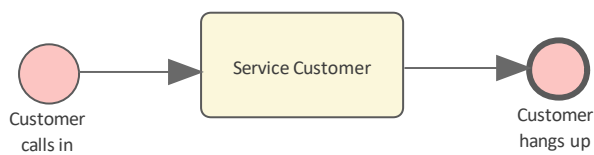
在这个例子中，我们创建了一个非常简单的模型来模拟服务台电话支持过程，基于日历设置。我们假设：

- 客户在工作日和周末以不同的时间间隔致电
- 工作日和周末的处理时间不同
- 工作日和周末有不同数量的支持资源

我们模型一步地模拟这个过程，然后创建日历并配置业务流程模拟，简单到可以用纸笔计算。之后，我们运行模拟以将该结果与笔和纸分析进行比较。

## 创建 BPMN 模型

模型本身很简单，由一个开始事件、一个任务和一个结束事件组成。

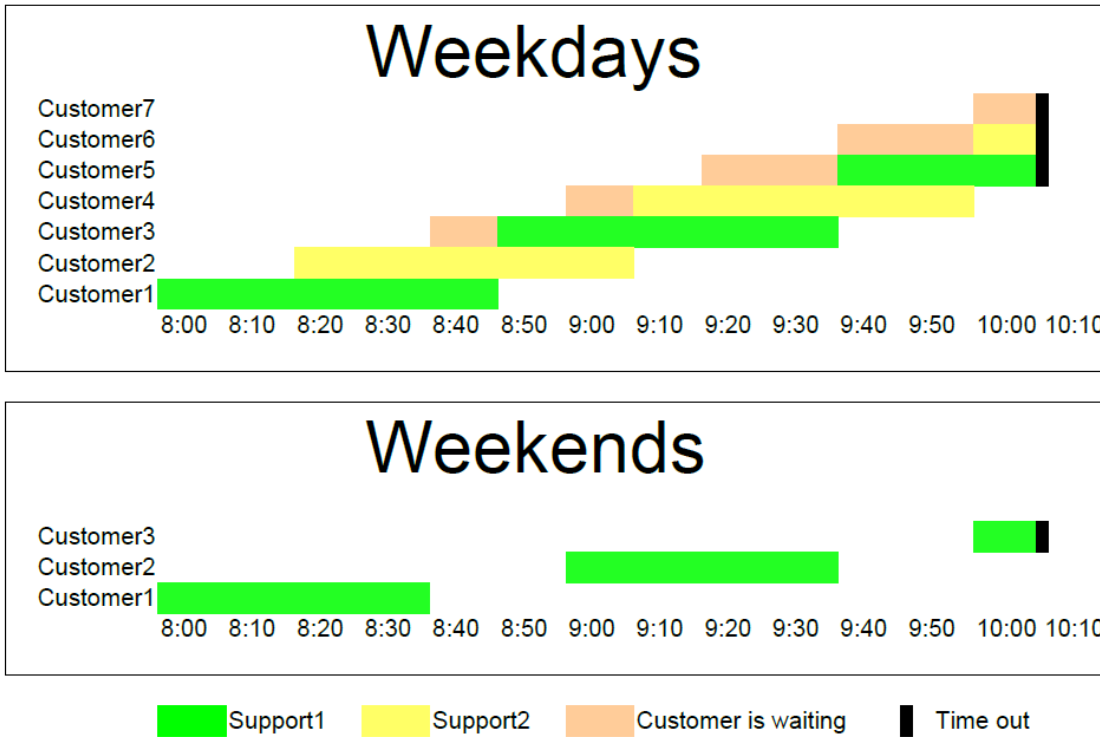


1. 创建一个开始事件顾客呼入。
2. 添加一个序列目标抽象任务活动服务顾客。
3. 将一个序列流添加到目标结束事件的顾客挂断。
4. 创建一个名为*Support*的 *BPMN2.0* 资源。
5. 在*Service*顾客中创建一个元素::*ResourceRole*，将其命名为*support*并将标签*resourceRef*设置为 *Resource ElementSupport*的名称。

## 笔和纸分析

我们可以用纸笔来分析这个案例：

- 模拟时间为 2 小时 10 分钟，从上午 8:00 到上午 10:10
- 工作日每 20 分钟就有A顾客打来电话
- 周末每 60 分钟就有A顾客打来电话
- 平日为每位客户服务需要 50 分钟
- 周末为每位顾客服务需要40分钟
- 平日有2个支持资源
- 周末有1支持资源



从这个结果来看，当应用资源限制时，对于这样一个简单的模型，计算是相当复杂的。

**在工作日**

- 7 位客户在 2 小时 10 分钟内每隔 20 分钟打一次电话
- 4 个客户电话正常挂断
- 2 客户通话因超时而中断
- 1 客户电话无人接听
- Support1 连续工作 130 分钟，Support2 连续工作 110 分钟

**在周末**


- 3 位客户在 2 小时 10 分钟内每隔 60 分钟致电一次
- 2 个客户电话正常挂断
- 1 客户通话因超时而中断
- Support1 工作了 90 分钟，每 40 分钟一次，通话间隔 20 分钟

现在我们将了解 BPSim 如何提供帮助。


**配置**

在本节中，我们首先创建日历，然后设置持续时间和开始参数。

对于元素参数，您可以为给定参数指定一个或多个日历。但是，如果为参数值设置了任何日历，则必须存在默认值（未指定任何日历），否则模拟将无法工作。

单击配置 BPSim 窗口工具栏上的  按钮将自动为您检查此约束。

任务	行动
创建工作件并设置包	1. 打开配置 BPSim 窗口（'仿真>过程分析>进程>打开BPSim 管理器'）。


	<ol style="list-style-type: none"> <li>2. 创建一个业务流程仿真辅助工件，命名为基于进程的流程仿真。</li> <li>3. 选择包含对应BPMN 2.0模型的包。</li> <li>4. 打开包含要模拟的模型的图表。</li> </ol>
日历	<ol style="list-style-type: none"> <li>1. 在配置 BPSim窗口的 配置”选项卡上，单击工具栏中的  图标。将显示编辑 BPSim 日历”对话框。</li> <li>2. 单击 新建”按钮以显示 事件重复”对话框并按照此处所述完成字段，以创建日历。（您将创建两个日历。）</li> <li>3. 在 事件时间”面板中，将 开始”设置为上午 08:00，将 结束”设置为下午 5:00。</li> <li>4. 在 重复模式”面板中选择 每周”并选中从 星期一”到 星期五”的复选框。</li> <li>5. 在 重复范围”面板中，将 开始”设置为 2020 年 2 月 11 日”，然后选择 无结束日期”。</li> <li>6. 点击确定按钮。系统会提示您输入日历名称；用 工作日”改写Calendar_1，然后单击确定按钮。</li> <li>7. 再次单击新建按钮并使用这些值重复步骤 3 到 6： <ul style="list-style-type: none"> <li>- '开始' - 08:00AM</li> <li>- '结束' - 05:00PM</li> <li>- 每周”</li> <li>- '周六和周日'</li> <li>- 开始”到 2020 年 7 月 11 日”和 无结束日期”</li> <li>- 用 周末”改写Calendar_2</li> </ul> </li> <li>8. 点击确定按钮。</li> </ol>
期间	<p>在图表上，单击 基于仿真进程流程配置 BPSim工件窗口的 配置”选项卡上，将 日历”字段设置为 “---无---”，创建或编辑此场景参数：</p> <ul style="list-style-type: none"> <li>● 持续时间 - 常量值为 0000 002:10:00，表示 0 天 2 小时 10 分钟</li> </ul>
顾客到来	<p>在图表上，单击 StartEvent 中的顾客调用，然后在配置 BPSim配置的 配置”选项卡上，创建或编辑控件参数：</p> <ul style="list-style-type: none"> <li>● InterTriggerTimer - 值：0 00:00:00，日历”字段设置为 “---None---”（此默认值是必需的）</li> <li>● InterTriggerTimer - 值：0 00:20:00，日历”字段设置为 工作日”</li> <li>● InterTriggerTimer - 值：0 01:00:00，日历”字段设置为 周末”</li> </ul>
处理时间	<p>在图表上，单击服务顾客活动，然后在配置 BPSim配置的 配置”选项卡上，创建或编辑此时间参数：</p> <ul style="list-style-type: none"> <li>● ProcessingTime - 值：0 00:00:00，日历”字段设置为 “---None---”（此默认值是必需的）</li> <li>● ProcessingTime - 值：0 00:50:00，日历”字段设置为 工作日”</li> <li>● ProcessingTime - 值：0 00:40:00，日历”字段设置为 周末”</li> </ul>
资源	<p>在图表上，单击支持资源，然后在配置 BPSim配置的 配置”选项卡上，创建或编辑此资源参数</p> <ul style="list-style-type: none"> <li>● 数量 - 值：0；日历，日历”字段设置为 “---无---”（此默认值是必需的）</li> <li>● 数量 - 价值：2；日历，日历”字段设置为 工作日”</li> <li>● 数量 - 价值：1；日历，日历”字段设置为 周末”</li> </ul>
资源选择（分配）	<p>在图表上，单击服务顾客活动，然后在配置 BPSim配置的 配置”选项卡上，将日历字段设置为 “---None---”，检查 值”字段是否为资源参数 选择”设置为：</p>

**bpsim::getResource('Support', 1)**作为表达式

该表达式默认从您的 BPMN模型中加载。您可以为任务的资源选择做一些高级配置。


## 运行仿真

### 平日

1. 单击 日历"字段并选择 工作日"。
2. 单击 执行"选项卡和  工具栏图标。

生成名为 *Calendar Based Support* 仿真进程的文件。此报告文件包含工作日模拟的结果，该结果显示在 标准结果报告"选项卡上的 配置 BPSim审阅窗口的 审阅"选项卡上。

### 周末

1. 单击 日历"字段并选择 周末"。
2. 单击 执行"选项卡和  工具栏图标。

基于日历的流程仿真支持 - 进程文件已更新以显示周末模拟的结果，并显示在 标准结果报告"选项卡上配置 BPSim审阅的 审阅"选项卡上。

在每种情况下，用笔和纸检查结果文件和我们的分析之间的匹配。



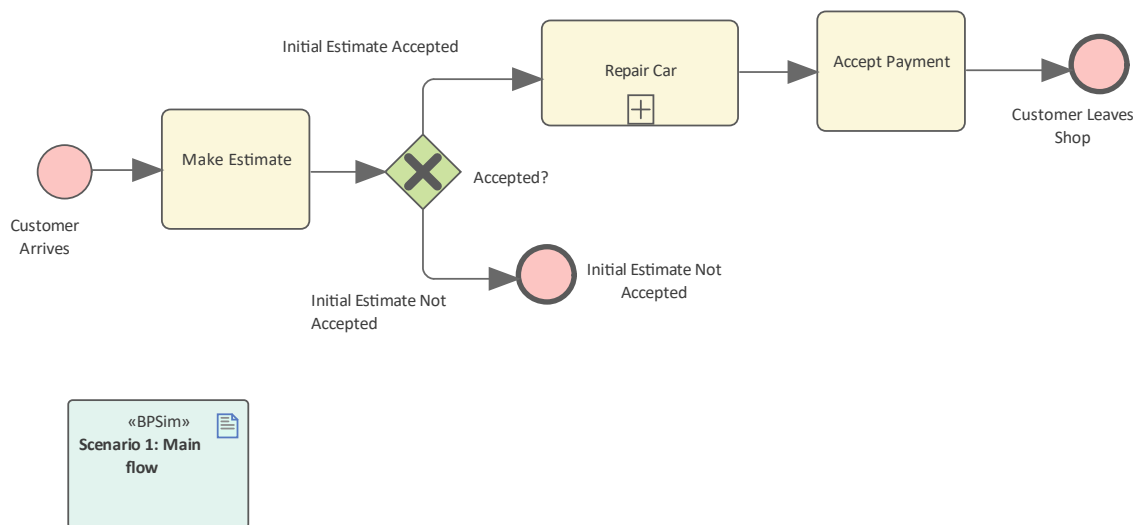
# 汽车维修进程

此示例模拟汽车维修店的流程。BPSim 配置：

- 使用由分布初始化的属性参数为每个客户生成随机数量的问题
- 应用概率来模拟：
  - 是否接受初步估计
  - 修复过程中是否会发现新问题
- 在每个任务中增加或减少属性参数的值
- 将属性参数的值用于从网关传出的序列的条件
- 模拟给定开始和持续时间的客户到达

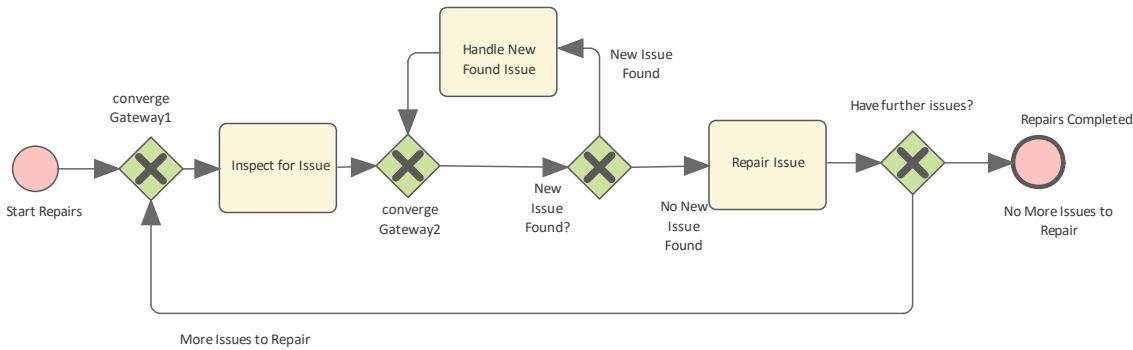
## 创建 BPMN 模型

### 创建主进程




1. 创建一个开始事件顾客到达。
2. 将序列流添加到目标抽象任务活动进行估计。
3. 将序列流添加到目标 Exclusive Gateway *Accepted?*。
4. 将序列流添加到：
  - A 未接受目标结束事件初始估计
  - A 目标子流程修理车
5. 从 *Repair Car* 中，将序列流添加到目标抽象任务活动接受付款。
6. 将序列流添加到目标结束事件顾客离开商店。






### 创建子流程修理汽车





1. 创建一个开始事件开始。
2. 将序列流添加到目标 Exclusive Gateway会聚 Gateway1。
3. 将一个序列添加到一个抽象任务活动中检查问题。
4. 将序列流添加到独占网关会聚网关 2。
5. 将序列添加到独占网关New问题?。
6. 将序列流添加到：
  - A目标抽象任务活动处理新发现的问题，然后添加一个流回汇聚序列
  - A目标抽象任务修复问题，然后添加一个序列一个 target Exclusive Gateway还有其他问题？
7. 从网关还有问题吗？将序列流添加到：
  - 目标结束事件修复完成
  - 汇聚网关1


### 配置 BPSim

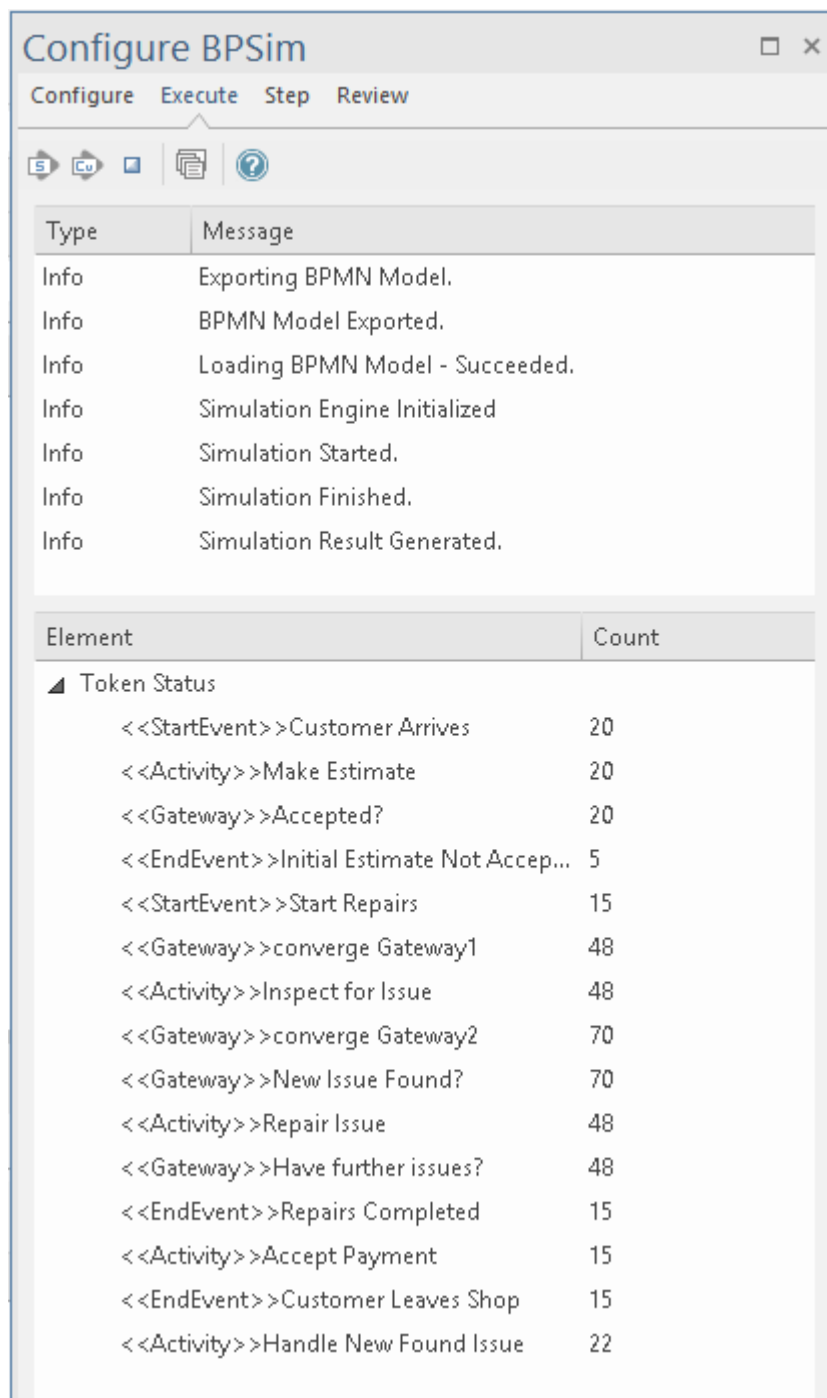
任务	描述
工件和包	<ol style="list-style-type: none"> <li>1. 打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )。</li> <li>2. 创建一个业务流程仿真工件的场景1：主流程。</li> <li>3. 选择包含对应BPMN 2.0模型的包。</li> </ol>
开始和持续时间	<p>我们将模拟一家汽车维修店的流程，该店的营业时间为上午 9:00 至下午 5:00，即 8 小时。我们还假设下午 4:50 之后走进的顾客当天不会得到服务。因此模拟开始时间为上午 9:00，持续时间为 7 小时 50 分钟。</p> <p>在“业务流程仿真”图上，单击“汽车维修名称”，在配置 BPSim窗口中，单击“主1”和“汽车维修名称”，更新这些工件：</p> <ul style="list-style-type: none"> <li>• 开始- 用任何日期（格式为 dd/mm/yyyy）改写“值”字段并将时间部分更改为“上午 9:00”</li> <li>• 持续时间 - 单击“值”字段中的  按钮并将其设置为“07:50:00”的恒定持续时间</li> </ul>
顾客来了	<p>我们将模拟每 24 分钟到达的客户。</p> <p>第一个客户在上午 9:00 到达，最后一个在下午 4:36 到达（下午 5:00 到达的客户今天将不会得到服务，因为这受“持续时间”设置的限制）。</p> <p>用笔和纸，我们可以计算出服务了 20 位客户（上午 9:00 到下午 4:36 = 456 分钟；客户数量为 <math>456/24 + 1 = 19 + 1 = 20</math>）。稍后我们将通过仿真结果验证这</p>

	<p>一点。</p> <p>在 “Car Repair”图表上，单击开始事件元素顾客，然后在配置 BPSim开始中：</p> <ol style="list-style-type: none"> <li>1. 单击新参数下拉箭头，然后选择 “控件”。</li> <li>2. 单击 “参数”下拉箭头并选择 “InterTriggerTimer”。</li> <li>3. 在 “值”字段中单击  按钮并设置常数数值 “24 分钟”。单击确定按钮和保存工具栏图标。</li> </ol>
<p>属性Parameters</p>	<p>我们假设每个客户的汽车最初可能有不同数量的问题。这可以使用随机数生成器来反映。BPSim 提供了许多发行版以满足您的需求。</p> <p>在此示例中，我们使用截断正态分布来初始化属性。任务修复问题和处理新发现问题将分别递减和递增noOfIssues的值。</p> <ol style="list-style-type: none"> <li>1. 在 “汽车维修”图表上，单击开始事件顾客到达。</li> <li>2. 在配置 BPSim配置的 “配置”选项卡上，单击新参数下拉箭头并创建一个名为noOfIssues的属性参数。</li> <li>3. 在 “值”字段中单击  按钮；显示 “CustomerArrives”对话框的 “配置”配置。</li> <li>4. 单击 “分布”选项卡并选择 “截断正常”；在田野里：             <ul style="list-style-type: none"> <li>- '平均值'，输入'2'</li> <li>- '标准偏差'，输入'1'</li> <li>- 'Min'，输入'1'</li> <li>- “最大”，输入'1000'”</li> </ul> </li> </ol> <p><b>重要注记：</b>诸如'TruncatedNormal'之类的分布，返回一个浮点值，但该属性用作整数。设置属性的类型很重要，尤其是在进行相等测试时的条件表达式中。例如，条件表达式<code>getProperty('noOfIssues') = 0</code>几乎永远不会满足，因为noOfIssues是由浮点分布初始化的。</p> <p><b>提示：如何自定义属性的类型</b></p> <p>创建属性并设置值后，单击工具栏上的  图标，然后单击  图标以显示 “属性参数”对话框。在属性的 “类型”字段中，单击下拉箭头并选择值 “int”而不是默认的 “double”。</p> <ol style="list-style-type: none"> <li>1. 在 “维修汽车”图表上，单击活动维修问题。</li> <li>2. 在配置 BPSim配置的 “配置”选项卡上，单击新参数下拉箭头并创建一个名为noOfIssues的属性参数。</li> <li>3. 在 “值”字段中单击  按钮。显示 “修复问题”对话框的 “配置”配置。</li> <li>4. 单击 “表达式”选项卡，然后在 “表达式”字段中输入<code>{noOfIssues} - 1</code>；点击确定按钮。</li> <li>5. 在 “修理汽车”图表上，单击活动处理新发现问题。</li> <li>6. 在配置 BPSim配置的 “配置”选项卡上，单击新参数下拉箭头并创建一个名为noOfIssues的属性参数。</li> <li>7. 在 “值”字段中单击  按钮。显示 “处理新发现的问题”对话框的 “配置”配置。</li> <li>8. 单击 “表达式”选项卡，然后在 “表达式”字段中输入<code>{noOfIssues} + 1</code>；点击确定按钮。</li> </ol>
<p>概率上的序列流</p>	<p>我们估计三分之一的客户不会接受最初的维修估价，剩下的两个会接受。我们还估计，四分之一的维修会发现新问题，其余三个维修不会发现新问题。</p> <p>在 “汽车维修”图上，参考网关元素接受了吗？.点击：</p> <ul style="list-style-type: none"> <li>• 初始接受序列流和配置 BPSim窗口中单击参数下拉箭头，并创建一个名</li> </ul>

	<p>为 概率参数”的控件参数；在 值”字段中输入 0.67”</p> <ul style="list-style-type: none"> <li>• <i>Initial Not Accepted</i>序列流，在配置 BPSim中点击参数-down箭头，并创建一个控件参数，名为'参数概率'；在 值”字段中输入 0.33”</li> </ul> <p>在 汽车维修”图上，请参阅网关元素新问题？.点击：</p> <ul style="list-style-type: none"> <li>• <i>No More to Repair SequenceFlow</i> 和配置 BPSim序列中单击参数下拉箭头，并创建一个控件 参数参数”的概率；在 值”字段中输入 0.75”</li> <li>• 修复序列流的更多问题，并在配置 BPSim窗口中单击参数下拉箭头，并创建一个名为 参数概率”的控件参数；在 值”字段中输入 0.25”</li> </ul>
<p>条件序列条件</p>	<p>我们使用表达式返回一个布尔值作为序列流的条件，这在流的逻辑中起着关键作用。</p> <p>在 维修汽车”图表上，请参阅还有其他问题？网关元素。点击：</p> <ul style="list-style-type: none"> <li>• 修复序列流的更多问题，并在配置 BPSim窗口中单击新参数下拉箭头，并创建一个名为 条件”的控件参数；在 值”字段中单击  按钮，单击 表达式 选项卡并在 表达式”字段中键入 <b>{noOfIssues} != 0</b></li> <li>• <i>No More to Repair SequenceFlow</i> 并在配置 BPSim序列单击参数向下箭头，并创建一个名为 参数条件”的控件参数；在 值”字段中单击  按钮，单击 表达式”选项卡并在 表达式”字段中键入 <b>{noOfIssues} = 0</b></li> </ul> <p>注记：网关的所有传出转换都应包含 100% 的逻辑；例如，您不会输入 <b>{noOfIssues} &gt; 10</b>和<b>{noOfIssues} &lt; 5</b>作为条件表达式，因为<b>[5, 10]</b>范围内的值不会被任何传出序列流处理。</p>

### 运行仿真


1. 在配置 BPSim窗口中，单击 执行”选项卡和工具栏中的  图标。
2. 当模拟完成时，执行选项卡提供类似于以下的结果：



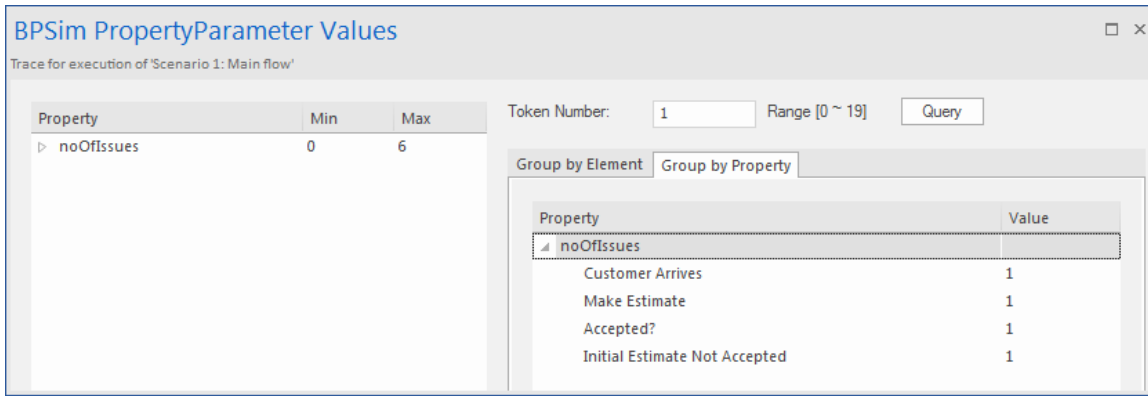
### 代币分析

- 20 位客户到达，与我们手动计算的数量相匹配（请参阅配置 *BPSim* 顾客表的顾客到达）
- 20 位客户中有 8 位不接受初步估价，而 20 位客户中有 12 位接受并修理了他们的汽车；这些数字大致符合  $1/3$  和  $2/3$  的概率
- 64 个令牌通过了网关新问题 *Found?*，其中 19 个有新问题，45 个没有；这些数字大致符合  $1/4$  和  $3/4$  的概率

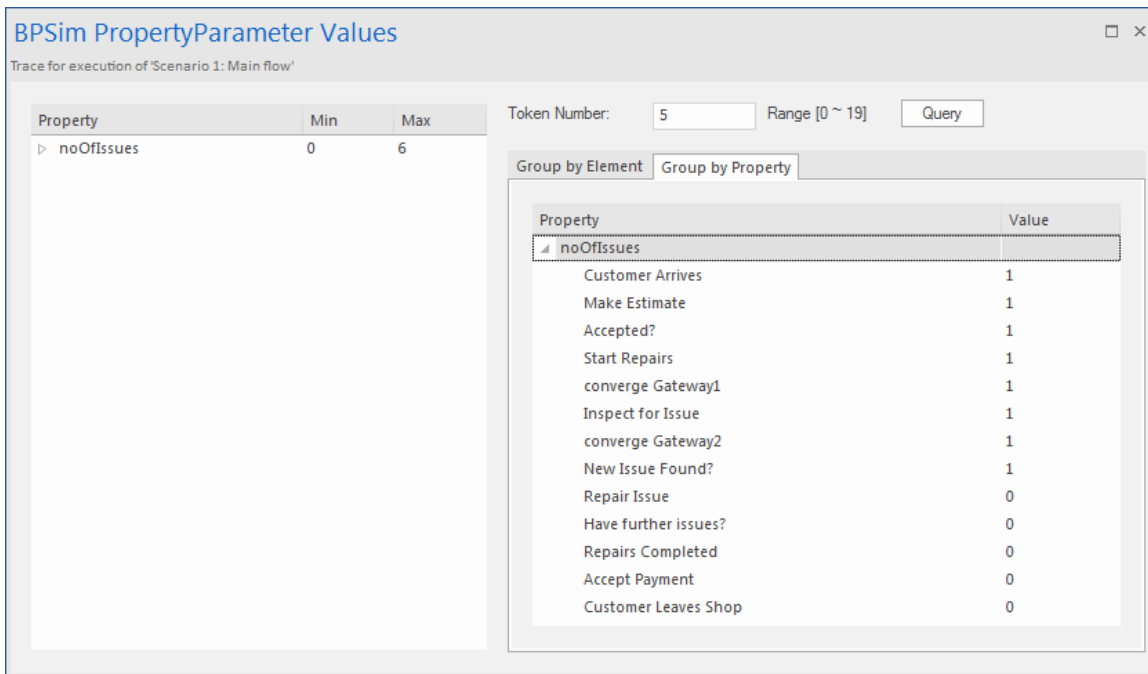
### 个人客户分析

单击工具栏上的  按钮以打开 *BPSim PropertyParameter Values*”对话框。由 有 20 个客户（令牌），您可以在 *令牌编号*”字段中输入一个介于 0 和 19 之间的值，然后单击查询按钮进行一些分析：

- 该客户不接受初始估算，如 *按属性分组*”选项卡上所示：



- 该客户的汽车只有一个问题，已修复：



- 该客户的汽车有三个已知问题，在维修过程中发现了三个其他问题，因此总共修复了六个问题（可能这是一辆非常旧的汽车）；切换到“按元素分组”选项卡：

### BPSim PropertyParameter Values

Trace for execution of 'Scenario 1: Main flow'

Token Number:  Range [0 ~ 19]

Property	Min	Max
noOfIssues	0	6

Group by Element | Group by Property

Element	Value
Customer Arrives	
noOfIssues	3
Make Estimate	
Accepted?	
Start Repairs	
converge Gateway1	
Inspect for Issue	
converge Gateway2	
New Issue Found?	
noOfIssues	3
noOfIssues	4
noOfIssues	5
noOfIssues	6
noOfIssues	5
noOfIssues	4
noOfIssues	3
noOfIssues	2
noOfIssues	3
noOfIssues	4
noOfIssues	5
noOfIssues	4
noOfIssues	3
noOfIssues	2
noOfIssues	1

## BPMN2.0事件示例

事件是在进程的过程中发生的事情。事件影响进程的流程，通常有原因或影响，一般要求或允许反应。例如，一个活动的开始，一个活动的结束，一个文档状态的变化，或者一个信息的到来，都可以被认为是事件。

事件允许描述“事件驱动”进程。在这些进程中，有三种主要类型的事件：

- 开始事件指示进程将从哪里开始
- 事件，表示一个进程的路径将在哪里结束
- 中间事件，指示在进程的开始和结束之间发生某事的位置

在这三种类型中，事件可以是两种子类型之一：

- 捕获触发器的事件- 所有开始事件和一些中间事件都在捕获事件
- 抛出 Result 的事件- 所有事件和一些中间事件都在抛出最终可能被另一个事件捕获的事件

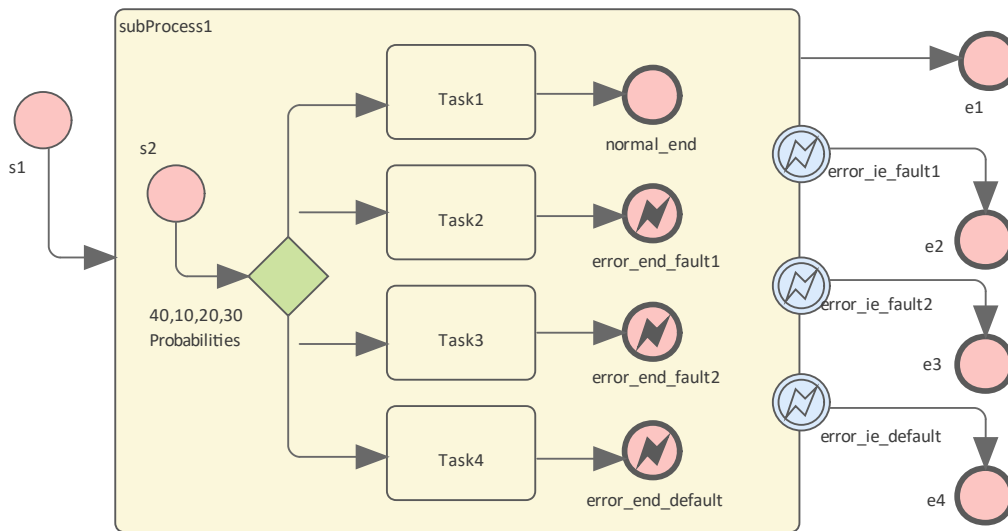
在本节中，我们提供的示例说明了许多常用的 BPMN 2.0 事件。在每个示例中，我们都提供了逐步的 BPMN 建模和 BPSim 配置说明，并对仿真结果进行了深入分析。EAExample模型中提供了所有示例。



# 错误事件

当中间错误事件连接到活动的边界时，它就成为异常流的一部分。当令牌导致在正常流程中引发故障名称时触发该事件，并进入错误结束事件。

## 创建 BPMN 模型



### 创建主进程

- 创建一个开始事件 *s1*
- 添加一个序列到一个目标活动元素 *subProcess1*；放大活动并右键单击，选择是展开的“选项，然后打开属性”对话框并将类型”设置为“子进程”
- 将序列流添加到目标结束事件元素 *e1*（'类型'设置为'无'）
- 创建三个中间事件，将元素从工具箱边界；从即时菜单中选择“边缘安装”和“错误”：
  - *error\_ie\_fault1*；将序列流添加到目标 EndEvent 元素 *e2*（类型”设置为“无”）
  - *error\_ie\_fault2*；将序列流添加到目标 EndEvent 元素 *e3*（'类型'设置为'无'）
  - *error\_ie\_default*；将序列流添加到目标 EndEvent elementr *e4*（'类型'设置为'无'）

### 创建子流程

在 *subProcess1* 活动中：

- 创建一个开始事件 *s2*，'Standalone' 并将 '类型' 设置为 'None'
- 为目标网关元素创建一个序列，设置为“独占”，名称为“40,10,20,30 Probabilities”
- 对类型为'abstractTask'的四个目标Activiy元素创建序列流，称为：
  - *Task1*，并将序列流添加到名为 *normal\_end* 的目标 EndEvent，类型”设置为“无”
  - *Task2*，并将序列流添加到名为 *error\_end\_fault1* 的目标 EndEvent 中，“类型”设置为“错误”
  - *Task3*，并将序列流添加到名为 *error\_end\_fault2* 的目标 EndEvent 中，类型”设置为“错误”
  - *Task4*，并将序列流添加到名为 *error\_end\_default* 的目标 EndEvent 中，类型”设置为“错误”


### 创建 BPMN2.0::Error 元素

创建错误元素 *Fault1* 和 *Fault2*，它们将被事件用作错误代码。

- 双击 *error\_end\_fault1* 元素，在属性”对话框的 BPMN2.0”选项卡中，找到“errorRef”标签
- 在“值”字段中，单击  按钮并浏览到包含此模型的包

- 单击“加新”按钮，在“名称”字段中输入名称*Fault1*，然后单击“保存”按钮
- 再次单击加新按钮，在“名称”字段中输入名称*Fault2*，然后单击保存按钮
- 单击确定按钮，然后再次单击下一个确定按钮


**为错误代码设置事件**

- 双击*error\_end\_fault1*元素，在“属性”对话框的“BPMN2.0”选项卡中，找到“errorRef”标签
- 在“值”字段中，单击按钮并浏览到包含此模型的包
- 单击*Fault1*，然后单击确定按钮，然后再次单击确定按钮。

对这些元素执行相同的操作：

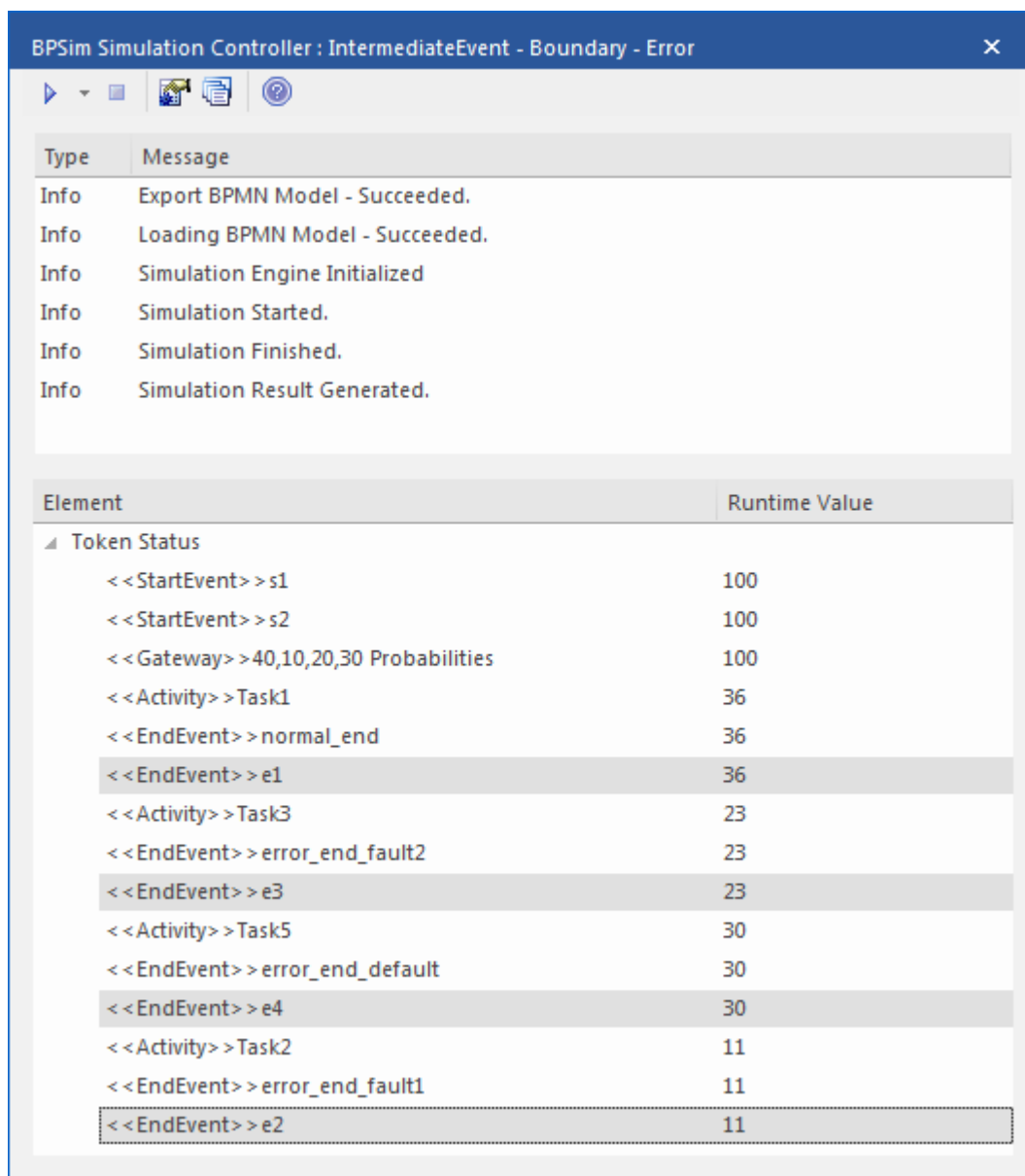
- *error\_end\_fault2* · 点击*Fault2*
- *error\_ie\_fault1* · 点击*Fault1*
- *error\_ie\_fault2* · 点击*Fault2*

## 配置 BPSim

物件	行动
工件&包	<ul style="list-style-type: none"> <li>• 打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )</li> <li>• 创建它的工件名称，单击“边界-资源包”中的“Intermediate - 资源包” ( 在“事件/创建工作件包”字段中选择“父元素包”，然后单击“加新按钮”，然后单击“新元素包”保存按钮和确定按钮 )</li> </ul> <p>然后所有的 BPMN 元素将被加载到配置 BPSim窗口中。</p>
s1	<ul style="list-style-type: none"> <li>• 从配置 BPSim窗口左侧的树中，展开“StartEvent”并单击“s1”</li> <li>• 在“控件”选项卡的“新建参数...”字段中，单击下拉箭头并选择“触发器计数”</li> <li>• 在“值”字段中，输入“100”</li> </ul>
概率	<p>从配置 BPSim窗口左侧的树中，展开“网关   40,10,20,30 概率”。</p> <p>提示：您也可以浮动配置 BPSim窗口，然后点击BPMN图上的元素或连接器；配置 BPSim窗口中的元素将被自动选中。</p> <p>对于每个任务元素，在“控件”选项卡中单击“新建参数”下拉箭头并选择“概率”，然后在“值”字段中键入相应的值：</p> <ul style="list-style-type: none"> <li>• 对于 Task1 类型“0.4”</li> <li>• 对于任务 2，键入“0.1”</li> <li>• 对于 Task3 类型“0.2”</li> <li>• 对于 Task4 类型“0.3”</li> </ul>

## 运行仿真

- 在“配置 BPSim”对话框工具 上，单击“运行”图标打开“运行控制器”对话框
- 点击运行按钮，选择“标准仿真”
- 模拟结果如下：



分析：

从概率流出的序列上的序列集的概率分别为1和0.3。

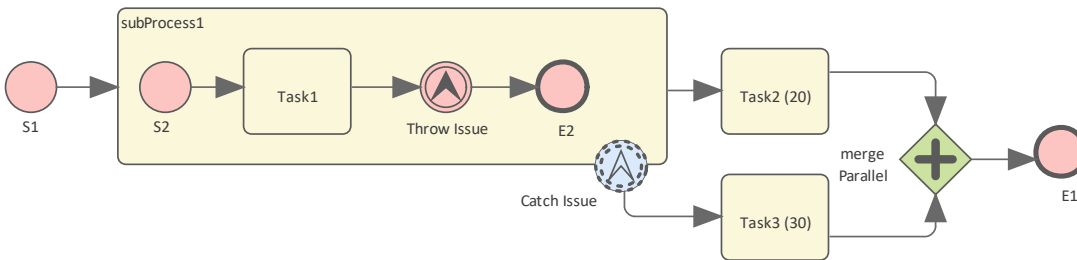
- 100 次传球中有 36 次在normal\_end完成，流向e1
- 100 次通过中有 11 次在error\_end\_fault1完成，由 ErrorRef Fault1 触发error\_ie\_fault1，异常流向e2
- 100 次中的 23 次在error\_end\_fault2完成，由 ErrorRef Fault2触发error\_ie\_fault2，异常流向e3
- 100 次传递中有 30 次在error\_end\_default完成，这触发了error\_ie\_default因为它们没有设置 ErrorRef 并且异常流向e4

36、11、23、30这三个数字加起来就是100，在s1中设置为TriggerCount，所以它们匹配100%的概率

# 升级事件

在 BPMN 中，Escalation 是 Error 的非中断对应物，具有类似的 throw-catch 行为。然而，与 Error 不同的是，从活动中退出的正常流程和异常流程是并行路径，而不是替代路径。

## 创建 BPMN模型



### 创建主进程

- 创建一个开始事件S1
- 将序列流添加到目标活动subProcess1；放大活动并右键单击，选择 是展开的”选项，然后打开 属性”对话框并将 类型”设置为 子进程”
- 将一个序列流添加到目标 abstractTask活动元素Task2 (20)（打开 属性”对话框并将 类型”字段设置为 抽象任务”）
- 将一个序列添加到目标并行网关元素合并并行（打开 属性”对话框并将 类型”字段设置为 并行”）
- 将序列流添加到目标事件
- 在subProcess1上，添加边界非中断 EscalationEventCatch问题（将 事件”图标拖到事件上，并从即时菜单中选择 Edge Mounted”和 Escalation”；双击元素以显示 属性”对话框并添加名称，然后在类型”字段中选择边界非中断>升级’）
- 将序列流添加到目标 abstractTask活动元素Task3 (30)（打开 属性”对话框并将 类型”字段设置为 抽象任务”）
- 添加一个序列到目标元素合并并行



### 创建子流程

- 在subProcess1内（或下），创建一个开始事件S2
- 将序列流添加到目标 abstractTask活动元素Task1（打开 属性”对话框并将 类型”字段设置为 抽象任务”）
- 将序列流添加到目标事件Escalation IntermediateEventThrow问题（打开 属性”对话框并在 类型”字段中选择抛出>升级”）
- 将序列流添加到目标结束事件E2

### 创建 BPMN2.0::Escalation 元素



在工具箱中，展开 图表2.0 Types”页面，将 Escalation”图标拖到图表上，并将元素命名为Escalation1；这将被事件用作升级代码。

### 为升级代码设置事件：


- 双击Throw问题并在 escalationRef 标记的 值”字段中单击  图标并找到并选择Escalation1
- 双击Catch问题并再次在 escalationRef 标签的 值”字段中单击  图标并找到并选择Escalation1

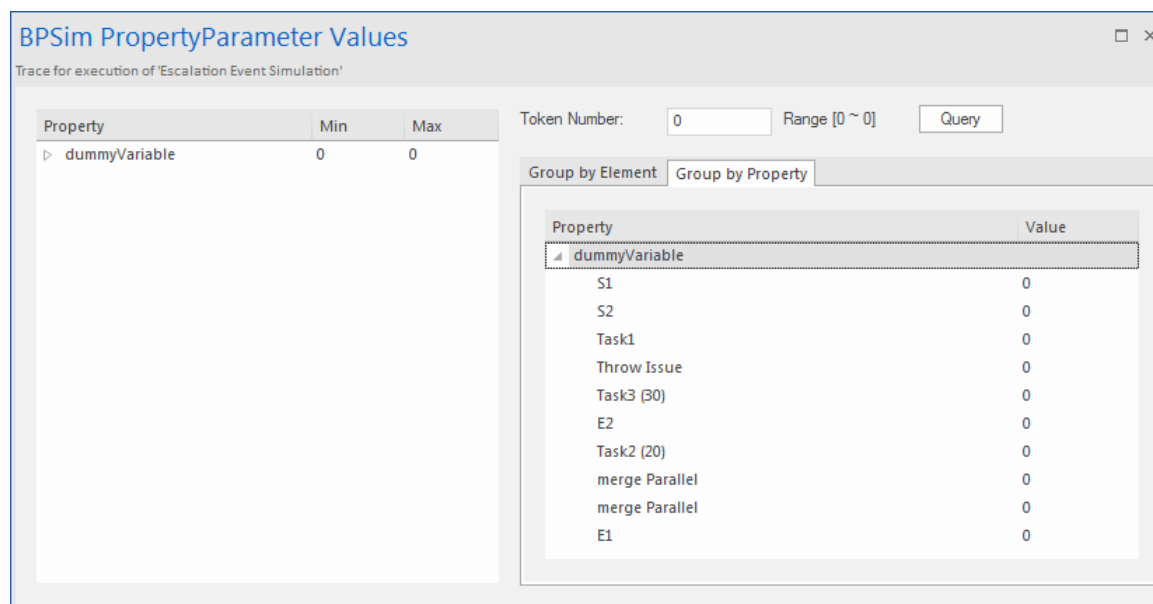
( 活动退出的异常流程是并行的。 )

## 配置 BPSim

任务	行动
工件&包	<ul style="list-style-type: none"> <li>打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )</li> <li>创建加新事件添加工件名称的 仿真事件" ( 在 选择/创建工件包名称"字段中，单击 选择/创建父元素名称"并  按钮，然后键入按钮，然后键入 保存"和确定按钮)</li> </ul> <p>然后所有的 BPMN 元素将被加载到配置 BPSim窗口中。</p>
触发器的开始事件计数	<ul style="list-style-type: none"> <li>从配置 BPSim窗口左侧的树中，展开 "StartEvent"并单击SI</li> <li>在 控件"选项卡的 新建参数..."字段中，单击下拉箭头并选择 触发器计数"</li> <li>在 值"字段中，输入 "1"</li> </ul>
处理时间	<ul style="list-style-type: none"> <li>在左侧树中展开 活动"并单击Task2 (20) ; 在 处理时间"的 值"字段中输入 "20"，在 单位"字段中输入 s" ( 20 秒)</li> <li>单击Task3 ( 30 ) ; 同样，将 'ProcessingTime' 设置为 30 秒</li> </ul>
dummyVariable 用于跟踪	<p>为了显示给定标记的准确轨迹，您必须在SI上设置一个虚拟变量。</p> <ul style="list-style-type: none"> <li>在左侧层次结构中单击SI，然后在 属性"选项卡上用变量名称覆盖新属性文本 ( 例如 属性" )</li> <li>在 值"字段中，单击  按钮，然后在 "&lt;&lt;StartEvent&gt;&gt;S1 : &lt;变量名称&gt;"对话框中单击 数字"并键入 常量数字"值 0" ; 点击确定按钮</li> </ul>

## 运行仿真

- 在 配置 BPSim "对话框工具 上，单击 运行"图标打开 运行仿真控制器"对话框
- 单击 运行"图标下拉箭头并选择 标准仿真"
- 仿真完成后，单击工具栏上的  按钮，显示 BPSim PropertyParameter Values"对话框
- 单击 查询属性"按钮和 按属性分组"选项卡，然后展开 dummyVariable" ( 或您分配给变量的名称 )



Property	Min	Max
dummyVariable	0	0

Token Number: 0 Range [0 ~ 0] Query

Group by Element Group by Property

Property	Value
dummyVariable	
S1	0
S2	0
Task1	0
Throw Issue	0
Task3 (30)	0
E2	0
Task2 (20)	0
merge Parallel	0
merge Parallel	0
E1	0

### 分析：

与Error不同，从subProcess1退出的正常流程和异常流程不是替代路径，而是并行的。这个特征很容易从trace中发现：

- 在Task3 (30)开始后，E2和Task2 (20)仍然被遍历
- 遍历mergeParallel两次后到达E1

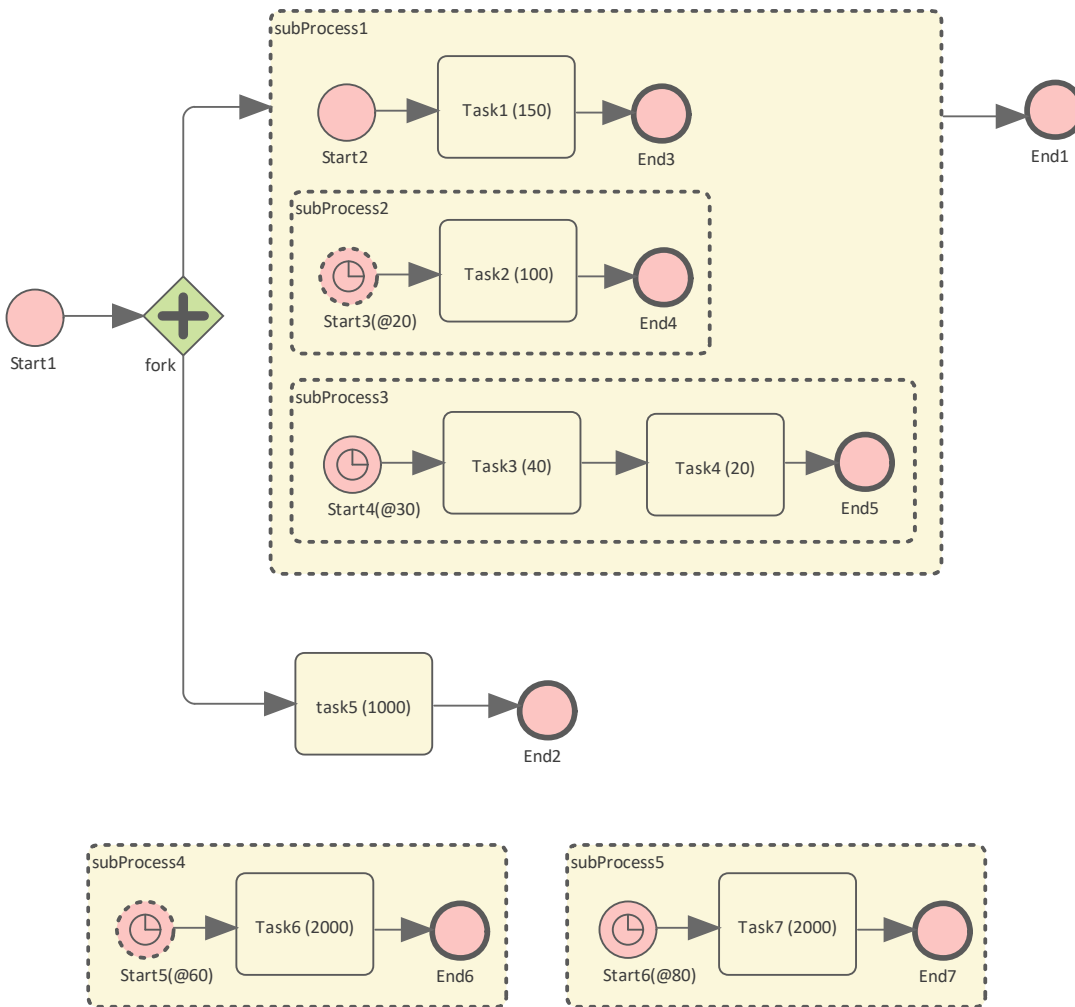
# 事件子流程

事件子流程使您的系统能够在给上下文流程或进程的时间内处理事件。一个事件子流程总是以一个开始事件开始；它不是由正常的控制流实例化，而是仅在触发相关的开始事件时实例化。事件子流程是自包含的，不得连接到子流程中的其余序列流。

- 如果设置了其开始事件的 `isInterrupting` 属性，则事件子进程取消执行封闭子进程
- 如果未设置 `isInterrupting` 属性，则封闭子流程的执行与事件子流程并行继续

在这个例子中，我们演示了 `Interrupting` 和 `Non-事件Sub-Processes` 如何影响封闭的 `Sub-Process` 和进程的生命线。

## 创建 BPMN 模型



### 主要流程模型

- 创建一个 `StartEvent Start1`
- 将序列流添加到目标并行网关元素分叉
- 添加一个序列
  - 一个子流程 `subProcess1`，并从中添加一个序列流目标结束事件元素 `结束1`
  - 一个抽象任务 `Task5`，并从中添加一个序列流到目标结束事件元素 `End2`

关于如何模型事件子流程的提示

- 将 活动-业务流程“工具箱中的活动拖到图表上
- 双击活动以显示 属性”对话框，并在 类型”字段中选择 子进程”；将 'triggeredByEvent' 设置为 'true' 并点击确定按钮
- 右键单击元素并选择 是展开的”选项；这将在左上角显示元素名称

**为主进程模型事件子进程**

- 创建事件子流程subProcess4
  - 创建事件开始(@60)，然后双击它以显示 属性”对话框并且，在 类型”字段中，选择 事件子进程非中断 > 定时器”；点击确定按钮
  - 将序列流添加到目标抽象任务活动Task6(2000)
  - 将序列流添加到目标结束事件元素End6
- 创建事件子流程subProcess5
  - 创建事件开始(@80)，然后双击它以显示 属性”对话框并且，在 类型”字段中，选择 事件子进程中中断 > 定时器”；点击确定按钮
  - 将序列流添加到目标抽象任务活动Task7(2000)
  - 将序列流添加到目标结束事件元素End7

**模型子流程 subProcess1 和封闭的事件子流程**


- 创建一个 StartEvent Start2
  - 将序列流添加到目标抽象任务活动S (150)
  - 添加一个 流向目标的序列事件
- 创建一个事件子流程subProcess2
  - 创建事件开始(@20)，然后双击它以显示 属性”对话框并且，在 类型”字段中，选择 事件子进程非中断 > 定时器”
  - 将序列流添加到目标抽象任务活动Task2(100)
  - 将序列流添加到目标结束事件元素End4
- 创建一个事件子流程subProcess3
  - 创建事件开始(@30)，然后双击它以显示 属性”对话框并且，在 类型”字段中，选择 事件子进程中中断 > 定时器”
  - 将序列流添加到目标抽象任务活动Task3(40)
  - 将序列流添加到目标抽象任务活动Task4(20)
  - 将序列流添加到目标结束事件元素End5

**配置 BPSim**

使用这个表，我们在配置包中创建工作件并配置每个元素的参数值。

任务	行动
创造工件	<ul style="list-style-type: none"> <li>• 打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )</li> <li>• 创建加新按钮工件'事件子进程和Non-Interrupting' ( 在'选择/创建工作件器名称'字段中，<input type="text"/>并单击它的包级名称，然后在元素上输入点击保存按钮和确定按钮 )</li> </ul> <p>然后所有的 BPMN 元素将被加载到配置 BPSim窗口中。</p>
开始用于事件子进程中的开始事件	<p>从 配置 BPSim “对话框左侧的树中，展开 “StartEvent”。</p> <p>对于此处列出的每个元素，在 控件”选项卡上单击 新参数...”字段中的下拉箭头并选择参数 InterTriggerTimer”。单击 值”字段中的<input type="text"/>按钮以打开 参数”对话框并选择 常量&gt;数字”，然后输入值并选择 秒”。</p> <ul style="list-style-type: none"> <li>• 开始 3(@20) : 20 秒</li> <li>• 开始 4(@30) : 30 秒</li> </ul>



	<ul style="list-style-type: none"> <li>• 开始 5(@60) : 60 秒</li> <li>• 开始6 (@80) : 80 秒</li> </ul>
任务的处理时间	<p>从配置 BPSim窗口左侧的树中，展开“活动”。</p> <p>对于此处列出的每个元素，在“时间”选项卡上单击“新参数...”字段中的下拉箭头并选择参数“处理时间”。单击“值”字段中的  按钮以打开“参数”对话框并选择“常量&gt;数字”，然后输入值并选择“秒”。</p> <ul style="list-style-type: none"> <li>• 任务 1 (150) : 150 秒</li> <li>• 任务 2 (100) : 100 秒</li> <li>• 任务 3 (40) : 40 秒</li> <li>• 任务 4 (20) : 20 秒</li> <li>• 任务 5 (1000) : 1000 秒</li> <li>• 任务 6 (2000) : 2000 秒</li> <li>• 任务 7 (2000) : 2000 秒</li> </ul>

## 运行仿真

- 在“配置 BPSim”对话框工具 上，单击“运行”图标打开“运行仿真控制器”对话框
- 单击“运行”图标下拉箭头并选择“标准仿真”

The screenshot shows the BPSim Simulation Controller interface. The top section displays a log of events, and the bottom section shows the current token status for various BPMN elements.

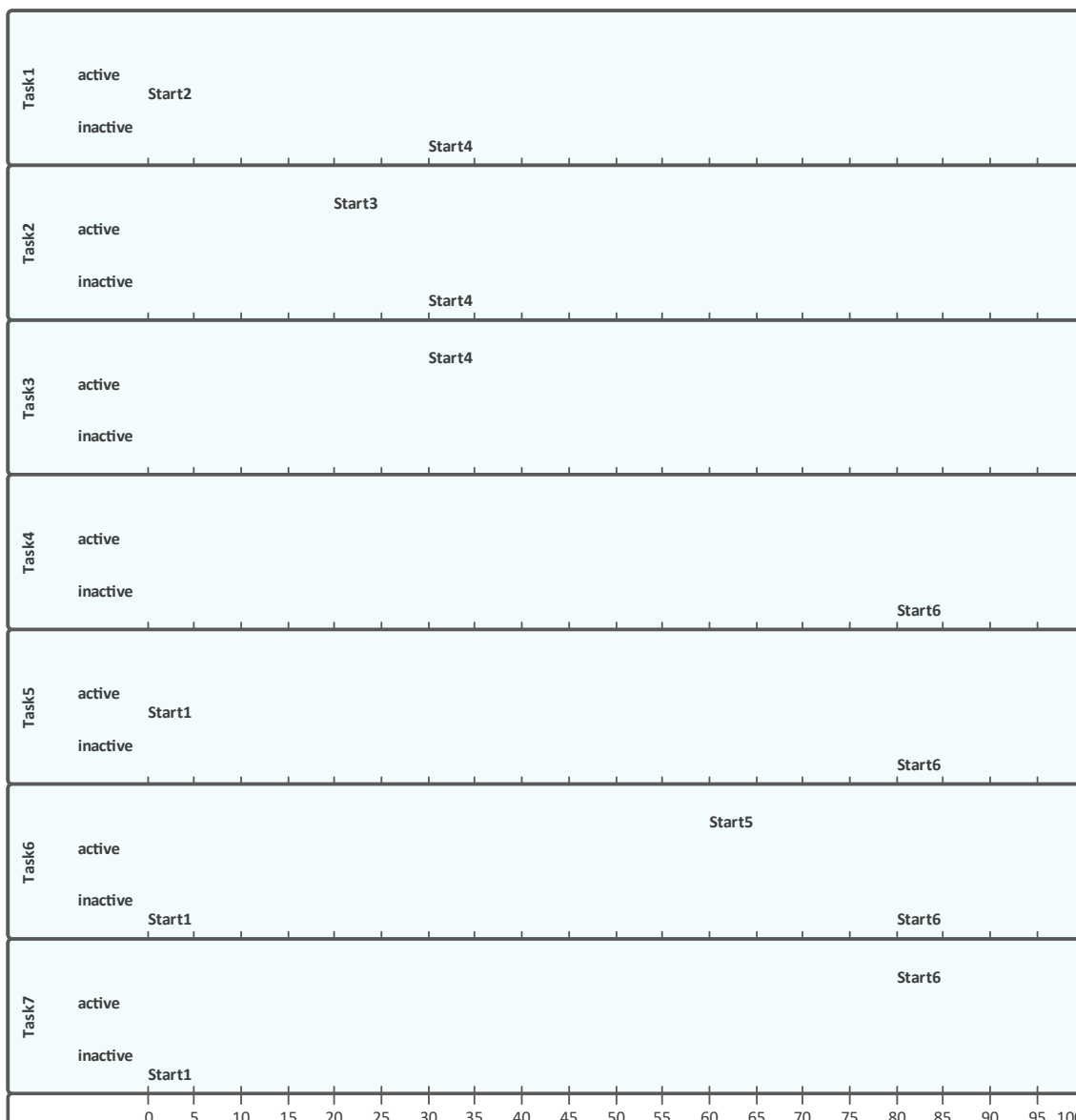
Type	Message
Info	Export BPMN Model - Succeeded.
Info	Loading BPMN Model - Succeeded.
Info	Simulation Engine Initialized
Info	Simulation Started.
Info	Simulation Finished.
Info	Simulation Result Generated.

Element	Runtime Value
Token Status	
<<StartEvent>> Start1	1
<<Gateway>> fork	1
<<Activity>> task5 (1000)	1
<<StartEvent>> Start2	1
<<Activity>> Task1 (150)	1
<<StartEvent>> Start3(@20)	1
<<Activity>> Task2 (100)	1
<<StartEvent>> Start4(@30)	1
<<Activity>> Task3 (40)	1
<<StartEvent>> Start5(@60)	1
<<Activity>> Task6 (2000)	1
<<Activity>> Task4 (20)	1
<<StartEvent>> Start6(@80)	1
<<Activity>> Task7 (2000)	1
<<EndEvent>> End7	1

### 分析

阅读结果，可能并不完全直截了当地看到发生了什么；但是，如果我们在时序图中为每个任务绘制生命线，它就会变得更加清晰。



- 事件(@20)是非中断的，它没有在 20 秒时停止Task1
- 事件(@30)正在中断，它在 30 秒时停止了Task1和Task2；它没有影响Task5因为Task5的封闭进程（主进程）级别高于Start4的封闭子进程（subProcess1）
- EventStart5(@60)是非中断的，它在 60 秒时启动了事件，而不影响Task3或Task5
- EventStart6(@80)正在中断，它在 80 秒时启动了 Task7事件并中断了与其封闭进程相同或更低级别的正在运行的任务（Task4、Task5、Task6）
- 只达到预期的End7

# 带有链接事件的斐波那契数生成器

链接事件是连接进程A两个部分的机制。链接事件可用于：

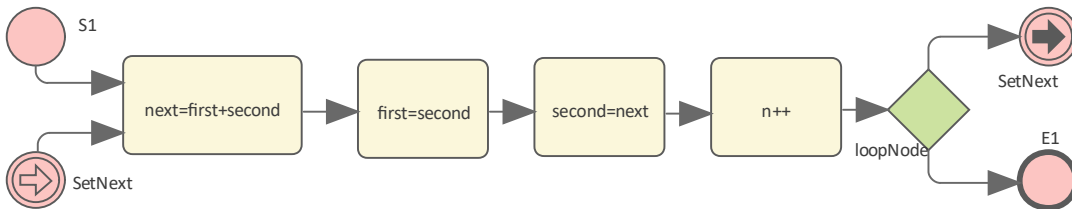
- 创建循环场景，作为“进程级别”中的通用 Go To 对象
  - 避免长序列流线；成对的链接事件可以用作跨多个页面打印进程的“页面”连接器
- 事件的使用仅限于单个进程级别（即不能链接父进程和子进程）。

可以有多个源链接事件，但只能有一个目标链接事件。

- 目标链接事件标记未填充，从源链接中“捕获”
- 源链接事件标记被填充以“扔”到目标链接

元素执行引擎在运行仿真时，源-目标链接事件是通过 ElementNAME 配对的，所以不能为空。

## 创建 BPMN 模型





- 创建一个 StartEvent *S1*
- 添加一个序列目标abstractTask活动元素 $next=first+second$ （打开“属性”对话框并将“类型”字段设置为“抽象任务”）
- 将序列流添加到目标 abstractTask 活动元素 $first=second$
- 将序列流添加到目标 abstractTask活动元素 $second=next$
- 将序列流添加到目标 abstractTask活动元素 $n++$
- 将序列流添加到目标独占网关元素 $loopNode$ （在即时菜单上，选择“独占”）
- 将一个序列流添加到这些目标元素中的每一个：
  - A Throwing Link 中间事件元素 $SetNext$ （打开“属性”对话框并设置“类型”字段到“投掷>链接”）和
  - 结束事件元素 $E1$
- 创建捕获链接中间事件元素 $SetNext$ （打开“属性”对话框并将“类型”字段设置为“捕获>链接”）
- 添加一个序列到目标元素 $next=first+second$

## 配置 BPSim


我们将使用属性参数来定义序列流如何形成一个循环，在该循环期间将生成一个斐波那契数。循环机制是通过一对事件实现的。

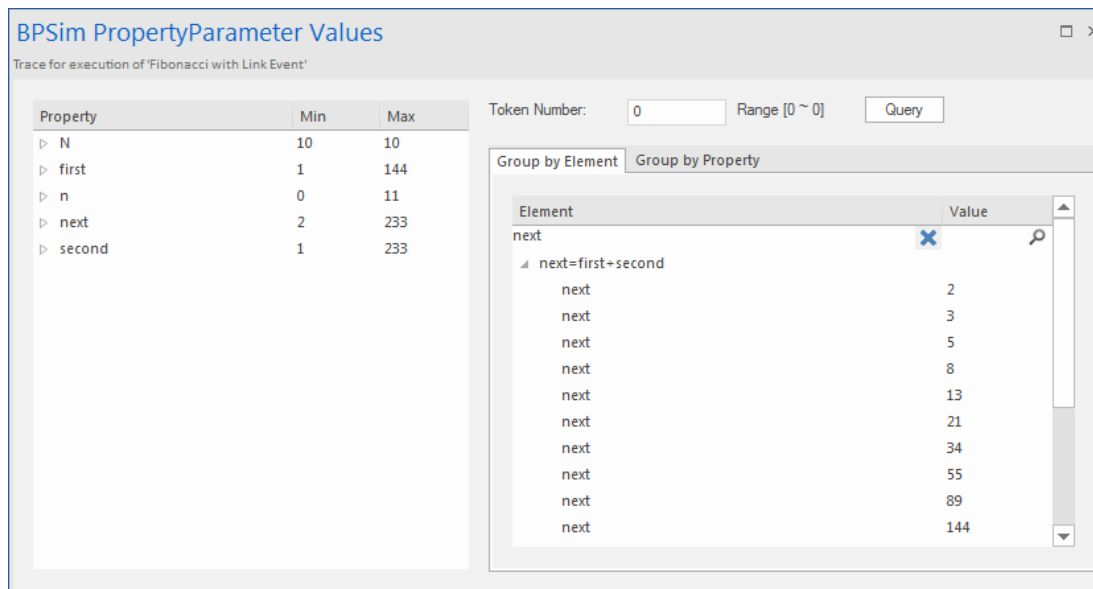
打开配置 BPSim窗口（'仿真>过程分析>进程>打开BPSim 管理器'）

任务	行动
元素：S1	在左侧的元素列表中，展开开始事件组并单击 <i>S1</i> 。 单击“控件选项卡”和“新参数”下拉箭头；选择“触发器计数”。 在“值”字段中输入“1”。

	<p><b>单击 属性”选项卡</b></p> <p>改写新属性文本以创建这些属性：</p> <ul style="list-style-type: none"> <li>• N -并在 值”字段中输入 <code>f0</code>”作为要生成的斐波那契数的总数</li> <li>• 首先-在 值”字段中输入 <code>1</code>”</li> <li>• 第二-在 值”字段中输入 <code>1</code>”</li> <li>• n -并在 值”字段中输入 <code>0</code>”作为第n个新的斐波那契数</li> </ul>
<p>元素：next=first+second</p>	<p>在元素类型列表中，展开活动组并单击<code>next=first+second</code>。</p> <p>单击 属性”选项卡并用 下一步”覆盖新属性文本。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{first}+{second}</code>”。</p> <p>点击确定按钮。</p>
<p>元素：first=second</p>	<p>在元素类型列表中，在活动组中单击<code>first=second</code>。</p> <p>单击 属性”选项卡并用 第一个”改写新属性文本。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{second}</code>”。</p> <p>点击确定按钮。</p>
<p>元素: second=next</p>	<p>在元素类型列表中，在活动组中单击<code>second=next</code>。</p> <p>单击 属性”选项卡并用 第二个”覆盖新属性文本。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{next}</code>”。</p> <p>点击确定按钮。</p>
<p>元素：n++</p>	<p>在元素类型列表中，在活动组中单击<code>n++</code>。</p> <p>单击 属性”选项卡并用 n”改写新属性文本。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{n}+1</code>”。</p> <p>点击确定按钮。</p>
<p>网关条件</p>	<p>在元素列表中，展开 Gateway 组和 LoopNode元素，然后单击<code>SetNext</code>。</p> <p>单击 控件”选项卡和 新参数”下拉箭头，然后选择 条件”。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{n} &lt;= { N }</code>”。</p> <p>点击确定按钮。</p> <p>现在点击<code>E1</code>。</p> <p>单击 控件”选项卡和 新参数”下拉箭头，然后选择 条件”。</p> <p>在 值”字段中，单击  按钮，单击 表达式”选项卡并输入表达式 <code>{n} &gt; { N }</code>”。</p> <p>点击确定按钮。</p>

## 运行仿真

- 在配置 BPSim “对话框的工具”中，单击“运行”图标；“仿真仿真控制器”对话框显示
- 单击“运行”图标下拉箭头并选择“标准仿真”
- 模拟完成后，单击工具  完成；“BPSim PropertyParameter Values”对话框显示
- 单击查询按钮和“按元素分组”选项卡，然后展开“下一步=第一+第二”；列出所有属性的快照值
- 应用过滤器“下一步”（右键单击列表标题，选择“切换过滤器栏”并在“元素”标题下键入“下一步”）；结果将类似于此图像：



The screenshot shows the 'BPSim PropertyParameter Values' dialog box. It has a title bar with a close button. Below the title bar, it says 'Trace for execution of 'Fibonacci with Link Event''. There are two tabs: 'Group by Element' (selected) and 'Group by Property'. On the left, there is a table with columns 'Property', 'Min', and 'Max'. On the right, there is a search bar and a 'Query' button. Below that, there is a table with columns 'Element' and 'Value'. The 'Element' column is expanded to show a list of 'next' values.

Property	Min	Max
N	10	10
first	1	144
n	0	11
next	2	233
second	1	233

Element	Value
next	
next=first+second	
next	2
next	3
next	5
next	8
next	13
next	21
next	34
next	55
next	89
next	144

再生成十个斐波那契数：

2,3,5,8,13,21,34,55,89,144

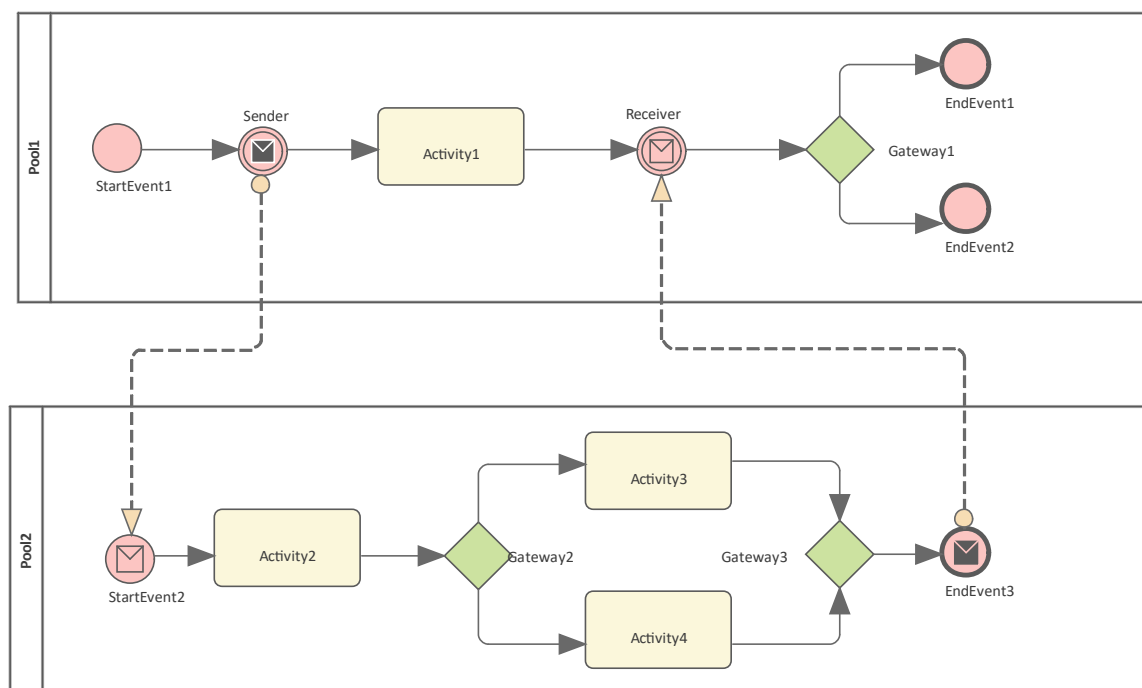
## 信息事件

在正常序列流中使用时，信息事件可用于发送或接收消息。

- 向参与者发送消息时，复制所有属性参数的值；一旦发送消息，令牌将继续沿着序列流
- 接收消息时，当收到消息时触发该事件。

这个例子展示了信息事件的特征。我们将首先创建 BPMN 模型，然后配置运行并运行仿真。

### 创建 BPMN 模型



### 序列

#### 池1

- 令牌从 StartEvent1 开始
- 收到令牌后，Sender（一个 Throwing Intermediate 信息事件）创建一条消息并将当前属性值复制到该消息中
- 发件人将消息发送给“收件人”参与者（Pool2，StartEvent2）
- 发送者沿着它的序列流转发令牌，直到接收者
- 令牌在 Receiver 处等待，直到消息到达

#### 池2

- StartEvent2 收到消息并启动令牌
- StartEvent2 复制消息中的值并将其设置在令牌中
- StartEvent2 沿其序列流转发令牌，直到 EndEvent3
- EndEvent3 创建一条消息并将当前属性值复制到该消息中
- EndEvent3 将消息发送给“发件人”参与者（Pool1，Receiver）

#### 池1续

- 等待的 Receiver 被唤醒，属性值从到达的消息中更新

### 创建图表

- 创建 BPMN 2.0协作图
- 选择 在新的 CollaborationModel 中创建此图表”选项
- 通过将 泳池”图标从工具箱拖到图表上来创建Pool1和Pool2

### 池1内

- 创建一个类型为 “None”的开始事件，命名为StartEvent1
- 向目标 “Throwing信息”类型的中间事件添加一个序列流，称为Sender
- 添加一个序列流到'abstract'类型的目标活动，称为Activity1
- 将序列流添加到目标：
  - 类型为 “无”的结束事件，称为EndEvent1
  - 类型为 “无”的结束事件，称为EndEvent2

### 池2内

- 创建一个 “信息”类型的开始事件，称为StartEvent2
- 添加一个序列流到'abstract'类型的目标活动，称为Activity2
- 将一个序列流添加到 “独占”类型的目标网关，称为Gateway2
- 将序列流添加到目标：
  - “抽象”类型的活动，称为Activity3
  - “抽象”类型的活动，称为Activity4
- 将Activity3和Activity4中的序列流添加到 “Exclusive”类型的目标网关，称为Gateway3
- 在 “信息”类型的结束事件中添加一个序列流，称为EndEvent3

### 信息流








- 添加一个从Sender到StartEvent2的信息流
- 将EndEvent3中的信息流添加到Receiver

## 配置 BPSim


为了显示信息流携带值的能力，我们创建了一个属性参数'M1'并在每个活动中更改其值。然后我们使用 'M1'的值作为序列流条件表达式的一部分。

任务	描述
创建工作件包	<ul style="list-style-type: none"> <li>• 打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )</li> <li>•  事件/创建工作件”字段中的 事件/创建按钮”并创建一个工作件 消息事件 -开始事件-中间事件-结束事件”的浏览器</li> <li>• 在 选择包”字段中选择包含模型的包 模型中的所有 BPMN 元素都加载到配置 BPSim模型中。</li> </ul>
属性值	<p>我们将在StartEvent1处为属性参数 'M1'赋予初始值 10。然后我们在令牌流流程时更改值，并且值在参与者之间复制。</p> <p>在对话框左侧的元素列表中：</p> <ul style="list-style-type: none"> <li>• 展开 “StartEvent”组，单击StartEvent1并在 属性”选项卡上，用 'M1'覆盖新属性；在 值”字段中单击  按钮并选择 常量”和 数字”，然后在 常量数字”字段中键入 '10'”</li> <li>• 展开 “活动”组，单击Activity1和 属性”选项卡，并用 'M1'覆盖新属性；在</li> </ul>



	<p>值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} + 100"</p> <ul style="list-style-type: none"> <li>单击Activity2和 属性"选项卡，然后用 M1"覆盖新属性；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} + 10"</li> <li>单击Activity3和 属性"选项卡，然后用 M1"覆盖新属性；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} + 10"</li> <li>单击Activity4和 属性"选项卡，并用 M1"覆盖新属性；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} + 1"</li> </ul> <p>提示：{PropertyName} 的格式是 <code>getProperty("PropertyName")</code> 的一种方便的缩写形式。</p>
<p>控件参数</p>	<p>在这个模拟中，我们只需要一个令牌来评估模型的行为。</p> <ul style="list-style-type: none"> <li>在展开的 "StartEvent" 组中，单击StartEvent1和 控件"选项卡；单击 新参数"下拉箭头并选择 触发器计数"，然后输入 "1" 的 值"</li> </ul> <p>现在为网关的传出序列流设置条件。在对话框左侧的元素列表中，展开 网关"组：</p> <ul style="list-style-type: none"> <li>展开Gateway1，单击EndEvent1并在 控件"选项卡上，然后单击 新参数"下拉箭头并选择 条件"；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} &gt;= 50"</li> <li>单击EndEvent2并在 控件"选项卡上，然后单击 新参数"下拉箭头并选择 条件"；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} &lt; 50"</li> <li>展开Gateway2，单击Activity3并在 控件"选项卡上，然后单击 新参数"下拉箭头并选择 条件"；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} &gt;= 15"</li> <li>单击 控件"控件卡，然后单击 新参数"下拉箭头并选择 条件"；在 值"字段中单击  按钮并选择 表达式"，然后在 表达式"字段中键入 "{M1} &lt; 15"</li> </ul>

### 运行仿真

- 在 配置 BPSim "对话框工具 上，点击运行按钮； 仿真仿真控制器"对话框显示
- 点击 运行"图标下拉箭头，选择 标准仿真"；模拟开始
- 模拟完成后，点击  按钮； 属性参数值"对话框显示，在模拟过程中跟踪属性值
- 在 令牌编号"字段中输入 0"，然后单击查询按钮和 按属性分组"选项卡

BPSim PropertyParameter Values

Trace for execution of 'MessageEvent-StartEvent-IntermediateEvent-EndEvent'

Property: M1, Min: 10, Max: 110

Token Number: 0, Range [0 ~ 0], Query

Group by Element, Group by Property

Property	Value
M1	
StartEvent1	10
Sender	10
StartEvent2	10
Activity2	20
Activity1	110
Gateway2	20
Activity3	30
Receiver	110
Gateway3	30
EndEvent3	30
Gateway1	30
EndEvent2	30

### 分析

由于Activity1的 'ProcessingTime' 被设置为分布值，结果是：

- [Process1] 在Pool1.StartEvent1之后的 'M1' 值为 '10'，正如预期的那样
- \*[Process2] Pool2.StartEvent2的 'M1' 值为 '10';此值来自Pool1.Sender发送的消息

现在实际上有两个'M1' - Process1.M1和Process2.M1

- [Process2] Pool2.Activity2增加Process2.M1 10； [进程2.M1 == 20]
- 【Process1】 Pool1.Activity1增加Process1.M1 100； [过程1.M1 == 110]
- [Process2] 评估条件表达式；当 '20 > 15' 时，令牌将流向Activity3 [Process2.M1 == 20]
- [Process2] Pool2.Activity3增加Process2.M1 10； [进程2.M1 == 30]
- [Process1] Pool1.Receiver已到达并等待[Process1.M1 == 110]
- [Process2] Pool2.Gateway3作为 Merge 节点，继续EndEvent3 [Process2.M1 == 30]
- \*[Process1] Pool1.Receiver 被消息唤醒（携带M1 == 30）并且Process1.M1的值从 110 变为 30
- [Process1] 评估条件表达式；作为 '30 < 50'，令牌将流向EndEvent2 [Process1.M1 == 30]

### 注记

- 标有星号(\*)的行是信息流的效果
- 定义了流程中的顺序；然而，两个进程之间的顺序并不总是可预测的
- 投掷信息事件分叉另一个进程；捕获信息用作线程同步

## 信号事件

A信号事件通过发布-订阅集成提供功能“投掷者”和“捕手”的功能。A“投掷者”将广播信号而不是将其发送给特定进程；侦听该特定事件的任何进程都可以使用信号开始事件触发新实例。

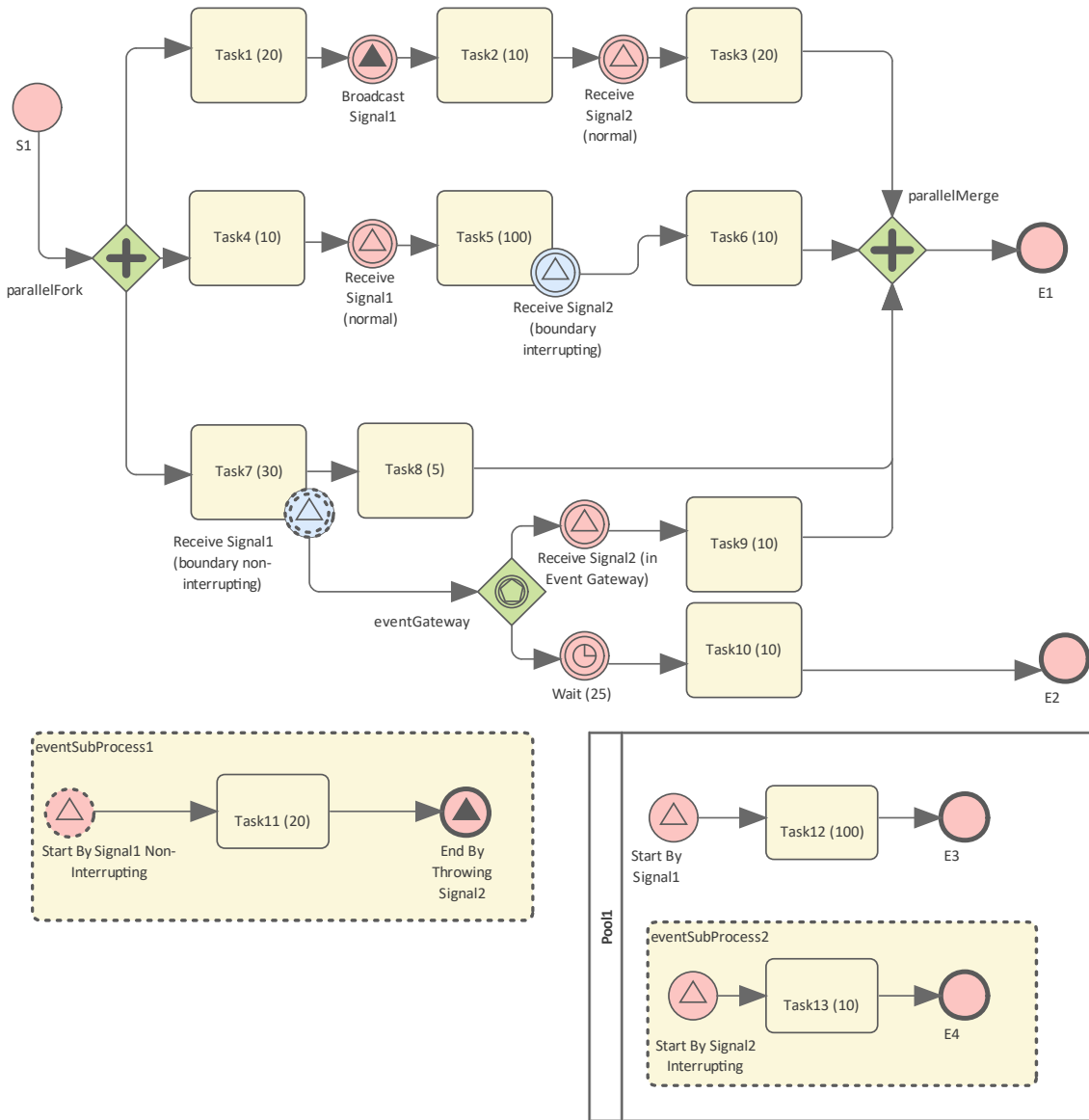
信号可以A抛出中间事件或抛出结束事件中抛出，并且可以在开始事件或捕获中间事件（包括边界信号事件中捕获。

在此示例中，我们通过 BPSim 参数设置演示了这些信号事件及其对任务生命线的影响。

- 开始信号事件：
  - 在顶层进程（Pool1）中通过 *Signal1* 开始
  - 开始 *Signal2* 中断事件子流程 *eventSubProcess2*
  - 开始 *Signal1* Non Interrupting in event subprocess *eventSubprocess1*
- 抛出中间信号事件：
  - 广播信号1
- 捕获中间信号事件：
  - 接收信号1（正常）
  - 接收信号2（正常）
  - 接收信号2（边界中断）
  - 接收信号1（边界不中断）
  - 接收信号2（在事件网关中）
- 结束信号事件：
  - 以抛出 *Signal2* 结束

## 创建 BPMN模型

为了演示通过协作信号事件跨进程进行通信的能力，我们创建了一个具有主泳池和另一个泳池（信号）中的进程的模型。



### 创建协作和主流程

创建一个新的协作图 *CollaborationForTestingSignalEvents* · ( 选择 在名为 BP 的新协作模型中创建此图表 "选项) 。 右键单击图表名称窗口并选择 封装进程浏览器" 选项。

创建了A泳池和一个流程 *BusinessProcess\_PoolMain* · 并使用自动值设置这些标签：

- *CollaborationForTestingSignalEvents.mainPool* 设置为 *PoolMain*
- *PoolMain.processRef* 设置为 *BusinessProcess\_PoolMain*

### 为主流程创建元素

创建一个开始事件 *S1* 并将一个序列流添加到一个分叉 Parallel Gateway *parallelFork*

将序列流添加到：

- 一个任务(20)然后添加这个序列流链：
  - 到一个 Throwing 中间信号事件广播 *Signal1*
  - 然后到一个任务(10)
  - 然后到一个捕获中间信号事件接收 *Signal2* ( 正常 )
  - 然后到一个抽象任务(20)
  - 然后到合并并行网关 *parallelMerge*

- 然后结束事件E1

- 一个任务(10)然后添加这个序列流链：
  - 到一个捕获的中间信号事件接收信号接收正常)
  - 然后到一个任务(100) · 在它上面创建一个边界捕获中间信号事件接收Signal2 ( 边界中断 )
  - 然后到一个抽象任务(10)
  - 然后到前面的 Merge Parallel Gateway *parallelMerge*
- 一个任务(30) · 然后添加这个序列流链：
  - 抽象任务8 (5)
  - 然后到前面的 Merge Parallel Gateway *parallelMerge*

在边界(30)上 · 创建一个事件非中断捕获中间信号接收信号接收 ( 边界非中断 ) 。将序列流添加到事件网关 *eventGateway* · 并将序列流添加到：

- A捕获中间信号事件接收2 ( 在网关中 ) · 然后这个序列流：
  - 抽象任务9 (10)
  - 然后到前面的 Merge Parallel Gateway *parallelMerge*
- A捕获中间定时器事件Wait (25) · 然后是这个序列流链：
  - 抽象任务10 (10)
  - 然后结束事件E2

**在主进程中创建事件子进程 ( 由非中断的开始信号事件触发 )**

- 创建一个活动 *eventSubProcess1* · 并在其 属性"对话框中 · 将 类型"字段设置为 *subProcess* 并将属性 "triggeredByEvent" 更改为 *true*
- 在 *eventSubProcess1* 中 · 通过开始开始-*Interrupting* 创建一个开始事件开始 · 并在其 属性"对话框中 · 将 类型"字段设置为事件-*Process Non-Interrupting* > 信号
- 将序列流添加到目标任务(20)
- 通过 *Throwing Signal2* 将序列流添加到目标事件 · 并在元素 属性"对话框中 · 将 类型"字段设置为信号

**创建另一个进程**

- 从工具中 · 将 泳池"图标拖放到图表上 · 并将元素池命名为工具箱
- 在浏览器窗口中右击 *Pool1* · 选择 封装进程"选项；创建了一个流程 *BusinessProcess\_Pool1* 并将标签 "Pool1.processRef" 设置为 *BusinessProcess\_Pool1*

**为 *Pool1* 创建主进程**


- 通过开始信号一个信号开始事件开始
- 将序列流添加到目标任务(100)
- 将序列流添加到目标结束事件E3

**创建Event子进程事件 *Pool1***

- 创建一个活动 *eventSubProcess1* · 并在 属性"对话框中 · 将 类型"字段设置为 *subProcess* ；将属性 "triggeredByEvent" 更改为 *true*
- 在 *eventSubProcess2* 中 · 通过 *Signal2* 开始创建一个开始事件开始 · 并在 属性"对话框中 · 将 类型"字段设置为事件子进程中中断>信号
- 将序列流添加到目标任务(10)
- 将序列流添加到目标结束事件E4

**创建信号元素并配置信号事件**

在 BPMN 2.0 工具箱中 · 展开 BPMN 2.0 - Types" 页面并将 信号"图标拖到图表上；命名 *ElementSignal1* 元素再次将图标拖到图表上以创建 *Signal2* 。这些是根元素 ( 所有进程都可以使用 ) · 因此它们将直接在模型包下创建。

双击每个信号事件元素 · 然后在 信号"标签的 值"字段中 · 单击  按钮并浏览到适当的信号元素。

提示：或者 · 您可以将浏览器窗口中的信号元素拖放到图表中的事件元素上；将显示一个上下文菜单 · 您可以从中选择 设置信号参考" 选项。

- 将 signalRef 设置为 ‘Signal1’ :
  - 广播信号1
  - 顶级进程 (开始) 中的Signal1开始
  - 在事件子进程eventSubprocess1中通过开始不中断
  - 接收信号1 (正常)
  - 接收信号1 (边界不中断)
- 将 signalRef 设置为 ‘Signal2’ :
  - 开始Signal2中断事件子进程eventSubProcess2
  - 接收信号2 (正常)
  - 接收信号2 (边界中断)
  - 接收信号2 (在事件网关中)

## 配置 BPSim


在本节中，我们创建配置工件，指定模型包并配置每个元素的值。

配置非常简单，因为所有信号事件都不需要任何信号配置。我们所要做的就是设置任务的处理时间，这样我们就可以观察进程、线程和任务是如何启动和中断的。

任务	描述
设置配置	<ul style="list-style-type: none"> <li>• 打开配置 BPSim窗口 ( '仿真&gt;过程分析&gt;进程&gt;打开BPSim 管理器' )</li> <li>• 创建加新元素完全工件示例'选择/创建工件  按钮并单击父包并单击按钮，然后在示例按钮上输入，然后键入单击保存按钮和确定按钮 )</li> </ul> <p>然后所有的 BPMN 元素将被加载到配置 BPSim窗口中。</p>
非信号事件	<ul style="list-style-type: none"> <li>• 在对话框左侧的元素列表中，展开 ‘StartEvent’组，然后单击SI和 控件“选项卡；单击 新参数”下拉箭头并选择 触发器计数”，然后在 值”字段中键入 ‘1’ ”</li> <li>• 展开 ‘IntermediateEvent’组，然后单击Wait (25)和 控件”选项卡；单击 新参数”下拉箭头并选择 ‘InterTriggerTimer’”，然后单击 值”字段中的  按钮；选择 ‘Constant’和 ‘Numeric’”，在 ‘Constant Numeric’”字段中输入 ‘25’”，在 ‘TimeUnit’”字段中输入 ‘seconds’”</li> </ul>
进程虚拟变量	<p>模拟控制器显示一个列表，显示每个元素的运行时令牌计数。例如，在模拟中，有 4 个令牌通过了Gateway元素。这对于某些统计和分析非常有用。但是，它没有显示在模拟期间何时遍历了parallelMerge。为了获得单个令牌的准确跟踪，我们使用属性跟踪实用程序，它依赖于属性参数。所以我们创建了一个虚拟参数。</p> <p>在 配置”对话框中，展开 业务流程”组。</p> <ul style="list-style-type: none"> <li>• 单击BusinessProcess_Main和 属性”选项卡，然后用属性改写属性；在 值”字段中，单击  按钮并单击 常量”和 数字”，然后在 常量数字”字段中键入 ‘0’ ”</li> <li>• 单击BusinessProcess_Pool1并执行与BusinessProcess_Main完全相同的操作</li> </ul>
任务处理时间	<p>展开 活动”组和此处列出的每个任务元素：选择 时间”选项卡，单击 新参数”下拉箭头并选择 处理时间”，然后单击 值”列上的  按钮，选择 常量”和 数字”，在 常量数字”字段中键入指示的值，然后在 时间单位”字段中选择 秒”。</p> <ul style="list-style-type: none"> <li>• 任务 1 (20) : 20 秒</li> <li>• 任务 2 (10) : 10 秒</li> </ul>

- 任务 3 (20) : 20 秒
- 任务 4 (10) : 10 秒
- 任务 5 (100) : 100 秒
- 任务 6 (10) : 10 秒
- 任务 7 (30) : 30 秒
- 任务 8 (5) : 5 秒
- 任务 9 (10) : 10 秒
- 任务 10 (10) : 10 秒
- 任务 11 (20) : 20 秒
- 任务 12 (100) : 100 秒
- 任务 13 (10) : 10 秒

## 运行仿真

- 在 配置 BPSim “对话框工具” 上，单击 运行“图标打开 运行仿真控制器”对话框
- 单击 运行“图标下拉箭头并选择 标准仿真”
- 仿真完成后，单击工具栏上的  按钮，显示 BPSim PropertyParameter Values“对话框
- 单击 查询属性”按钮和 按属性分组”选项卡，然后展开 dummyVariable”

The screenshot shows the 'BPSim PropertyParameter Values' window. It has a title bar with a close button. Below the title bar, it says 'Trace for execution of 'SignalEvent Complete Example''. There are two tabs: 'Group by Element' and 'Group by Property'. The 'Group by Property' tab is selected. On the left, there is a small table with columns 'Property', 'Min', and 'Max'. The 'Property' column contains 'dummyVariable', and both 'Min' and 'Max' columns contain the value '0'. On the right, there is a larger table with columns 'Property' and 'Value'. The 'Property' column lists various tasks and signals, and the 'Value' column lists the value '0' for each. At the top right of the main area, there are input fields for 'Token Number: 0', 'Range [0 ~ 0]', and a 'Query' button.

Property	Min	Max
dummyVariable	0	0

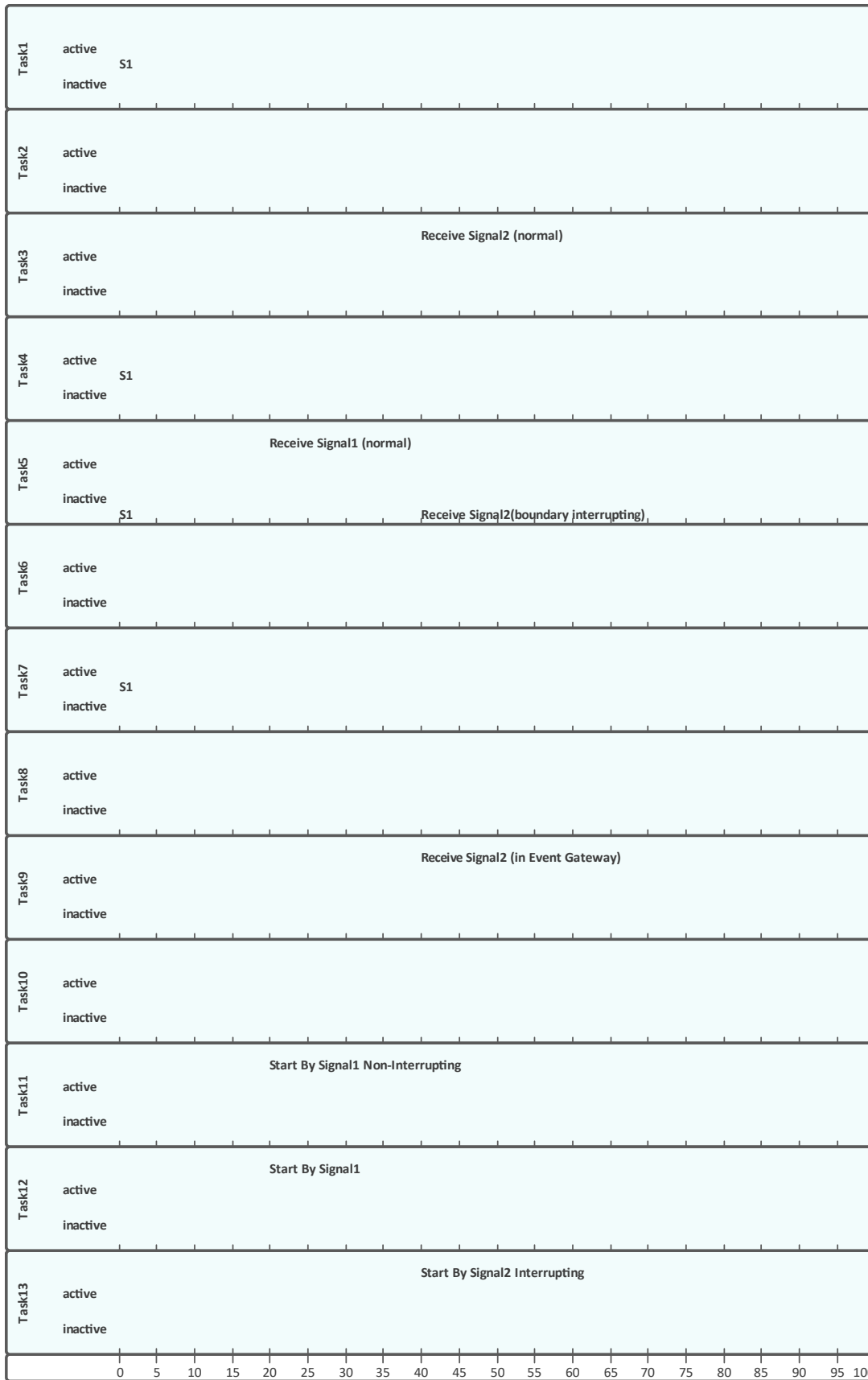
  

Property	Value
dummyVariable	0
S1	0
parallelFork	0
Task7 (30)	0
Task4 (10)	0
Task1 (20)	0
Receive Signal1 (normal)	0
Broadcast Signal1	0
Task5 (100)	0
Start By Signal1 Non-Interrupting	0
Task11 (20)	0
eventGateway	0
Start By Signal1	0
Task12 (100)	0
Task2 (10)	0
Task8 (5)	0
Receive Signal2 (normal)	0
parallelMerge	0
End By Throwing Signal2	0
Task3 (20)	0
Task9 (10)	0
Task6 (10)	0
Start By Signal2 Interrupting	0
Task13 (10)	0
parallelMerge	0
parallelMerge	0
E4	0
parallelMerge	0
E1	0

### 分析

从模拟的直接结果来看，发生了什么可能并不明显；但是，如果我们为每项任务画出生命线，就会变得非常清晰。





- Task1、Task4和Task7并行启动

- *Task2*在*Task1*完成后立即开始 ( 没有在投掷事件处停止 )
- 在 20 秒时, *Signal1*由 *Throwing IntermediateEventBroadcast*事件广播, 并且:
  - 接收信号 1 ( 正常 ) 被激活, 任务 5开始
  - 开始*Signal1 Non-Interrupting*已激活, 并且*eventSubProcess1*中的*Task11*已启动
  - 开始*Signal1*已激活, 并且*Pool1*中的 *Task12*已启动
- 在 40 秒时, 事件*By Throwing Signal2*广播了 *Signal2*, 并且:
  - 接收信号 2 ( 正常 ) 被激活并且任务 3开始
  - *Task5*被中断, *Task6*开始
  - 接收信号 2 ( 在事件网关中 ) 被激活并且任务 9开始
  - 开始通过 *Signal2* 中断被激活, 并且:
    - > *Pool1*中的主进程被中断, *Task12*停止
    - > *eventSubProcess2*中的*Task13* 已启动
- 50 秒到达*E4*时, *BusinessProcess\_Pool1*中的*eventSubProcess2*完成
- 在 60 秒到达*E1*时, *BusinessProcess\_MainPool*完成
- *Intermediate*事件(25)没有被激活, 因为网关中的信号事件首先被激活; 结果, *Task10*从未启动

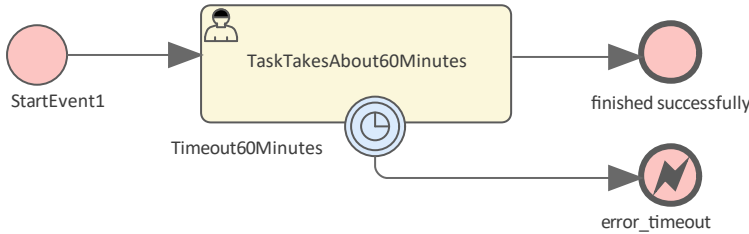
笔记: 每个任务的实际运行时间可以从生成的 *BPSimReport*元素中观察到, 通过:

1. 双击 <<*BPSimReport*>>元素。
2. 扩展 “时间”组。
3. 扩展任务元素。
4. 检查 “任务总时间”。

例如*ElementTask5 (100)*, 虽然我们设置它的*processingTime*为 100 秒, 但*Total*时间是 20 秒, 任务元素秒时被接收信号 2 ( 边界中断 ) 中断。

# 边界事件

## 创建 BPMN 模型



- 创建一个开始事件 *StartEvent1*
- 给目标用户添加一个序列流任务 *TaskTakes.About60Minutes*
- 将序列流添加到目标结束事件成功完成
- 创建一个中间事件，将工具箱中的图标拖放到工具箱上；从自动菜单中选择 'Edge-Mounted' 和 'Timer'，然后调用元素
- 将序列流添加到目标事件(Error) *error\_timeout*


## 配置 BPSim

在本节中，我们创建配置工件，识别父包并设置每个元素的参数值。

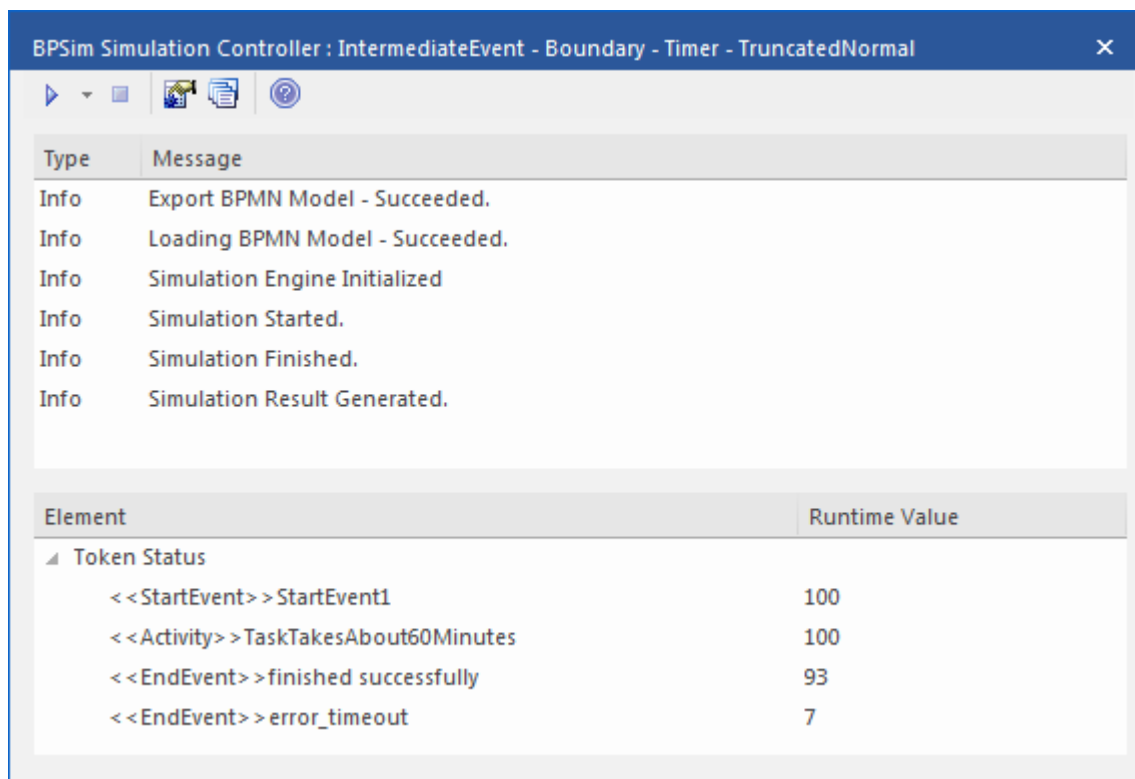
对象	行动
创建工作包	<ul style="list-style-type: none"> <li>• 打开 配置 BPSim “对话框 ( 仿真&gt;流程分析&gt;仿真&gt;进程BPSim 管理器” )</li> <li>• '在'创建新包元素名称'字段中的'创建资源包/工件'字段中选择 'Intermediate'边界工件，然后单击 <input type="button" value="..."/> 按钮，选择其加新组件，单击按钮，然后单击在保存按钮和确定按钮上)</li> </ul> 然后所有的 BPMN 元素将被加载到 配置 BPSim “对话框中。
开始事件1	在对话框左侧的元素列表中，展开 ‘StartEvent’组并单击 <i>StartEvent1</i> 。 <ul style="list-style-type: none"> <li>• 单击 控件”选项卡</li> <li>• 单击 新参数”下拉箭头并选择 “TriggerCount”</li> <li>• 在 值”字段中输入 ‘100”</li> </ul>
任务大约需要60分钟	在对话框左侧的元素列表中，展开 活动”组并单击 <i>TaskTakes.About60Minutes</i> 。 <ul style="list-style-type: none"> <li>• 点击 时间”标签</li> <li>• 单击 新参数”下拉箭头并选择 ProcessingTime”</li> <li>• 在 值”字段中单击 <input type="button" value="..."/> 按钮并选择 分布”和 截断正常”</li> <li>• 在 平均值”字段中输入 ‘50”</li> <li>• 在 标准偏差”字段中输入 ‘10”</li> <li>• 在 Min”字段中输入 ‘0”</li> <li>• 在 最大”字段中输入 ‘1000”</li> </ul>

	<ul style="list-style-type: none"> <li>• 点击确定按钮</li> </ul>
超时60分钟	<p>在对话框左侧的元素列表中，展开 <code>IntermediateEvent</code> 组并单击 <code>Timeout60Minutes</code>。</p> <ul style="list-style-type: none"> <li>• 单击 控件”选项卡</li> <li>• 单击 新参数”下拉箭头并选择 <code>InterTriggerTimer</code>”</li> <li>• 将值设置为 <code>000:000:000 001:00:00</code>” (即1小时)</li> </ul>

## 运行仿真

- 在配置 BPSim窗口工具栏上，单击 运行”图标，打开 运行仿真控制器”对话框
- 点击 运行”图标下拉箭头并选择 标准仿真”
- 仿真完成后，单击工具栏上的  按钮，显示 `BPSim PropertyParameter Values`”对话框
- 单击 查询属性”按钮和 按属性分组”选项卡，然后展开 `dummyVariable`”

在模拟中，我们得到以下结果：



## 分析

由于 `TaskTakesAbout60Minutes` 的 `ProcessingTime` 被设置为一个分布值，结果是：

- 100 人中有 93 人在 1 小时内完成，所以正常流程成功完成才生效
- 100 个中有 7 个在超过 1 小时内完成，因此 `error_timeout` 的异常流程生效

## 其它配置

示例中还有另外两个业务流程仿真管理文件夹，分别设置了工件的常量值 50 分钟和 80 分钟，其他设置保持不变。

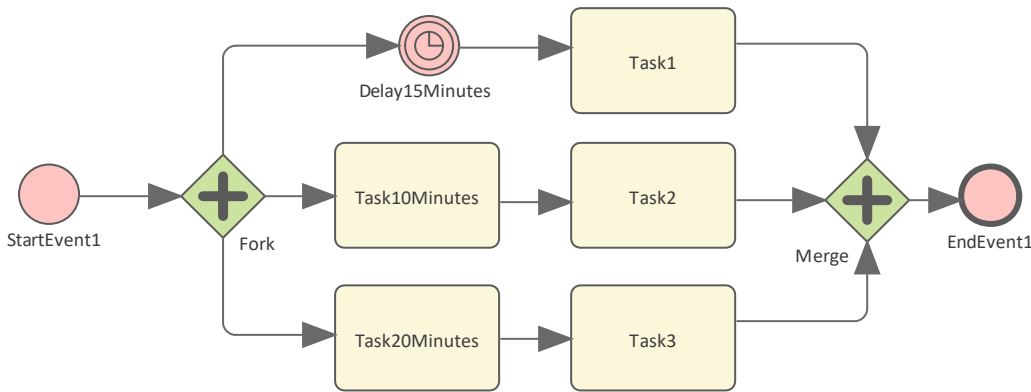
在这两个上运行工件：

- 配置为 50 分钟的处理时间始终以正常流程完成
- 配置为 80 分钟的处理时间始终在异常流中完成

# 事件独立中间事件

当 Timer事件在正常序列流中用作独立元素时，它充当延迟机制。

## 创建 BPMN模型



- 创建一个名为开始的启动事件
- 向名为分叉的目标并行网关添加一个序列
- 将序列流添加到：
  - 一个名为Delay15Minutes的独立定时器中间事件，并由此产生一个序列流到一个名为Task1的活动
  - 一个名为Task10Minutes的活动，从那个序列流到一个名为Task2的活动
  - 一个名为Task20Minutes的活动，从那个序列流到一个名为Task3的活动
- 从Task1、Task2和Task3创建序列称为Merge的合并并行网关
- 将序列流添加到名为EndEvent1的目标EndEvent


## 配置 BPSim

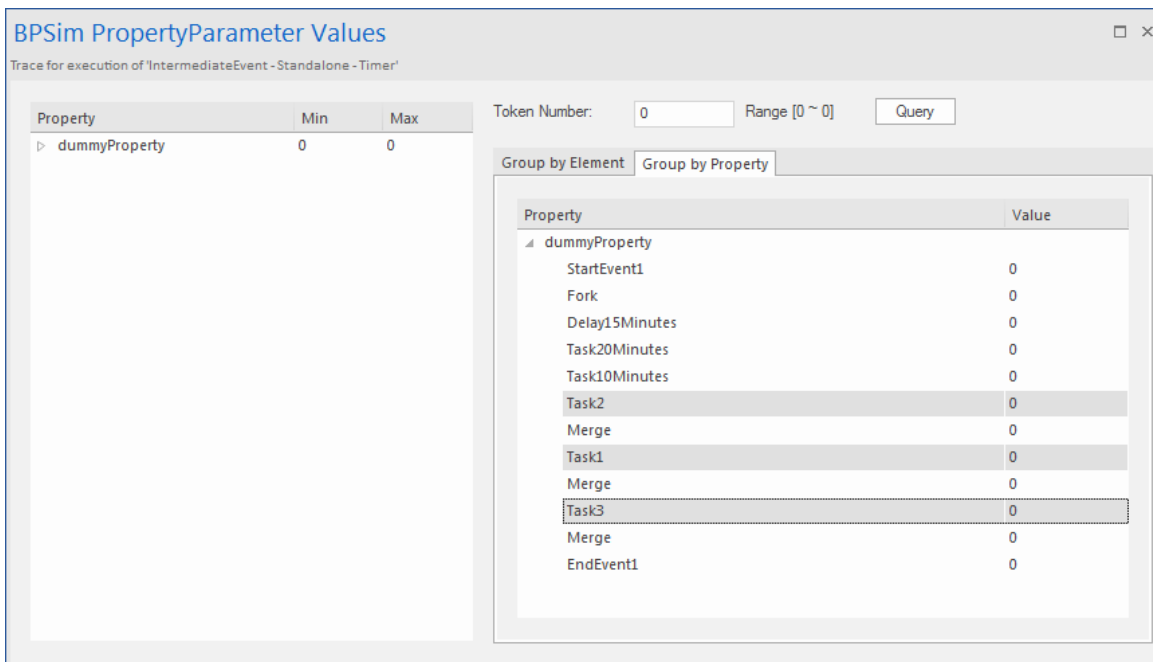
在本节中，我们创建配置工件，指定模型包并配置每个元素的值。

物件	行动
创建工作包	<ul style="list-style-type: none"> <li>• 打开 配置 BPSim “对话框 ( 仿真&gt;流程分析&gt;仿真&gt;进程BPSim 管理器” )</li> <li>• 创建一个工件名称，然后单击 <b>IntermediaEvent</b> “按钮，在 选择/创建工作” 字段中单击 选择/创建独立 <input type="button" value="..."/> “按钮并选择其父包，然后在元素名称中输入 加新”并单击保存按钮和确定按钮 )</li> </ul> 然后所有的 BPMN 元素将被加载到 配置 BPSim “对话框中。
开始事件1	<ul style="list-style-type: none"> <li>• 在对话框左侧的元素列表中，展开 <b>StartEvent</b>”，然后单击<b>StartEvent1</b>和控件”组</li> <li>• 单击 新参数”下拉箭头并选择 触发器计数”，然后在 值”字段中键入 <b>1</b>”</li> <li>• 单击 属性”选项卡</li> <li>• 用dummyProperty改写属性；在 值”字段中，单击 <input type="button" value="..."/> 按钮并单击 常量”和 数字”，然后在 常量数字”字段中键入 <b>0</b>”</li> </ul>

	使用此属性，属性跟踪”对话框将能够在模拟期间显示序列的元素。
延迟15分钟	<ul style="list-style-type: none"> <li>在左侧的元素列表中，展开 IntermediateEvent”组，然后单击 Delay15Minutes并在 控件”选项卡上</li> <li>单击 新参数”下拉箭头并选择 InterTriggerTimer”，然后将 值”字段设置为 15 分钟 (“000:000:000 000:15:00”)</li> </ul>
任务10分钟	<ul style="list-style-type: none"> <li>在对话框左侧的元素列表中，展开 活动”组，然后单击Task10Minutes和时间”选项卡</li> <li>单击 新参数”下拉箭头并选择 处理时间”，然后将 值”字段设置为 10 分钟 (“000:000:000 000:10:00”)</li> </ul>
任务20分钟	<ul style="list-style-type: none"> <li>在对话框左侧的元素列表中，展开 活动”组，然后单击Task20Minutes和时间”选项卡</li> <li>单击 新参数”下拉箭头并选择 处理时间”，然后将 值”字段设置为 20 分钟 (“000:000:000 000:20:00”)</li> </ul>

### 运行仿真

- 在 配置 BPSim”对话框工具 上，单击 运行”图标打开 运行仿真控制器”对话框
- 点击 运行”图标下拉箭头并选择 标准仿真”
- 仿真完成后，单击工具栏上的  按钮，显示 BPSim PropertyParameter Values”对话框
- 单击查询按钮和 按属性分组”选项卡



### 分析

分叉并行网关将同时激活传出序列流 ( 顺序未定义且不重要 )。但是，我们希望任务的顺序是准确的：

- 任务2
- 任务1

- 任务3

此顺序由在两个活动 (ProcessingTime) 和计时器中间事件(InterTriggerTimer) 上设置的 BPSim 参数确定。“ 列 PropertyParameter Values”对话框中显示的 列确认Task2领先于Task1 ，后者领先于Task3 。



## 画墙进程仿真(调用活动)

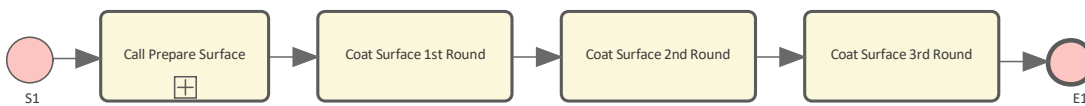
这是一个模拟墙壁粉刷过程的简单示例。我们将主要过程定义为准备表面，然后将其绘制 3 次。准备表面进一步分为打磨和清洁等任务。

我们假设涂上三层油漆中的每一层都是相同的过程，只是在每层油漆上随机花费的时间可能不同。

### 创建 BPMN 模型

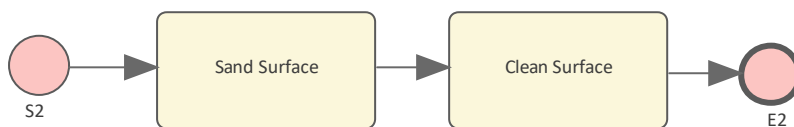
该模拟对两个过程进行。

#### 主要流程 - Paint Wall 进程



1. 创建一个名为S1的开始事件。
2. 将一个序列流添加到一个名为调用Prepare Surface 的目标调用调用。
3. 将序列流添加到名为Coat Surface 1st Round 的目标 callGlobalTaskActivity。
4. 将一个序列流添加到名为Coat Surface 2nd Round 的目标 callGlobalTaskActivity。
5. 将一个序列流添加到名为Coat Surface 3rd Round 的目标 callGlobalTaskActivity。
6. 将序列流添加到名为E1的目标结束事件。

#### 重用流程进程Surface流程





1. 创建一个名为S2的开始事件。
2. 将一个序列流添加到一个名为Sand Surface的目标抽象任务中。
3. 将一个序列流添加到一个称为清理表面的目标抽象任务。
4. 将序列流添加到名为E2的目标结束事件。

#### 设置任务用流程来调用活动

1. 创建一个名为Coat Surface的全局任务活动。
2. 双击Coat Surface 1st Round、Coat Surface 2nd Round和Coat Surface 3rd Round中的每一个，并将标签 称为 ActivityRef" 设置为Coat Surface。  
提示：也可以从浏览器窗口中将全局任务 涂层表面"拖放到调用活动元素上，单击上下文菜单上的 设置调用活动引用"选项。
3. 双击调用Prepare，设置 调用ActivityRef" 标签为Prepare Surface进程。  
提示：您也可以从浏览器窗口中将 Prepare Surface进程元素"进程拖放到调用活动中，单击上下文菜单上的 "Set调用"选项。


### 配置 BPSim

1. 打开 配置 BPSim "对话框 ( 仿真>流程分析>仿真>进程BPSim 管理器" ) 。
2. 点击  图标仿真创建一个业务流程仿真管理名工件名

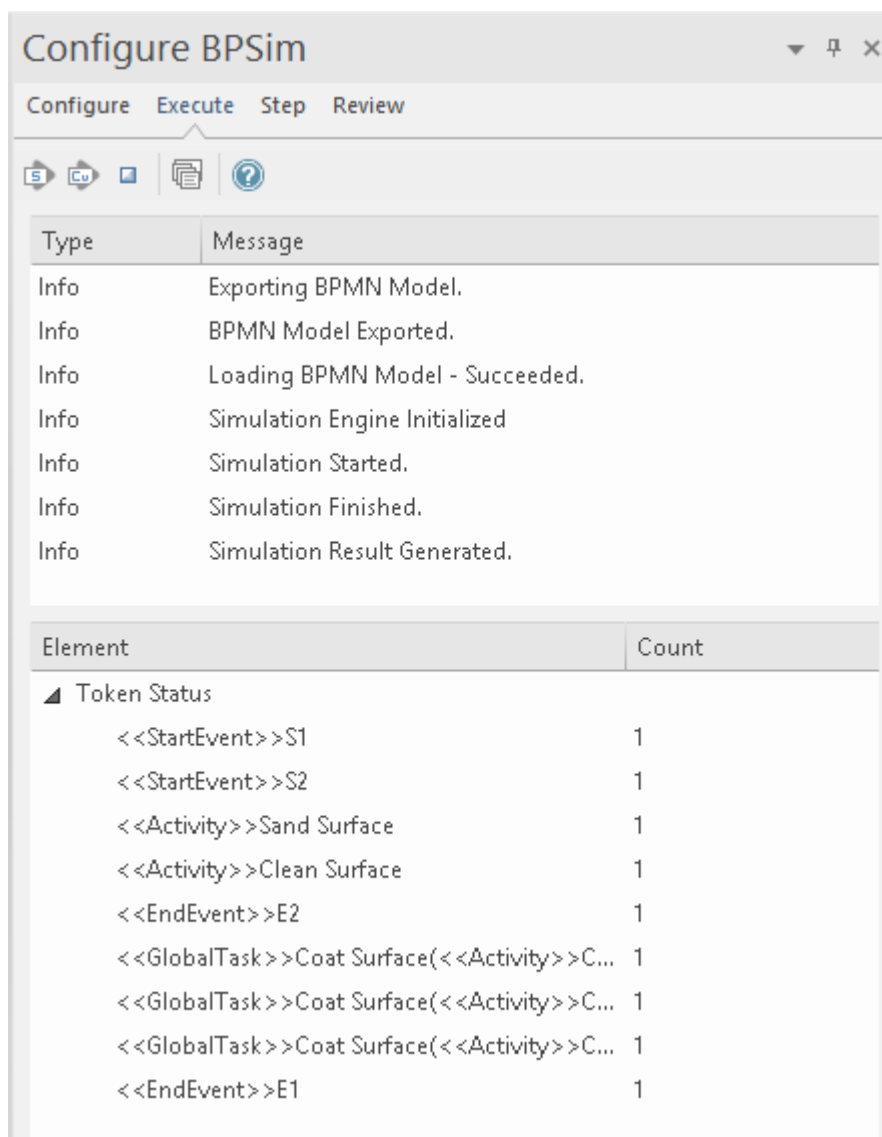
3. 单击  图标并选择包含相应 BPMN 2.0模型的包。

物件	活动
固定缩放时间	<ol style="list-style-type: none"> <li>1. 打开“Prepare Surface”进程图。</li> <li>2. 单击活动沙面，在配置 BPSim窗口中，单击新参数下拉箭头并创建一个名为“处理时间”的时间参数。</li> <li>3. 在“值”字段中，将设置更改为 0 00:30:00（即 30 分钟）。</li> <li>4. 单击活动清理，然后在配置 BPSim窗口中，单击新参数下拉箭头并创建一个名为“处理时间”的时间。</li> <li>5. 在“值”字段中，将设置更改为 0 00:10:00（即 10 分钟）。</li> <li>6. 单击  图标。</li> </ol>
随机涂层时间	<ol style="list-style-type: none"> <li>1. 在浏览器窗口中，单击全局任务活动涂层表面。</li> <li>2. 在配置 BPSim窗口中，单击新参数下拉箭头并创建一个名为“处理时间”的时间参数。</li> <li>3. 在“值”字段中单击  按钮，在参数对话框中单击“分布”选项卡和“泊松”。</li> <li>4. 在“平均值”字段中输入“10”，然后点击确定按钮。</li> <li>5. 单击  图标。</li> </ol> <p>使用此设置，泊松分布生成的随机数的平均值为 10。如果您愿意，可以选择其他类型的分布。</p>
S1 上的触发计数	<p>在“Paint Wall”进程图中，单击开始事件 S1。</p> <ol style="list-style-type: none"> <li>1. 在配置 BPSim窗口中，单击参数parameter 下拉箭头并创建一个名为“控件”的控件参数。</li> <li>2. 在“值”字段中输入“1”。</li> <li>3. 单击  图标。</li> </ol>

## 运行仿真

1. 在配置 BPSim窗口的“执行”选项卡上，单击  图标。

当模拟完成时，它会提供类似于以下的结果：



### 流量分析

对于在S1上启动的唯一令牌，我们可以从配置 BPSim窗口的“执行”选项卡中看到流程是如何发展的：

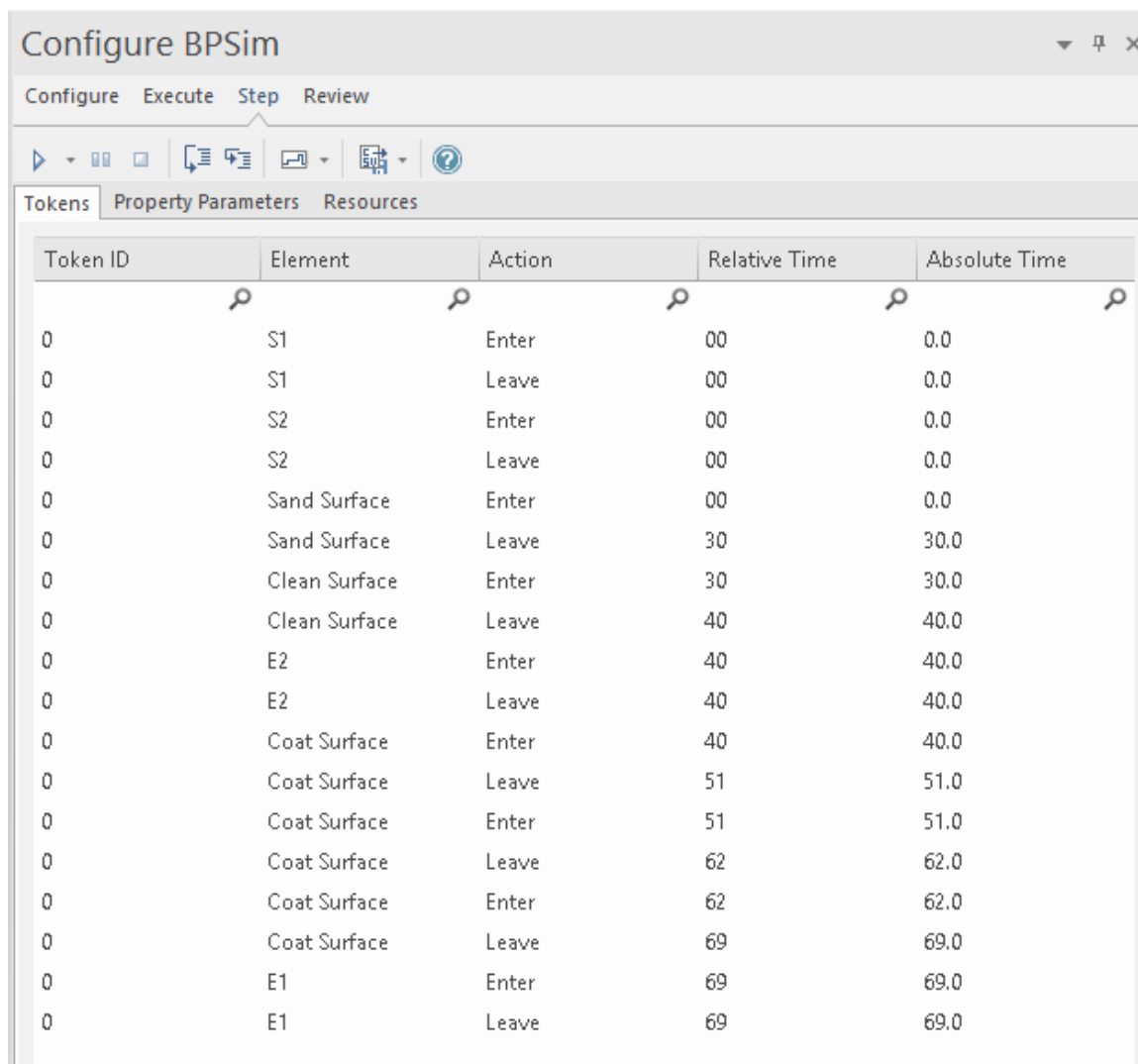
Element	Count
▲ Token Status	
<<StartEvent>>S1	1
<<StartEvent>>S2	1
<<Activity>>Sand Surface	1
<<Activity>>Clean Surface	1
<<EndEvent>>E2	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 1st Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 2nd Round)	1
<<GlobalTask>>Coat Surface(<<Activity>>Coat Surface 3rd Round)	1
<<EndEvent>>E1	1

- 当到达callProcessActivity时，被调用的进程被激活；所以我们有S2 ~ E2

- 当到达 `callGlobalTaskActivity` 时，被调用的任务被激活 - 符号为：`GlobalTask name`（被称为活动名称）；  
全球 `Coat Surface` 被调用了 3 次：
  - 涂层表面（涂层表面第一轮）
  - 涂层表面（涂层表面第二轮）
  - 涂层表面（涂层表面第三轮）

### 时间分析

单击配置 BPSim 窗口的“步骤”选项卡，然后单击类似于下图的“令牌”选项卡：



Token ID	Element	Action	Relative Time	Absolute Time
0	S1	Enter	00	0.0
0	S1	Leave	00	0.0
0	S2	Enter	00	0.0
0	S2	Leave	00	0.0
0	Sand Surface	Enter	00	0.0
0	Sand Surface	Leave	30	30.0
0	Clean Surface	Enter	30	30.0
0	Clean Surface	Leave	40	40.0
0	E2	Enter	40	40.0
0	E2	Leave	40	40.0
0	Coat Surface	Enter	40	40.0
0	Coat Surface	Leave	51	51.0
0	Coat Surface	Enter	51	51.0
0	Coat Surface	Leave	62	62.0
0	Coat Surface	Enter	62	62.0
0	Coat Surface	Leave	69	69.0
0	E1	Enter	69	69.0
0	E1	Leave	69	69.0

您可以按原样检查列表中的时间，但为了使过程更容易，在“行动”列的过滤器栏字段中键入“离开”以仅显示该列中包含该文本string的记录。

Tokens					
Property Parameters					
Resources					
Token ID	Element	Action	Relative Time	Absolute Time	
		Leave			
0	S1	Leave	00	0.0	
0	S2	Leave	00	0.0	
0	Sand Surface	Leave	30	30.0	
0	Clean Surface	Leave	40	40.0	
0	E2	Leave	40	40.0	
0	Coat Surface	Leave	51	51.0	
0	Coat Surface	Leave	62	62.0	
0	Coat Surface	Leave	69	69.0	
0	E1	Leave	69	69.0	



报告显示如图所示，我们可以进行这样的分析：

- 调用活动调用准备表面耗时 40 分钟，由沙面（30 分钟）和清理表面（10 分钟）组成，定义为
- 涂层表面（第一轮）耗时 11 分钟；涂层表面（第 2 轮）耗时 11 分钟；涂层表面（第 3 轮）耗时 7 分钟 - 数字 11、11、7 由 Poisson(10) 分布随机生成；这里重要的是任务的每个调用实例都有自己的值
- 涂层表面从所有实例中收集的总时间： $11 + 11 + 7 = 29$
- *Paint Wall* 进程的求和处理时间为 69 分钟，由四个调用活动组成： $40 + 11 + 11 + 7 = 69$

## 定制仿真

我们可以在 BPMN 元素上配置一个“结果请求”来自定义模拟报告，以便我们只报告我们感兴趣的参数。

### 配置结果请求

1. 在 *Paint Wall*“进程图中，单击活动 *Coat Surface 1st Round*”。
  2. 在配置 BPSim 窗口中，单击新参数下拉箭头并创建一个名为 *ProcessingTime*“的时间参数。
  3. 单击  工具栏图标。 “结果请求”列显示在“参数”列的右侧；单击下拉箭头并选择“总和”复选框。单击确定按钮。
  4. 在“值”字段中输入“1”。
  5. 单击  图标。
  6. 重复步骤 1 到 5 进行活动调用准备表面、涂层表面第 2 轮、涂层表面第 3 轮
- 展开 *Paint* 进程流程组的“业务流程”并重复这些步骤


### 运行仿真

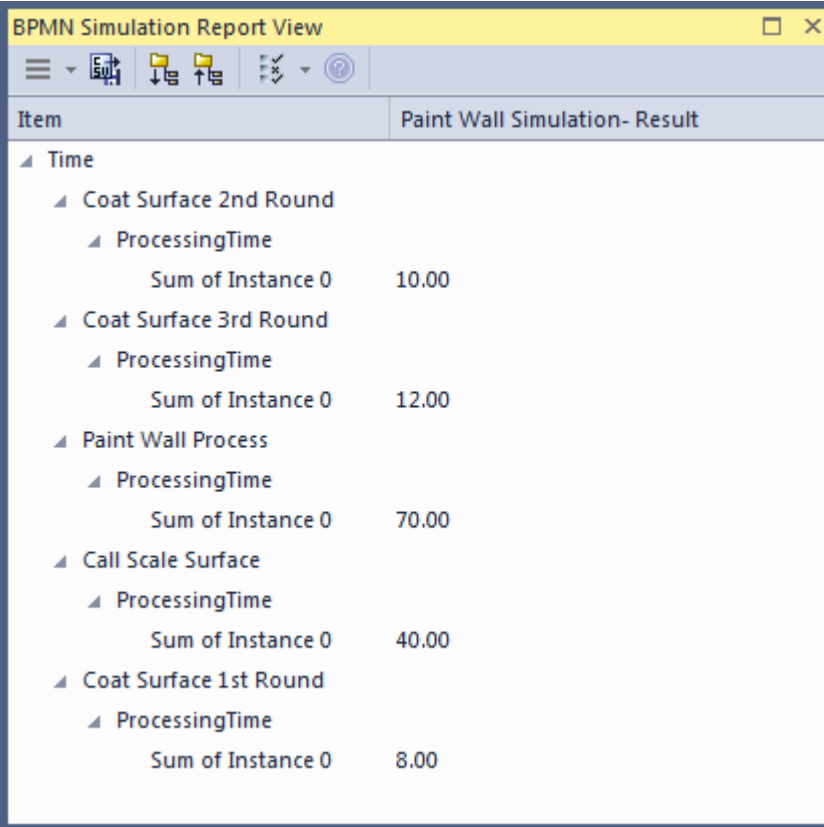
- 在配置 BPSim“对话框工具”上，点击运行按钮；“仿真仿真控制器”对话框显示
- 单击运行按钮下拉箭头，在本例中选择“定制仿真”

### 流量分析

流量分析与标准仿真完全相同。

### 时间分析

在 仿真仿真  控制器；“BPMN仿真报告视图”显示。



The screenshot shows a window titled "BPMN Simulation Report View" with a toolbar and a table of simulation results. The table has two columns: "Item" and "Paint Wall Simulation- Result". The data is as follows:

Item	Paint Wall Simulation- Result
Time	
Coat Surface 2nd Round	
ProcessingTime	
Sum of Instance 0	10.00
Coat Surface 3rd Round	
ProcessingTime	
Sum of Instance 0	12.00
Paint Wall Process	
ProcessingTime	
Sum of Instance 0	70.00
Call Scale Surface	
ProcessingTime	
Sum of Instance 0	40.00
Coat Surface 1st Round	
ProcessingTime	
Sum of Instance 0	8.00

时间分析与标准仿真相同；但是，该报告仅包含我们要求的 总和”结果。

注记：目前，在时间分析中，我们不能在被调用进程本身或被调用进程包含的活动上请求ProcessingTime。如果您有此要求，请使用标准仿真。

## BPSim 成本参数

BPSim 1.0 提供了设置成本参数和从过程模拟实验接收成本统计数据的方法。BPSim 提供了一个框架来根据两个参数确定可变成本，这两个参数都与模拟过程中执行的活动水平有关。这些参数是：

- 完成成本 ( BPSim 规范中的 “固定成本”) - 每次操作完成时产生的成本；该成本可以与任务、进程、子流程、调用活动或资源元素相关
- 时间成本 ( BPSim 规范中的 “单位成本”) ——任务、流程、子流程、调用活动或资源在一段时间内忙碌时产生的成本

活动、资源和流程支持成本参数。上：

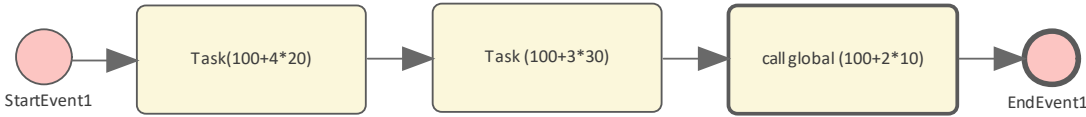
- 活动结束后都会产生活动、完成成本和时间成本 ( 单位成本 \*时间 )
- 每当每个涉及的资源完成A活动时，都会产生资源、完成成本和时间成本
- A进程，完成成本和时间都是在一个进程完成的时候产生的

BPSim 不支持那些无需模拟即可知道的成本——例如，总体劳动力雇佣成本。

成本参数的配置和模拟通过两个例子来演示：

- [Set Cost Parameters on Activity](#)
- [Set Cost Parameters on Resource](#)

# 在活动中设置成本参数



## 创建 BPMN模型 (活动)

1. 在浏览器窗口中，创建一个 *StartEvent1*、一个 *GlobalTask1*、两个 *AbstractTask* 和一个 *EndEvent1*。
2. Ctrl+拖动浏览器窗口中的元素到图表上，将 *GlobalTask1* 粘贴为调用（调用活动），称为 *call global (100+2\*10)*。
3. 给元素命名并将它们与序列流连接起来；应该调用两个 *AbstractTasks*：
  - 任务 (  $100+3*30$  ) 和
  - 任务 (  $100+4*20$  )。

## 配置



在图中创建业务流程仿真配置配置 BPSim 工件选项。设置配置以链接到包含 BPMN 模型元素的包，并按照指示配置这些 BPSim 参数。

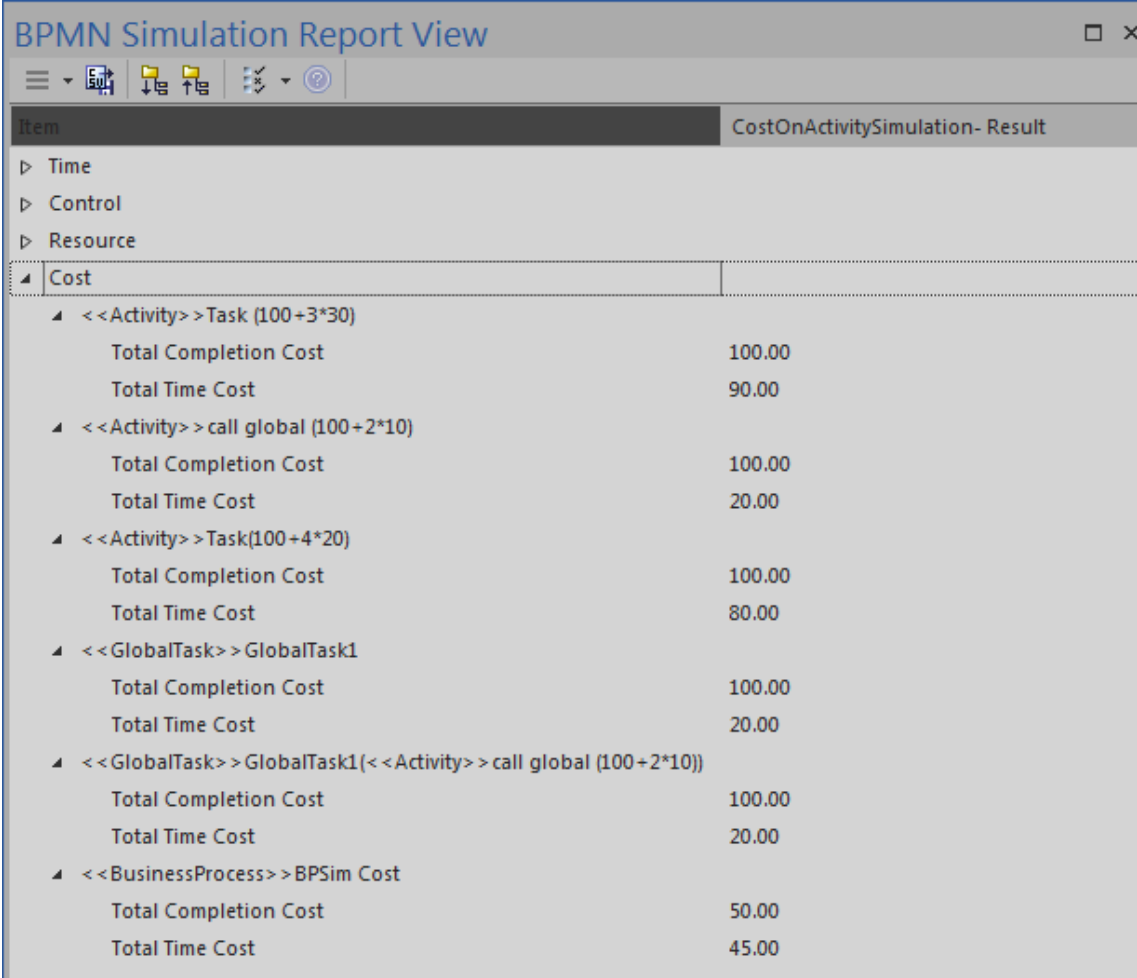
参数	设置
场景参数	<ol style="list-style-type: none"> <li>1. 点击配置工件和参数，对于 Scenario "TimeUnit" 时间点击 'Value' 下拉箭头 'hours'。</li> <li>2. 在 持续时间"参数的 值"字段中，将值设置为 0001 00:00:00" ( 1天 )。此时间单位用于计算时间成本 ( 时间成本 = 单位成本 * 时间 )，因此请确保单位成本基于正确的时间单位。</li> </ol>
控件参数	<ol style="list-style-type: none"> <li>1. 在图表上，单击 <i>StartEvent1</i>。</li> <li>2. 单击新参数下拉箭头并选择 控件"。</li> <li>3. 在 参数"字段中，单击下拉箭头并选择 TriggerCount"。</li> <li>4. 在 值"字段中输入 1"。</li> </ol>
时间参数	<ol style="list-style-type: none"> <li>1. 在图表上单击任务 ( <math>100+4*20</math> )。</li> <li>2. 单击新参数下拉箭头并选择 时间"。</li> <li>3. 单击 参数"下拉箭头并选择 ProcessingTime"。</li> <li>4. 在 值"字段中，将值设置为 000:000:000 004:00:00" ( 4 小时 )。</li> <li>5. 单击图表上的任务 ( <math>100+3*30</math> ) 并重复步骤 2、3 和 4，将 值"字段设置为 000:000:000 003:00:00" ( 3 小时 )。</li> <li>6. 单击图表上的 <i>GlobalTask1</i> 并重复步骤 2、3 和 4，将 值"字段设置为 "000:000:000 002:00:00" ( 2 小时 )。</li> </ol>
成本参数	<ol style="list-style-type: none"> <li>1. 在图表上，单击任务 ( <math>100+4*20</math> )。</li> <li>2. 单击新参数下拉箭头并选择 成本"。</li> <li>3. 在 参数"字段中单击下拉箭头并依次选择：                     <ul style="list-style-type: none"> <li>- "FixedCost"，然后在 Value"字段中单击  按钮，选择</li> </ul> </li> </ol>



	<p>常量”选项卡和“浮动”，以及“常量浮动”字段输入“100”；点击确定按钮</p> <p>- 'UnitCost' - 做同样的事情，将'Constant Floating' 字段设置为'20'。</p> <p>4. 在图表上，单击任务 (<math>100+3*30</math>) 并重复步骤 2 和 3，设置：</p> <p>- '固定成本' 到 '100'</p> <p>- “单位成本”到 '30'。</p> <p>5. 在图表上，单击GlobalTask1并重复步骤 2 和 3，设置：</p> <p>- '固定成本' 到 '100'</p> <p>- 'UnitCost' 到 '10'。</p> <p>6. 在图表上，单击BPSim Cost并重复步骤 2 和 3，设置：</p> <p>- '固定成本' 到 '50'</p> <p>- 'UnitCost' 到 '5'。</p>
--	--

## 仿真

1. 在“配置 BPSim”对话框中，单击“执行”选项卡。
2. 单击  按钮。
3. 模拟完成后，单击“审阅”选项卡和“标准结果报告”选项卡。
4. 过滤器通过单击  按钮并选择“仅显示非空项”选项来过滤报告。



**BPMN Simulation Report View**

Item	CostOnActivitySimulation- Result
▶ Time	
▶ Control	
▶ Resource	
▲ Cost	
▲ <<Activity>>Task (100+3*30)	
Total Completion Cost	100.00
Total Time Cost	90.00
▲ <<Activity>>call global (100+2*10)	
Total Completion Cost	100.00
Total Time Cost	20.00
▲ <<Activity>>Task(100+4*20)	
Total Completion Cost	100.00
Total Time Cost	80.00
▲ <<GlobalTask>>GlobalTask1	
Total Completion Cost	100.00
Total Time Cost	20.00
▲ <<GlobalTask>>GlobalTask1(<<Activity>>call global (100+2*10))	
Total Completion Cost	100.00
Total Time Cost	20.00
▲ <<BusinessProcess>>BPSim Cost	
Total Completion Cost	50.00
Total Time Cost	45.00

## 分析

活动	分析
任务(100+4*20)	<ul style="list-style-type: none"><li>• Total Completion Cost 为 100 · 与 BPSim 中的 FixedCost (100) 设置相匹配</li><li>• 时间为 80 · 计算方式为 ProcessingTime (4 hours) * UnitCost (20/hour)</li></ul>
任务(100+3*30)	<ul style="list-style-type: none"><li>• Total Completion Cost 为 100 · 与 BPSim 中的 FixedCost (100) 设置相匹配</li><li>• 时间为 90 · 计算为 ProcessingTime (3 hours) * UnitCost (30/hour)</li></ul>
调用全局 ( 100+2*10 )	<ul style="list-style-type: none"><li>• Total Completion Cost 为 100 · 与 BPSim 中 <i>GlobalTask 1</i> 设置的 FixedCost (100) 匹配</li><li>• 时间为 20 · 计算为 GlobalTask1 上的 ProcessingTime (2 hours) * <i>UnitCost</i> (10/hour)</li></ul>
BPSim 成本流程	<ul style="list-style-type: none"><li>• Total Completion Cost 为 50 · 与 BPSim 中的 FixedCost (50) 设置相匹配</li><li>• 时间为 45 · 计算为所有任务的 Total ProcessingTime (4 + 3 + 2 = 9 hours) * UnitCost (5/hour)</li></ul>

# 在资源上设置成本参数





## 创建BPMN模型(资源)

1. 在浏览器窗口中创建一个 *StartEvent1* 、一个 *GlobalTask1* 、两个名为 *Task* 任务 (*by Junior*) 和 *Task* (*by Senior*) 任务的 *abstractTask* 以及一个 *EndEvent1* 。
2. Ctrl+ 将浏览器窗口中的元素拖到图表上，将 *GlobalTask1* 粘贴为名为 *call global (by Junior)* 的 *Invocation* (调用活动)。
3. 将元素与序列流连接起来。
4. 创建两个 BPMN2.0 资源元素：初级开发人员和高级开发人员。



## 配置

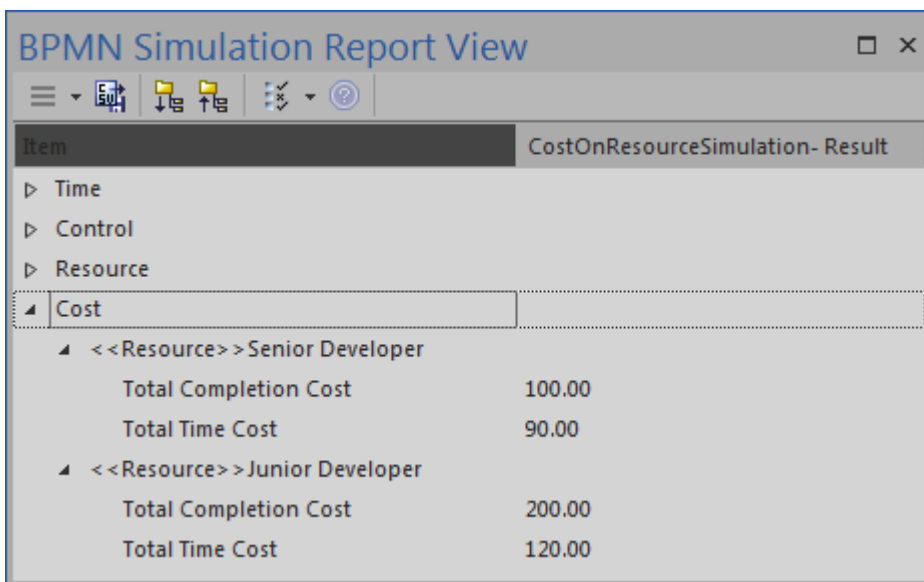
在图中创建业务流程仿真配置 BPSim 工件选项，然后将配置设置为链接到包含 BPMN 元素的包，并按照指示配置这些模型参数。

参数	环境
场景参数	<ol style="list-style-type: none"> <li>1. 点击配置工件和参数，对于 Scenario "TimeUnit" 时间点击 'Value' 下拉箭头 'hours' 。</li> <li>2. 在 "持续时间" 参数的 "值" 字段中，将值设置为 "001 00:00:00" ( 1天 ) 。此时间单位用于计算时间成本 ( 时间成本 = 单位成本 * 时间 ) ，因此请确保单位成本基于正确的时间单位。</li> </ol>
控件参数	<ol style="list-style-type: none"> <li>1. 在图表上，单击 <i>StartEvent1</i> 。</li> <li>2. 单击新参数下拉箭头并选择 "控件" 。</li> <li>3. 在 "参数" 字段中，单击下拉箭头并选择 "TriggerCount" 。</li> <li>4. 在 "值" 字段中输入 "1" 。</li> </ol>
时间参数	<ol style="list-style-type: none"> <li>1. 在图表上单击任务 (<i>by Junior</i>) 。</li> <li>2. 单击新参数下拉箭头并选择 "时间" 。</li> <li>3. 单击 "参数" 下拉箭头并选择 "ProcessingTime" 。</li> <li>4. 在 "值" 字段中，将值设置为 "000:000:000 004:00:00" ( 4 小时 ) 。</li> <li>5. 单击图表上的任务 ( 按高级 ) 并重复步骤 2、3 和 4，将 "值" 字段设置为 "000:000:000 003:00:00" ( 3 小时 ) 。</li> <li>6. 单击图表上的 <i>GlobalTask1</i> 并重复步骤 2、3 和 4，将 "值" 字段设置为 "000:000:000 002:00:00" ( 2 小时 ) 。</li> </ol>
资源参数	<ol style="list-style-type: none"> <li>1. 在图表上，单击初级开发人员资源。</li> <li>2. 单击新参数下拉箭头并选择 "资源" 。</li> <li>3. 单击 "参数" 下拉箭头并选择 "选择" 。</li> </ol>

	<ol style="list-style-type: none"> <li>在“值”字段中，单击  按钮以打开“编辑资源选择”对话框。</li> <li>单击“初级开发人员”，然后单击“按资源添加选择”按钮将选择移动到“资源或角色”面板。</li> <li>“需要数量”列默认为“1”；用“10”改写这个值。</li> <li>单击 AND 单选按钮以设置逻辑关系；资源选择的最终表达式组成并显示在文本字段中。</li> <li>单击确定按钮返回到配置 BPSim 确定，其中表达式显示在“值”字段中。</li> <li>单击高级开发人员资源并重复步骤 2 到 8，在“需要数量”字段中键入“5”。</li> </ol>
成本参数	<ol style="list-style-type: none"> <li>在图表上单击 <i>Junior Developer</i>。</li> <li>单击新参数下拉箭头并选择“资源”。</li> <li>单击“参数”下拉箭头并依次选择： <ul style="list-style-type: none"> <li>- “FixedCost”，然后在“Value”字段中单击  按钮，选择“常量”选项卡和“浮动”，然后在“常量浮动”中字段类型“100”和“货币单位”字段类型“AUD”；点击确定按钮</li> <li>- 'UnitCost' - 做同样的事情，将'Constant Floating' 字段设置为'20'。</li> </ul> </li> <li>在图表上单击高级开发人员并重复步骤 2 和 3，设置： <ul style="list-style-type: none"> <li>- '固定成本' 到 '100'</li> <li>- “单位成本”到“30”。</li> </ul> </li> </ol>

### 仿真

- 在“配置 BPSim”对话框中，单击“执行”选项卡。
- 单击  按钮。
- 模拟完成后，单击“审阅”选项卡和“标准结果报告”选项卡。
- 过滤器通过单击  按钮并选择“仅显示非空项”选项来过滤报告。




## 分析

资源	结果
初级开发人员	<ul style="list-style-type: none"><li>总完成成本为 200” · 计算方式为 FixedCost (100) * 涉及的活动数量 (2)</li><li>时间为 120” · 计算为 ProcessingTime (4 + 2 = 6 hours) * UnitCost (20/hour)</li></ul>
高级开发人员	<ul style="list-style-type: none"><li>总完成成本为 100” · 计算方式为 FixedCost (100) * 涉及的活动数量 (1)</li><li>时间为 90' · 计算为 ProcessingTime (3 hours) * UnitCost (30/hour)</li></ul>

## 导出一个导出配置


当您在模型中定义了配置后，您可以将其导出为 XMI 文件以导入到其他项目中。配置所基于的BPMN 2.0模型也随配置一起导出。当您把 XMI 文件导入另一个项目时，模型将绑定到相应的配置。

### 访问

上下文菜单	在图表上点击浏览器窗口或业务流程仿真管理窗口或右键-导出工件配置
其它	配置 BPSim窗口工具栏  导出icon

### 发布模型包

导出 BPSim 配置及其模型的过程使用“发布模型包”对话框将模型发布到 XMI 文件。

选项	描述
包	默认为包含业务流程仿真工件包的名称。
文件名	类型输入或浏览（单击  图标）要导出模型的文件路径和 XML 文件名。
XML类型	选择“BPMN 2.0 XML”。
导出	单击此按钮可导出配置和 BPMN 2.0模型。当“进度”字段中显示确认消息时，导出完成。
格式XML输出	默认为选中；离开选中。
视图XML	如果要检查导出的 XML，请单击此按钮。

### 注记

- 要将模型从 XMI 导入到新项目中，请在新项目中选择目标包，然后选择“发布>模型交换>导入>本地文件”或“导入>XMI文件”功能区选项

