

SIMIO系统仿真培训



无锡迅合信息科技有限公司



(0510) 82132584



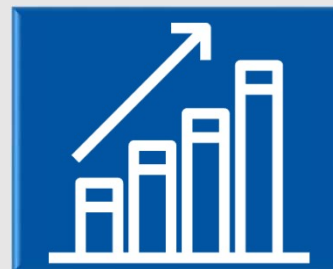
无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

目录

- 01 系统仿真简介
- 02 Simio介绍和基本动画
- 03 Simio建模框架
- 04 固定对象
- 05 动态对象移动
- 06 模型数据
- 07 使用过程扩展模型逻辑
- 08 解释仿真结果
- 09 建立对象定义
- 10 计划和调度总览
- 11 分析输入数据



第一章



系统仿真技术简介



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

何为仿真？

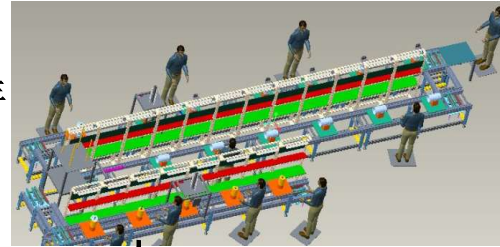


- ▶ 我们的目标是通过使用模型来模拟真实系统的行为，模型行为和实际系统行为相似。
- ▶ 随着时间轴的展开，模拟真实世界的过程和系统运作规律。
- ▶ 仅仅为了实现项目的目标提供“足够”的逼真度。

何为仿真?



流程数字孪生



亚原子世界

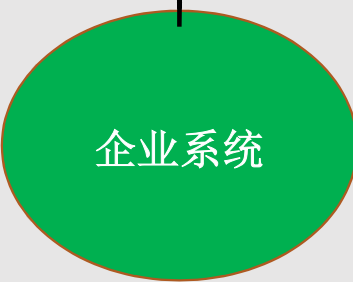
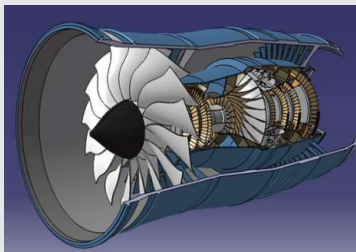
量子系统



电子/机械系统

连续系统

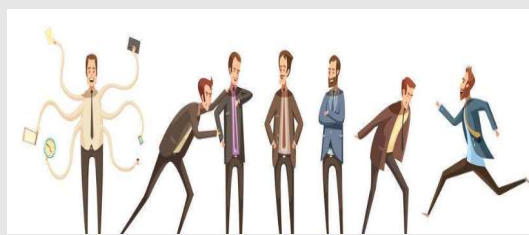
- 有限元仿真
- 物理/机械/化学行为
- 流体动力学
- 物料科学
- 设备设计



企业系统

离散事件系统

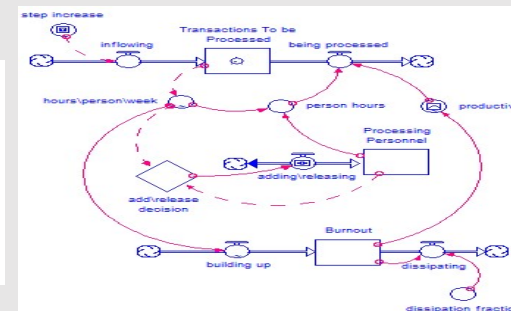
- 人造流程/协议的仿真模拟
- 离散化的决策
- 事件驱动的决策程序



社会经济/生态系统

连续系统

- 系统动力学
- 因果环分析/正负反馈
- 大规模系统工程



宇宙论系统

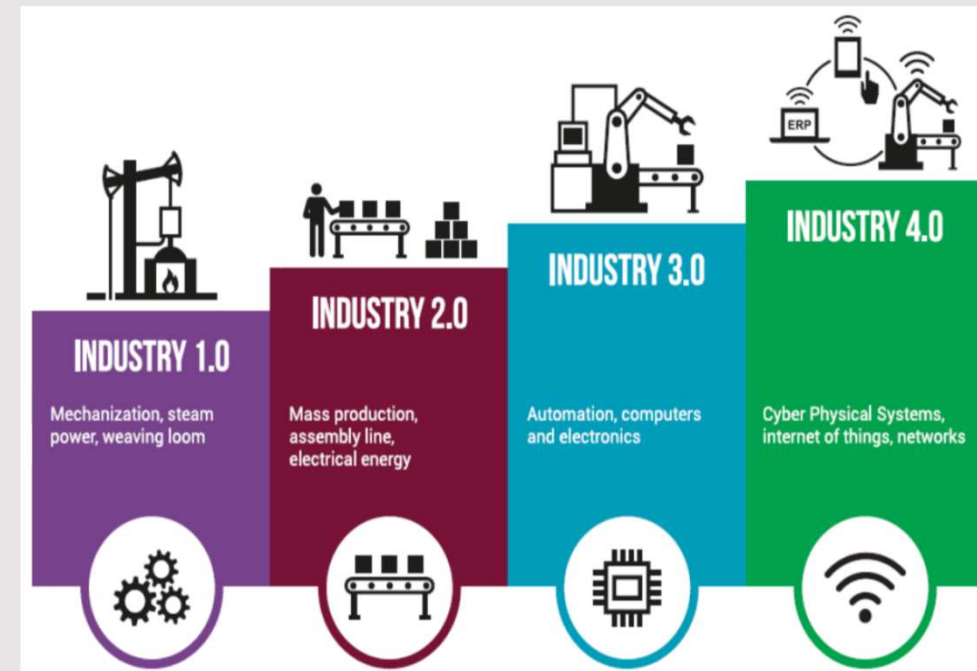
量子系统

仿真模型实例 (Simio)



智慧工厂和工业4.0

- ▶ 万物互联和自动化生产系统的大趋势
- ▶ 四大设计原则：
 - ▶ 组件和数据的协同能力 (IoT)
 - ▶ 信息透明
 - ▶ 支持系统能帮助解决问题
 - ▶ 系统组件能够自主地执行
- ▶ 系统仿真技术，尤其是基于风险的计划和调度技术是支持工业演进的关键技术。



图片取自于书籍《Deliver on Your Promise, How Simulation-based Scheduling Will Change Your Business》 《交付承诺：基于仿真的调度技术如何变革您的业务》

为什么要仿真？

▶ 对象可以相互影响。



▶ 所有系统都有随机性、机器失效、人员生病、迟到等等。



▶ 以上的组合导致系统复杂性。



+



=



▶ 系统仿真在管控系统复杂性方面，独一无二。

典型的应用实例

门诊手术中心的流程改进

设计/改进一个加工中心（分销中心）

评估应急中心的灾难应急计划

设计/改进一个游乐园的旅客流

为医院设计新的CT扫描区域

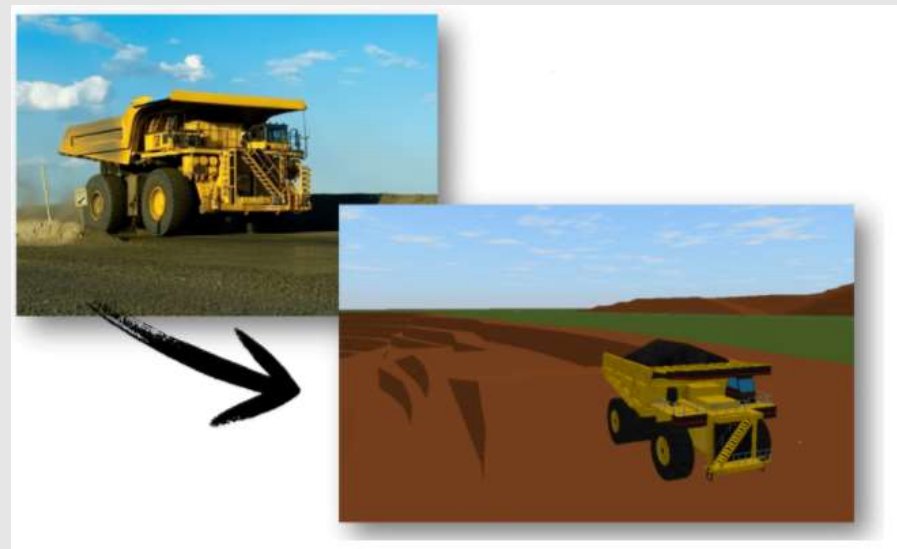
设计/改进汽车装配线

评价配套科室的利用率和改进措施

设计/改进银行的后台服务流程

设计/改进机场的包裹处理系统

地下采矿系统的最佳设计和可视化



仿真用于系统化设计

- ▶ 系统仿真模型传统上用于改进系统的设计
 - ▶ 我如何改进我的系统，以改善长期的绩效水平？
- ▶ 模型主要用于：
 - ▶ 比较不同的系统方案
 - ▶ 优化系统参数
 - ▶ 预测绩效



显著的收益

- ▶ 预见未来
- ▶ 对系统过程文档化
- ▶ 回答“假设性问题”
- ▶ 在模型中，较早发现错误
- ▶ 可视化流程
- ▶ 沟通效率
- ▶ 改善绩效



仿真的关键优势

▶ 灵活的模型

- ▶ 绝大多数系统有独有的特征，他们是系统绩效的关键约束。

▶ 随机性

- ▶ 绝大多数真实系统具有对系统绩效产生关键影响的变化值。

▶ 可视化

- ▶ 三维动画展示是强大的沟通工具。

仿真的灵活性



一些应用领域

制造

物流，供应链和分销

建筑工程和项目管理

军事作业

运输方式和交通

商业流程

医疗卫生系统

机场和大规模中转系统

变动的影响

绝大多数系统存在变动。

- 客户需求, 产品数量
- 设备失效/人员缺勤
- 物料的短缺

随机性是绝大多数系统的重要特征。

动态过程无法用静态工具来分析。



系统变化性有关系吗？

设计2小时的数码冲印店

到达间隔平均
为60分钟



确保“2小时”
相片处理



平均服务时间是
55分钟

系统的表现如何？
24x7模式下顾客的等待时间？

到达间隔

服务时间

等待时间？



常量
随机*
随机*

常量
常量
随机*

我们来模拟
一下...

* 指数分布

管理变动

- ▶ 我们可以通过减少变动大大地提高系统的表现：
 - ▶ 计划工作和预约
 - ▶ 重新设计/改进流程使其变的更加一致
 - ▶ 通过信息来平衡实体流动
- ▶ 我们必须在模型中精确地体现变动
- ▶ 相对于平均值，变动的程度对系统行为造成影响更大

利益干系人目标

利益干系人（*Stakeholder*）是特指那些对仿真项目提出申请、支付费用、耗用资源，以及受仿真项目及结果影响的那部分人群。

- 在不同的利益干系人之间有目标冲突是很常见的。

对于系统而言，没有“唯一的仿真模型”之说——模型是否“正确”，取决于系统和研究目标的组合：

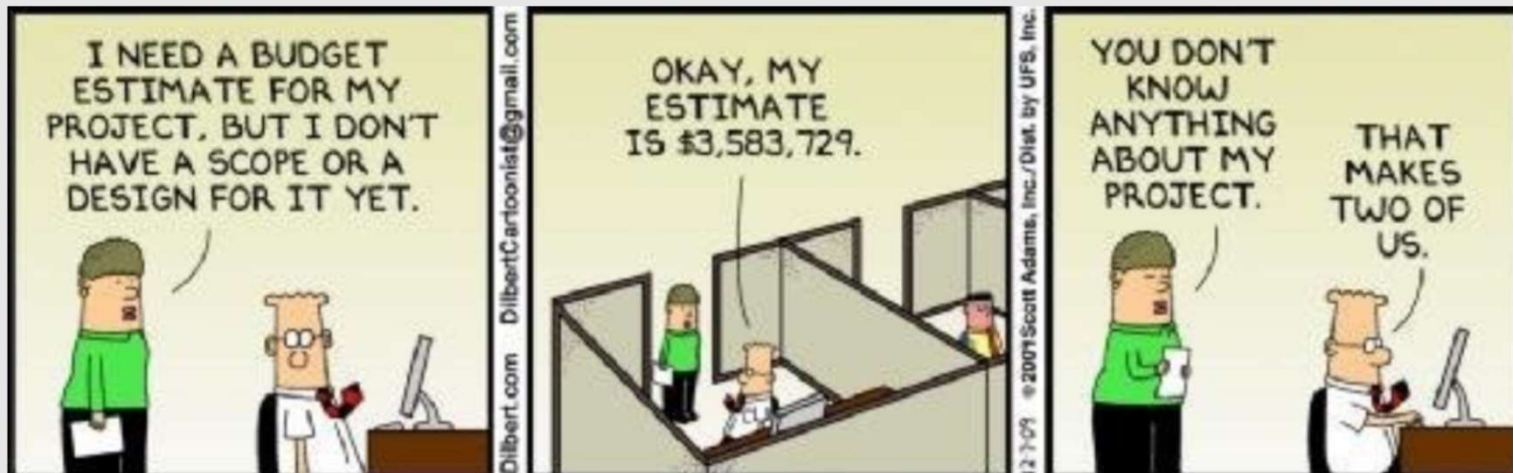
- 您想评估什么，学习什么，你希望证明什么？
- 项目的边界和范围是什么？
- 有哪些现成的数据，或者哪些可采集的数据？
- 你希望的结果是什么形式的？

功能规格书

“如果你不知道你要去哪里，你如何知道已经到达那里？”

木匠的忠告：“三思而后行”。

- ▶ *功能规格书* – 描述了项目的交付物、交付时间、交付形式和交付人的详细项目文档。
- ▶ 这个项目的工期是多久？



功能规格书

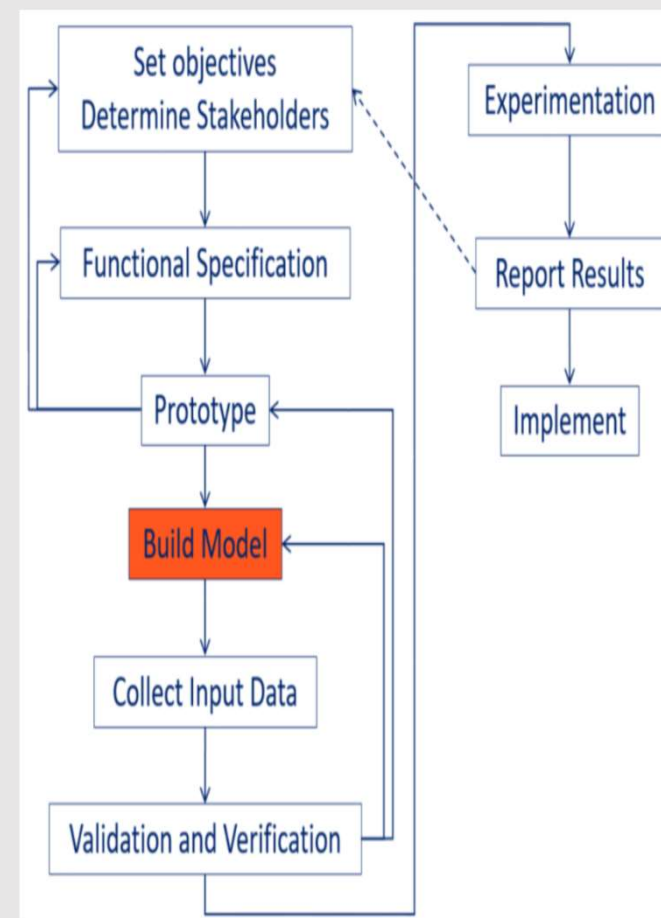
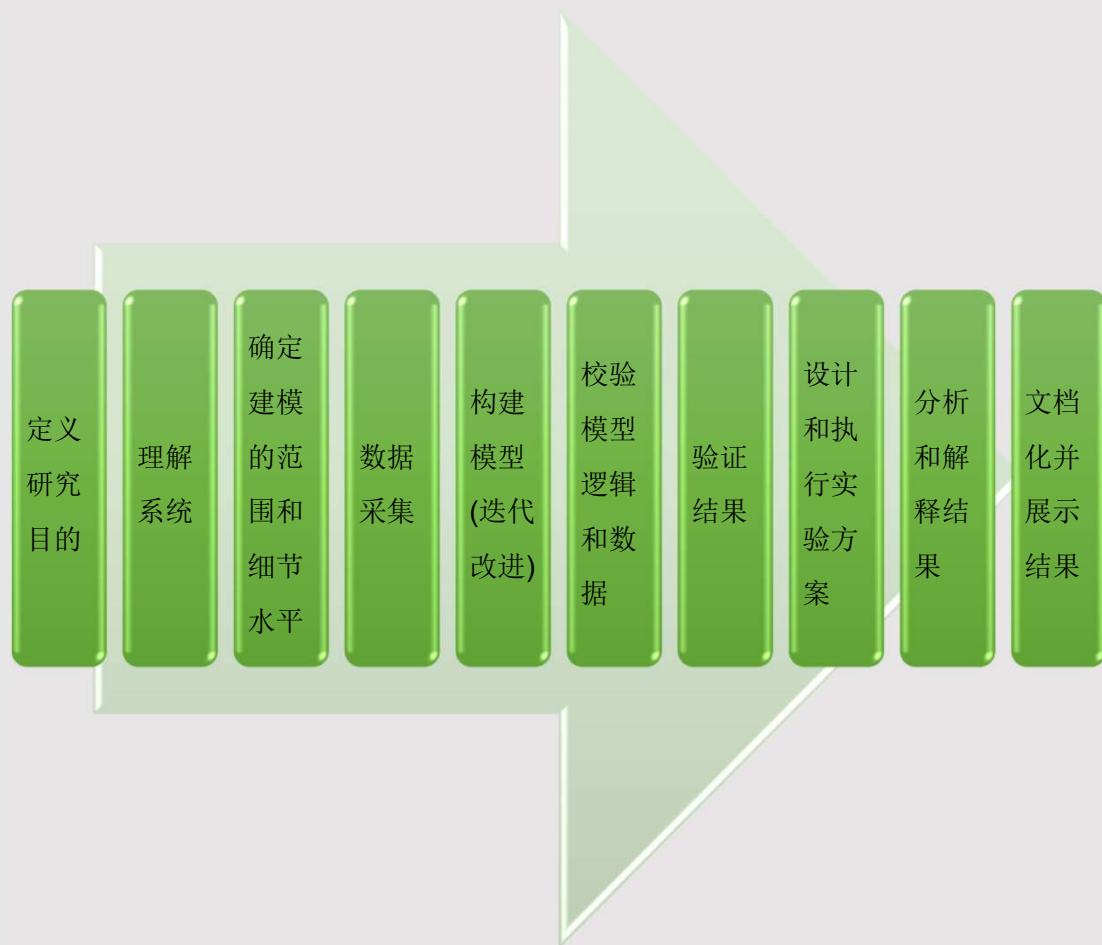
适用于绝大多数模型:

根据实战经验，仿真项目大约需要花费 **5-10%** 的项目总时间用于开发目标和功能规格书。

注意这并不是多余的时间。而是将我们选择的重要工作提前到项目开始阶段来进行而已，作为整个项目的计划和评估。

- 目标
- 系统描述和建模方法
- 所需输入数据
- 期望的实验方案
- 交付物

系统仿真项目步骤



仿真实施的小循环

- ▶ 仿真新手们总是从开始建模时，不断给模型增加内容直到模型完成，直到那个时候才运行仿真模型。

千万不要这样做！

- ▶ 使用循环迭代的模型构建方式
 - ▶ 添加功能/模型部件
 - ▶ 运行/测试模型
 - ▶ 重复小循环
- ▶ “尽早保存，经常保存！”

仿真的常见误区

- 从头就不理解仿真的目的
- 想的太大/太复杂
- 没有完全考虑系统的变化性
- 获得必要的数据
- 没有充分校验/验证模型
- 错误地解释随机结果
- 不及时的结果

Simio的实用价值

快速构建模型

灵活的建模方法

易于扩展

整合的三维动画

强大的输出分析工具

强大且易用的纠错工具

庞大活跃的用户社群

Simio软件版本

Simio 个人版

- 用于学习和建立小规模模型
- 全功能，但是模型的规模有限制
- 可以运行专业版和企业版建立的模型

Simio 设计版

- 使用过程开发定制化逻辑，完全的灵活性掌控
- 建立和重用定制的对象库

Simio 专业版

- 在计算机组内多个电脑上执行模型实验
- 定制化的图形面板和报告
- 可以发布在个人版中运行的模型

Simio 企业版

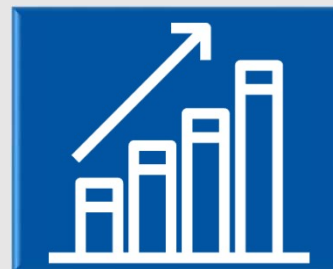
- 基于风险的高级计划和调度技术 (RPS)
- 高端的建模功能 (甘特图, 计划目标, ...)

Simio 门户版

- 将模型运行结果部署到任何设备，任何地方
- 使用云计算资源运行大量的并行仿真实验

第一章总结

- ▶ 仿真可以用于设计广泛的系统
- ▶ 精确地应对系统变化
- ▶ 提供目标衡量指标，帮助用户预测绩效，测试创意想法，降低风险，交付卓越的表现
- ▶ 吸引人的三维动画促进交流沟通



第二章



Simio介绍和基本动画



(0510) 82132584



无锡迅合信息科技有限公司©

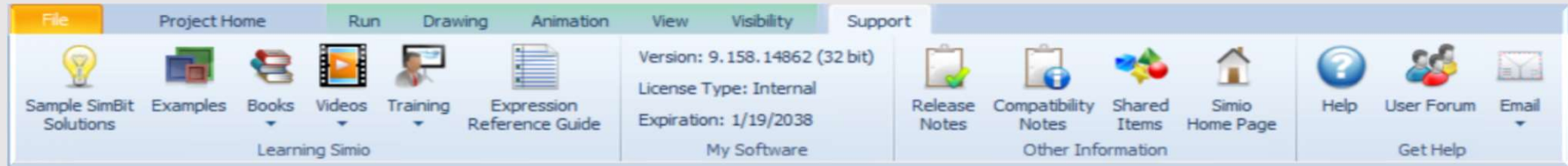


<http://www.simio-china.com/>

Simio介绍

- ▶ 学习资源
- ▶ 对象概念的概览
- ▶ 用户界面和标准库概览
- ▶ 表达式和属性
- ▶ 仿真实验基础- 运行样本
- ▶ 响应变量和响应结果
- ▶ 引用属性
- ▶ 小练习- 优化缓冲区分配
- ▶ 基本动画

► Support页面



► 多个视频集，好几百个YouTube在线视频

► 出版物（英文）

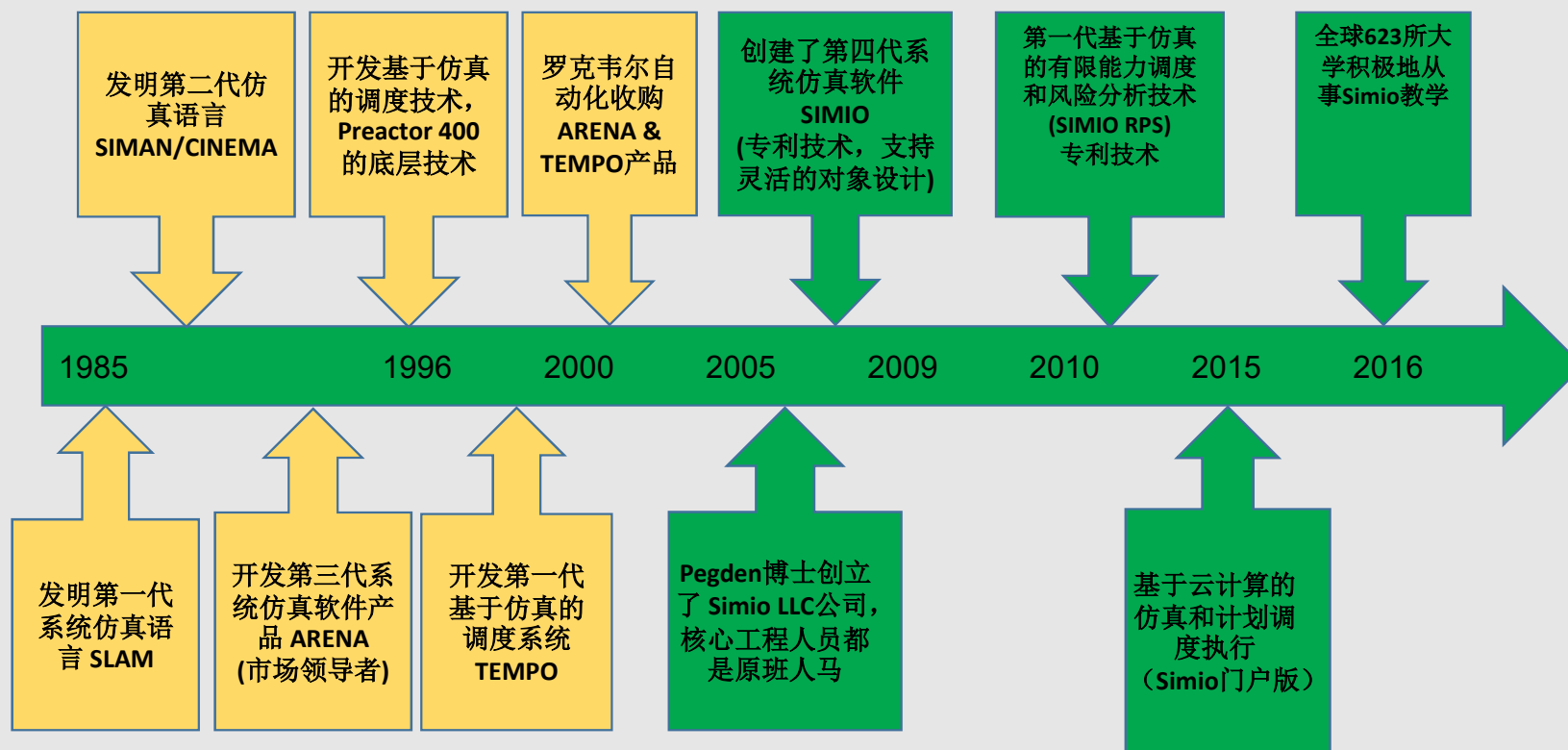
- 《使用Simio仿真建模：工作手册》，作者**Joines**和**Roberts**
- 《**Simio**和仿真：建模，分析，应用》，作者**Kelton, Smith, Sturrock**
- 《快速建模解决方案：系统仿真和Simio介绍》
- 《流程改进原则》作者：**Pegden**博士
- 《交付你的承诺》作者：**Pegden**博士

► 内部教材：

- 《互动学习仿真建模：Simio操作手册》中文版
- 《系统仿真-Simio手册》中文版



Simio产品历史背景信息



C. Dennis Pegden, 博士
创始人和首席执行官



David Sturrock先生
运营副总裁



Richard Ritchie先生
市场营销副总裁

对象概念的概览

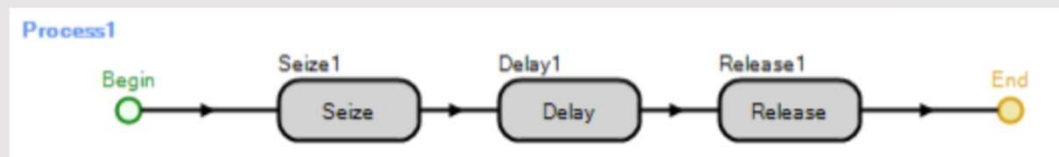
对象 (Object)	定义数据，逻辑，行为，视图，事件，和其他对象进行交互。
模型 (Model)	可执行的一个对象。
项目 (Project)	模型/对象的集合。 项目可以被重新加载，作为库模板被使用。
属性 (Property)	对象的静态输入值。
状态值 (States)	和对象关联的可变数值。
事件 (Event)	支持对象之间的沟通交流，当关键情况发生时被触发。
资源 (Resource)	系统的约束。一个对象（甚至一个实体）可能是资源。
实体 (Entity)	在系统内移动的动态对象。
运输器 (Transporter)	可以“加工”或“搬运”其他实体的特殊实体，车辆对象(Vehicle)和工人(Worker)对象都是实体（和资源）。

Simio中的过程

- ▶ 过程是一系列的活动，随着时间推进而可以改变系统的状态。
- ▶ 所有的对象都是通过流程和其他对象的组合、组建而成。
- ▶ 附加过程（**Add-on Processes**）提供了强大的机制能够增加模型灵活性而不需要编写代码。

- ▶ 过程步骤执行操作，例如：

- ▶ 延迟一段指定的时间
- ▶ 抓取或释放一个对象
- ▶ 触发一个事件，或等待事件的发生
- ▶ 根据概率和条件进行决策
- ▶ 将实体转移到站点内部
- ▶ 寻找某个实体或对象...



Simio用户界面

The screenshot shows the Simio software interface with several components highlighted by red arrows and Chinese labels:

- File菜单**: Points to the File menu in the top-left corner.
- 功能按钮**: Points to the Run, Step, Fast-Forward, and Reset buttons in the top toolbar.
- 项目标签页**: Points to the 'Definitions' tab in the top navigation bar.
- 库 (模版)**: Points to the 'Standard Library' on the left side of the interface.
- 项目**: Points to the 'MySimioProject' folder in the 'Browse' panel on the right.
- 当前模型**: Points to the 'Model' object selected in the 'Browse' panel.
- 属性窗口**: Points to the 'Properties: Model (Fixed Model)' window on the right side.
- 建模视图 (Facility)**: Points to the main workspace area where the model is built.

Simio标准库

发生器 (**Source**) : 产生特定类型和到达模式的实体。

销毁器 (**Sink**) : 销毁 (回收) 实体, 使其离开系统边界。

服务器 (**Server**) : 具有产能的过程, 例如一个机器设备。

工作站 (**Workstation**) : 具有产能的过程, 包括了准备, 加工, 拆卸等更复杂的工作。

合并器 (**Combiner**) : 将实体和“父实体”合并在一起 (例如: 托盘为一个父实体)。

分离器 (**Separator**) : 拆分已合并的批次, 或者复制实体。

资源 (**Resource**) : 被对象所获取和释放。

车辆 (**Vehicle**) : 按固定线路, 或按需进行装载/卸载的资源。

工人 (**Worker**) : 可移动资源, 执行静态/或非静态任务。

基本节点 (**BasicNode**) : 简单的网络交点, 固定对象的输入口。

转移节点 (**TransferNode**) : 改变实体的目的地/搭载, 固定对象的输出节点。

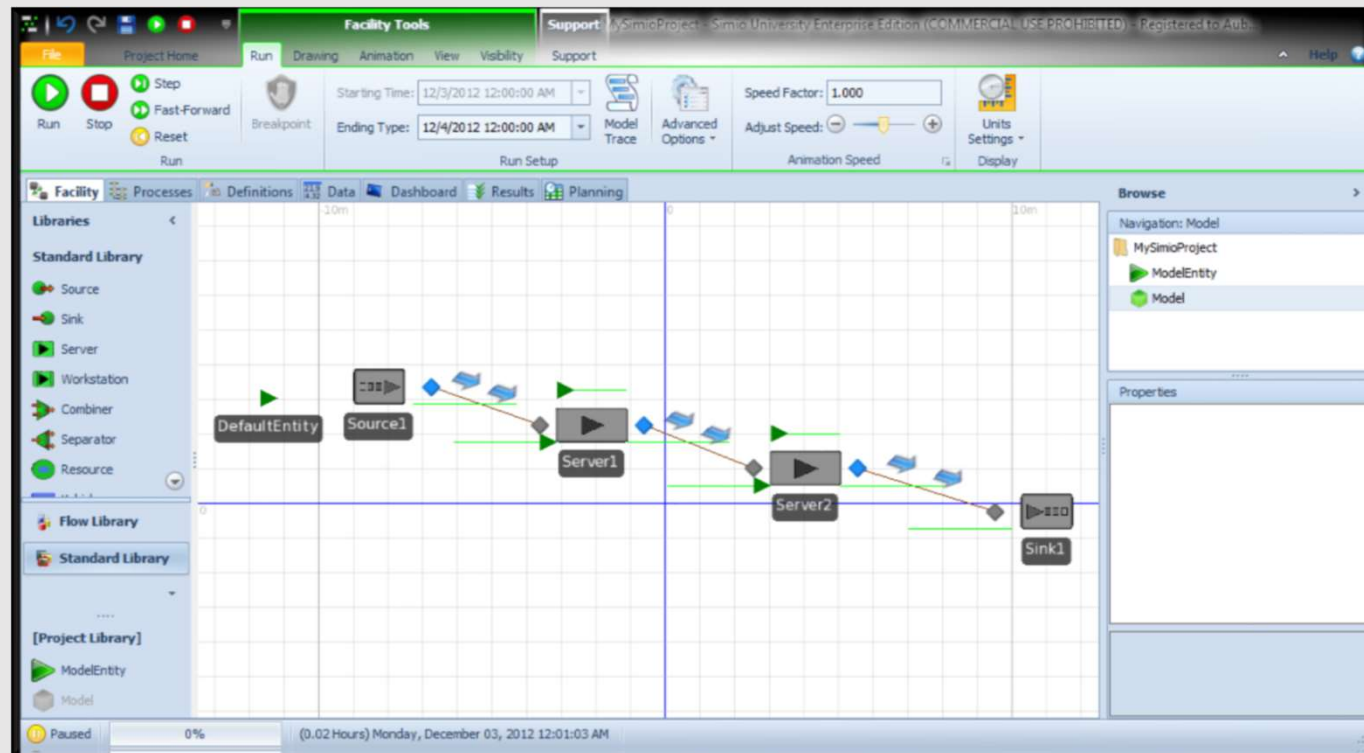
连接器 (**Connector**) : 零移动时间。

路径 (**Path**) : 实体在该路径上以指定的速度独立地移动。

时间路径 (**TimePath**) : 实体在该路径上以指定的总时间通过。

传送带 (**Conveyor**) : 累积式/非累积式的传送带设备。

Simio简单流水线

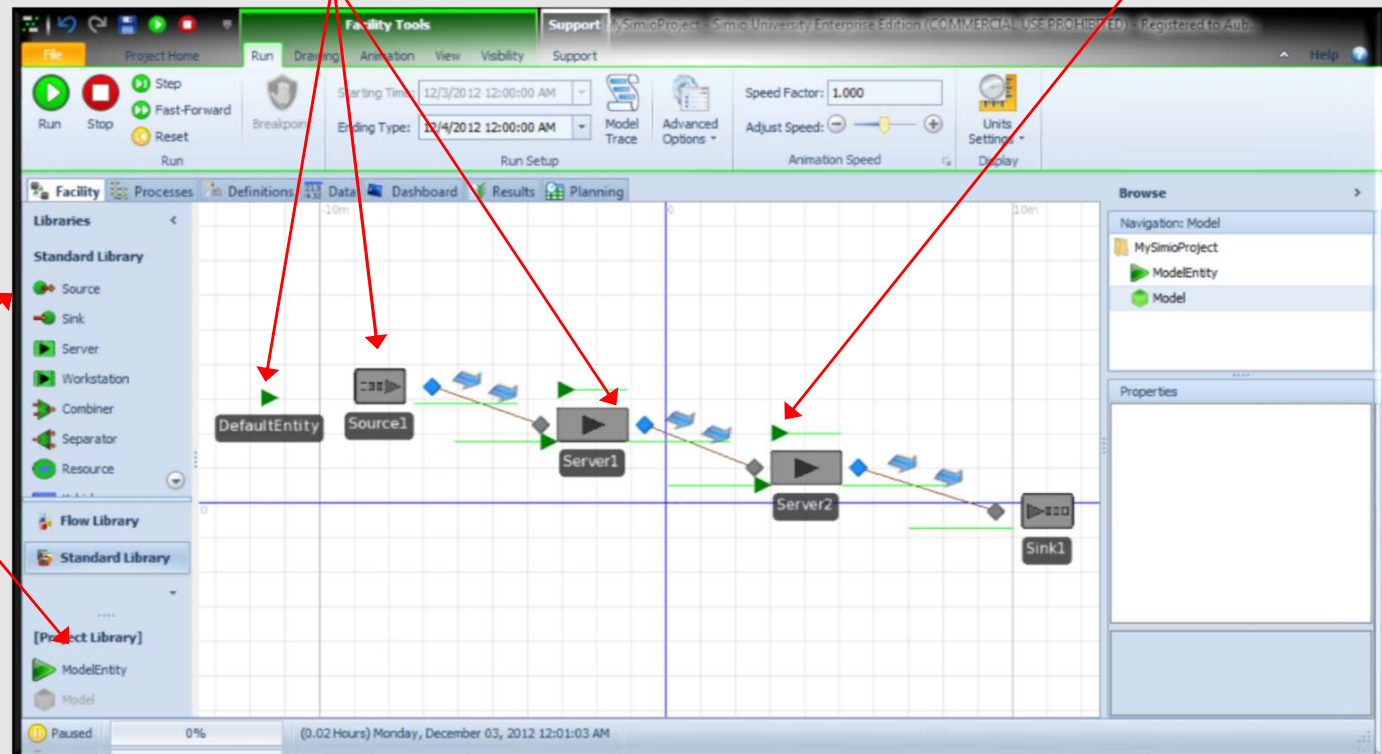


对象结构层次-“对象”家族

Object *Instance*
(对象实例)

Object *RunSpace*(*Realization*)
对象运行空间 (对象实现)

Object *Definition*
(对象定义)



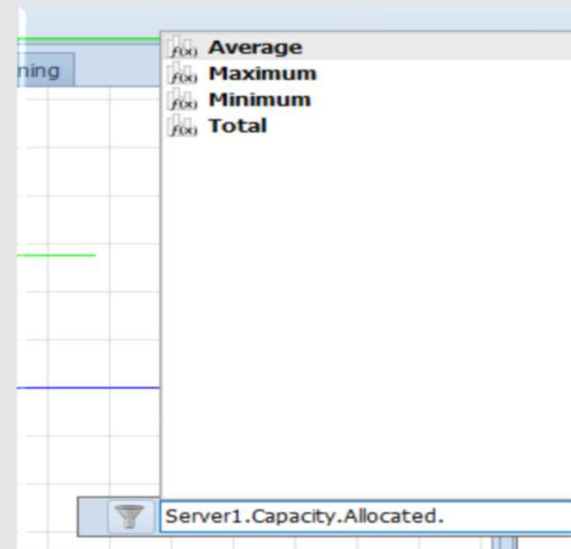
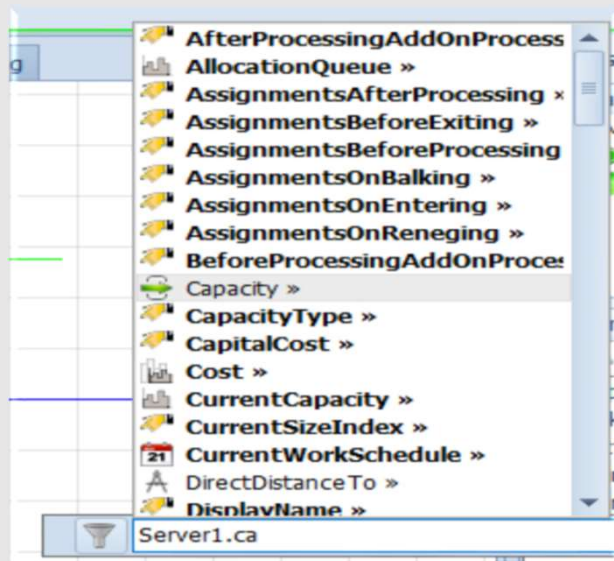
简单流水线-细节



串联的队列模型				
Arrival Rate (到达率)	25.00	实体/小时		
Mean IAT (平均到达间隔)	0.04	小时		
Mean IAT (平均到达间隔)	2.40	分钟		
		服务台1	服务台2	
Service Rate (服务率)	30.00	35.00	实体/小时	
Mean ST (平均服务时间)	0.03	0.03	小时	
Mean ST (平均服务时间)	2.00	1.71	分钟	
ρ (利用率)	0.83	0.71		
L (系统实体数量)	5.00	2.50	7.50	实体
W (系统停留时间)	0.20	0.10	0.30	小时

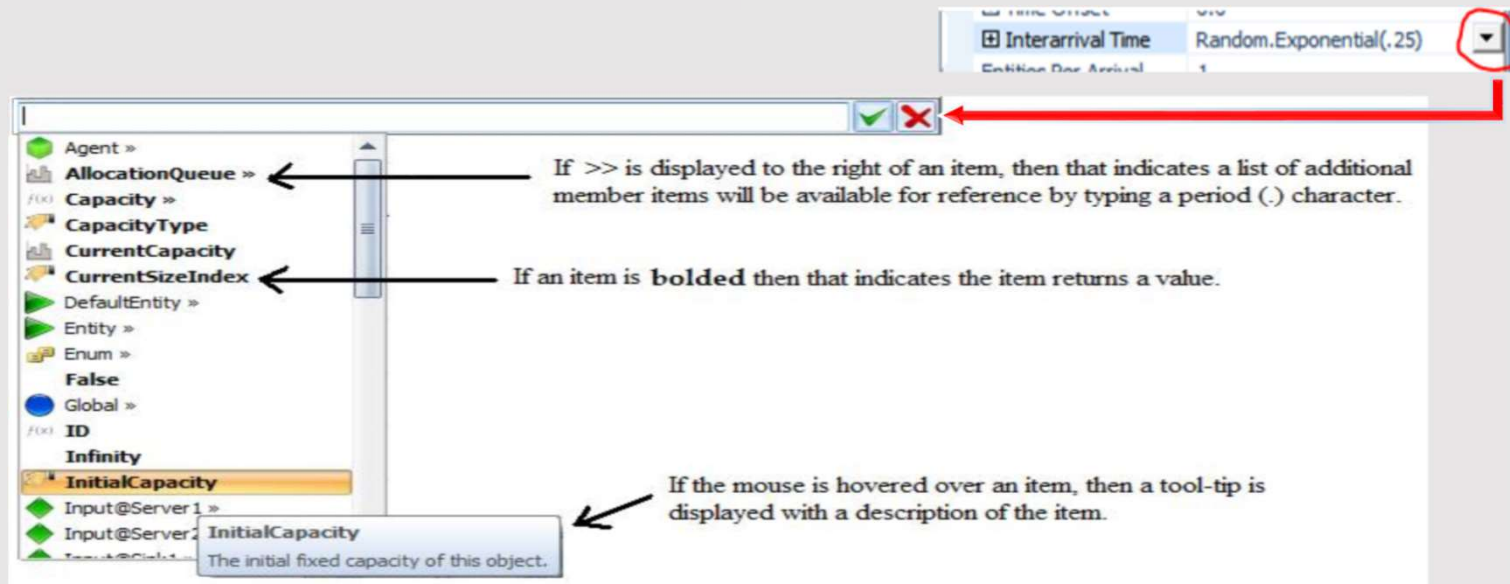
Simio的点标记法

- ▶ Simio使用“点”标记法访问对象的数据，例如属性和状态值。
- ▶ 一般表达形式为“xxx.yyy”，yyy是对象xxx的组成部分。
- ▶ **Server1.Capacity.Allocated.Average** 返回Server1对象的平均分配能力。



编辑对象的属性

- ▶ 在属性窗口显示
- ▶ 由对象的创建者定义
- ▶ 整型, 布尔型, 选择规则, 表达式, ...
- ▶ 表达式编辑器 (右边的下拉按钮)
- ▶ 默认的结果是经过过滤的



排名前十位的表达式

- ▶ **X** (数值 – 整数或实数)
- ▶ **RANDOM.Exponential(mean)** 指数分布
- ▶ **RANDOM.Triangular(min, mode, max)** 三角分布
- ▶ **RANDOM.Uniform(min, max)** 均匀分布
- ▶ **MATH.{Lots of Functions}** 数学公式 (有大量的选项)

- ▶ **ServerName.CAPACITY.ScheduledUtilization** 服务器的计划利用率
- ▶ **ServerName.InputBuffer.Contents** 服务器的输入缓冲队列的长度
- ▶ **DefaultEntity.POPULATION.TimeInSystem.Average** 实体在系统中平均停留时间
- ▶ **DefaultEntity.POPULATION.NumberInSystem.Average** 实体在系统中的平均数量
- ▶ **SinkName.InputBuffer.NumberEntered** 实体销毁的数量
- ▶ **SinkName.TimeInSystem.Average** 销毁器接受到的实体在系统中的平均停留时间

提示

KEYWORD

Object Name

Station Name

简单流水线-表达式

- ▶ Source1的到达间隔：
 - ▶ Interarrival Time: Random.Exponential(2.4)
- ▶ Server1加工时间：
 - ▶ Processing Time: Random.Exponential(2)
- ▶ Server2加工时间
 - ▶ Processing Time: Random.Exponential(1.7)

*单位是Minutes

数据透视表

The screenshot shows a software interface with a Pivot Grid. The interface includes a top menu bar with 'File', 'Project Home', 'Pivot Grid', and 'Support'. Below the menu bar are several toolbars with icons for 'Run', 'Stop', 'Step', 'Fast-Forward', 'Reset', 'Export Results', 'Change View', 'Add View', 'Manage Views', 'Units Settings', and 'Display'. The main area is divided into tabs: 'Facility', 'Processes', 'Definitions', 'Data', 'Results', and 'Planning'. The 'Results' tab is active, showing a Pivot Grid with a table of data.

The Pivot Grid table is structured as follows:

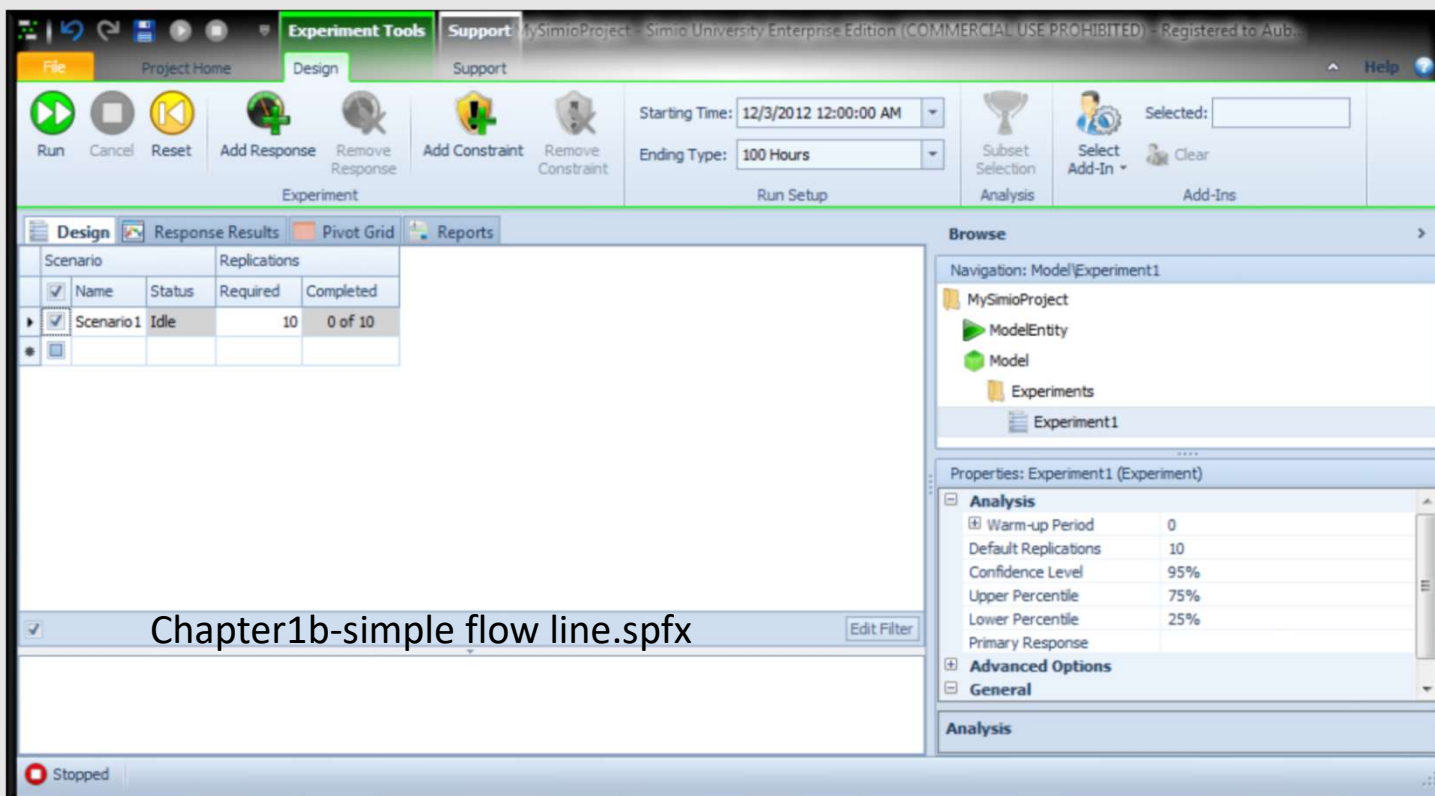
Drop Filter Fields Here						Drop Column Fields Here	
Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total	
ModelEntity	DefaultEntity	[Population]	Content	NumberInSystem	Average	8.2724	
					Maximum	20.0000	
			FlowTime	TimeInSystem	Average (Ho...	0.3210	
					Maximum (Ho...	0.7386	
					Minimum (Ho...	0.0060	
			Throughput	NumberCreated	Total	620.0000	
					NumberDestroyed	Total	618.0000
Path	Path1	[Travelers]	Content	NumberOnLink	Average	0.0115	
					Maximum	2.0000	
			FlowTime	TimeOnLink	Average (Ho...	0.0004	
					Maximum (Ho...	0.0004	
					Minimum (Ho...	0.0004	
			Throughput	NumberEntered	Total	620.0000	
				NumberExited	Total	620.0000	
Path2	Path2	[Travelers]	Content	NumberOnLink	Average	0.0051	

At the bottom of the interface, there is a status bar showing 'End of run', '100%', and '(24.00 Hours) Tuesday, August 29, 2017 12:00:00 AM'.

数据透视表-观察结果

- ▶ 操作，统计，显示，研究你的数据。
- ▶ 分组：拖动表格的列至不同的位置，使得数据可以不同地分组。
- ▶ 排序：点击列标题上的小三角使得用户可以按照列来进行升序和降序排列。
- ▶ 筛选：点击列标题上方的小漏斗图标，弹出下拉菜单，用户可以打开或关闭在那一列显示某个内容的开关。

简单流水线-仿真实验



简单流水线-响应变量



The screenshot displays the Simio software interface with the following components:

- Design Tab:** A table showing simulation results for 'Scenario1'.
- Response Results Table:**

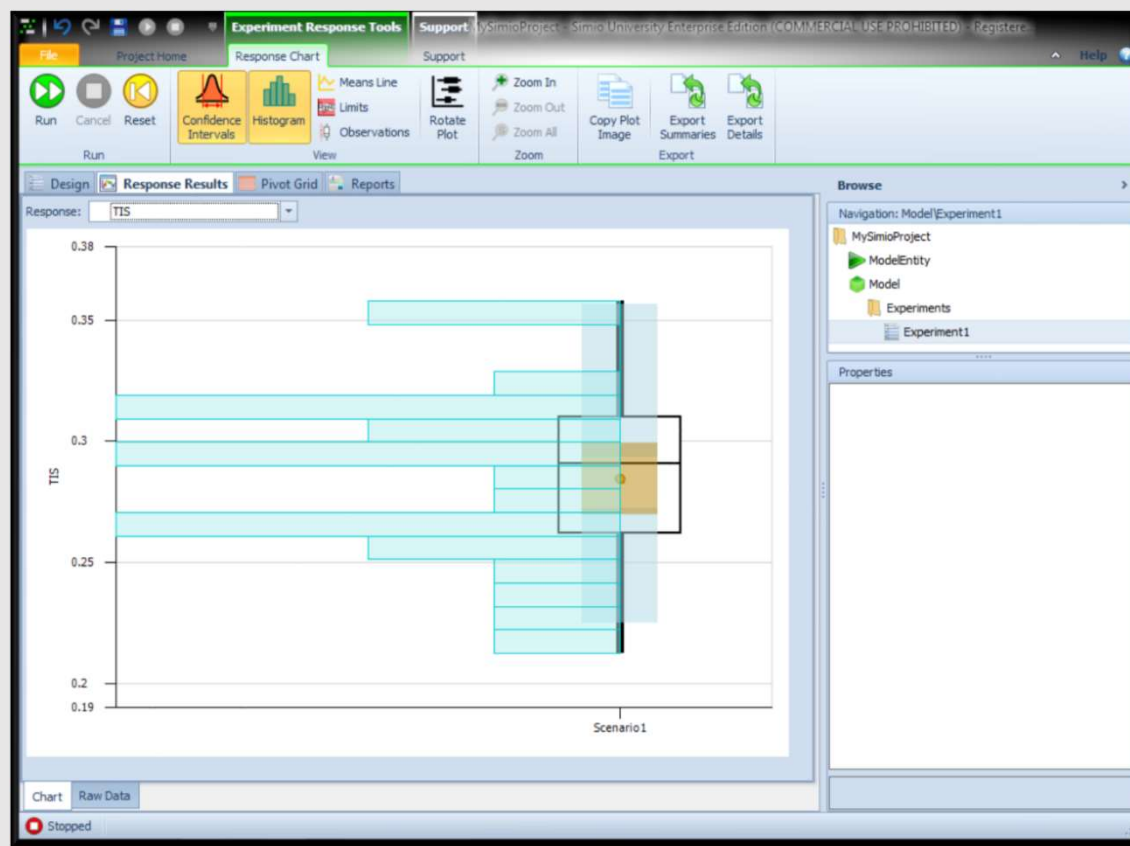
Scenario	Replications	Responses					
Name	Status	Required	Completed	UtilS1	UtilS2	NIS	TIS
Scenario1	Completed	10	10 of 10	83.6819	71.2385	7.38759	0.296855

- Properties Panel (Experiment1):**
 - Analysis: Warm-up Period: 0, Default Replications: 10, Confidence Level: 95%, Upper Percentile: 75%, Lower Percentile: 25%, Primary Response: UtilS1.
 - Advanced Options: (Expanded)
 - General: (Expanded)
- Log:** A list of messages indicating that 10 replications of Scenario1 are completed at simulation time 100.
- Status:** Stopped

名称和显示名称

- ▶ **Name**（名称）是唯一能够表示一个对象或组件的
 - ▶ 在对象里必须是唯一的，不能重复
 - ▶ 可以直接在表达式中使用
 - ▶ 不能和数字和表达式混淆
 - ▶ 没有以下这些操作符“.”，“-”，“+”，“/”，“*”
 - ▶ 不能用纯粹的数字，例如“123”或者“1e5”，但是可以字母包含数字，例如“10mmWidget”或“Whatzit1025”都是合法的名称
 - ▶ 内部不能有空格
 - ▶ 支持标准字符集或者扩展字符集包括大写和小写的Unicode统一码
 - ▶ 大小写不敏感
- ▶ **Display Name**（显示名称）
 - ▶ 默认和名称一样 (先设置Name)
 - ▶ 显示上更加友好 (比如e.g.列的标题)
 - ▶ 可以包括空格和特殊符号

简单流水线-SMORE图例（风险/误差指标）



简单流水线-通过率响应变量



Design | Response Results | Pivot Grid | Reports

Scenario		Replications		Responses				
Name	Status	Required	Completed	UtilS1	UtilS2	NIS	TIS	Thput
Scenario1	Completed	100	100 of 100	83.1296	70.8338	7.38734	0.295401	24.9042

Browse

Navigation: Model\Experiment1

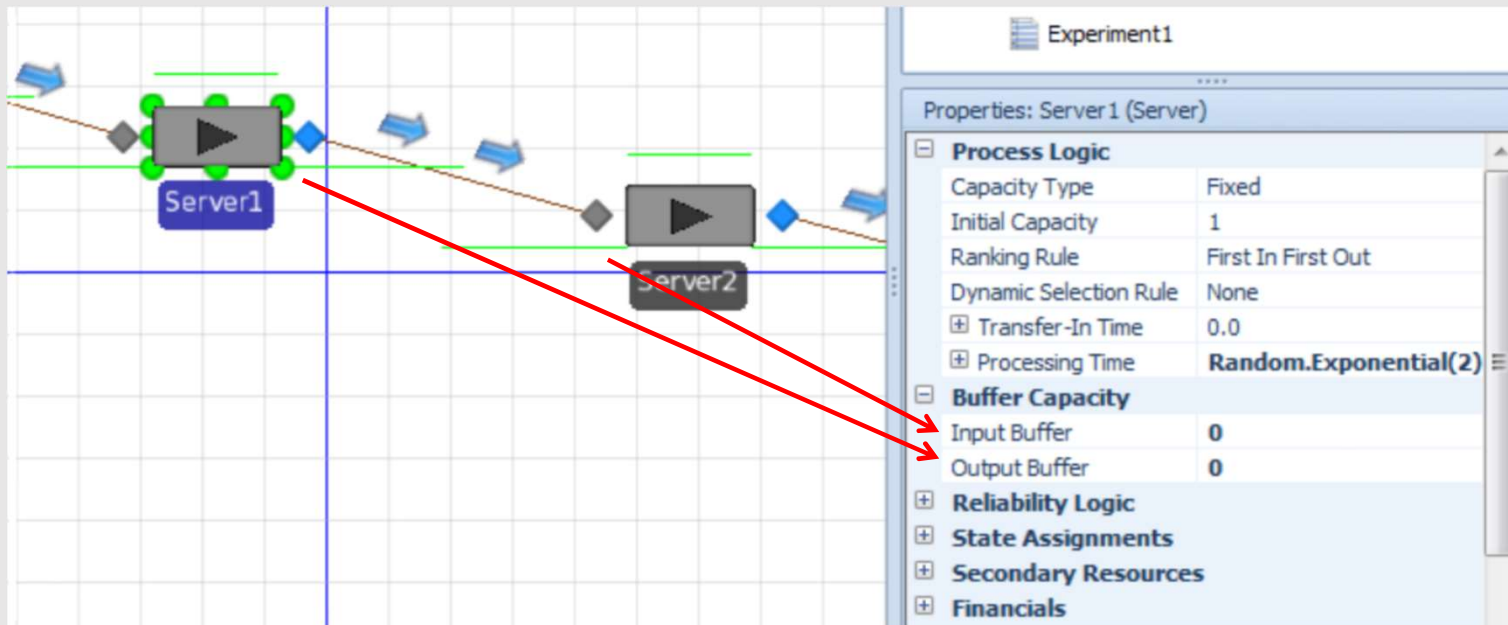
- Model-02
 - ModelEntity
 - Model
 - Experiments
 - Experiment1

Properties: Thput (Response)

General

Name	Thput
Display Name	Thput
Expression	Sink1.InputBuffer.NumberEntered / TimeNow
Unit Type	Unspecified
Objective	None
Lower Bound	
Upper Bound	

缓冲区和引用属性



缓冲区和引用属性



The screenshot displays a simulation environment with a process flow diagram on the left and several property windows on the right. The process flow diagram shows two servers, Server1 and Server2, connected by a path. Red arrows point from the servers to their respective property windows.

Properties: Server 1 (Server)

- Process Logic
 - Capacity Type: Fixed
 - Initial Capacity: 1
 - Ranking Rule: First In First Out
 - Dynamic Selection Rule: None
 - Transfer-In Time: 0.0
 - Processing Time: **Random.Exponential(2)**
- Buffer Capacity
 - Input Buffer
 - Output Buffer
- Reliability Log
- State Assignm
- Secondary Re
- Financials
- Add-On Proce

Views

Name	Object Type	Display Name
> Properties (Inherited)		
> WorkDayExceptions.Properties (Inherited)		
> WorkPeriodExceptions.Properties (Inherited)		
> Properties		
Buff1	Numeric Property	Buff1
Buff2	Numeric Property	Buff2

Properties: Buff1 (Numeric Property)

- Logic
 - Default Value: 0
 - Switch Property Name
 - Switch Condition: Equal
 - Switch Value

缓冲区和引用属性



Experiment Tools | Support | Model-02 - Simio University Enterprise Edition (COMMERCIAL USE PROHIBITED) - Register...

File | Project Home | Design | Support

Run | Cancel | Reset | Add Response | Remove Response | Add Constraint | Remove Constraint

Starting Time: 12/3/2012 12:00:00 AM | Ending Type: 100 Hours | Subset Selection | Add-Ins

Scenario	Name	Status	Replications		Controls		Responses				
			Required	Completed	Buff1	Buff2	UtilS1	UtilS2	NIS	TIS	Thput
Scenario1	Completed	Completed	10	10 of 10	0	0	71.0587	61.6989	166.889	6.65862	21.344
1x1	Completed	Completed	10	10 of 10	1	1	79.779	68.997	67.1015	2.69668	23.819
2x2	Completed	Completed	10	10 of 10	2	2	83.0473	70.3753	27.0905	1.08529	24.574
3x3	Completed	Completed	10	10 of 10	3	3	83.938	70.7633	11.4646	0.460191	24.799
3x2	Completed	Completed	10	10 of 10	3	2	83.0473	70.3753	27.0905	1.08529	24.574
3x1	Completed	Completed	10	10 of 10	3	1	79.779	68.997	67.1015	2.69668	23.819

Design | Response Results | Pivot Grid | Reports

Browse: Model-02 | ModelEntity | Model | Experiments | Experiment1

Properties: Experiment1 (Experiment)

Analysis: Warm-up Period: 0, Default Replications: 10, Confidence Level: 95%, Upper Percentile: 75%

Experiment Experiment1, Scenario 3x2, replication 9 completed at simulation time 100. Actual run time: 1.5110865 seconds.
Experiment Experiment1, Scenario 3x1, replication 9 completed at simulation time 100. Actual run time: 1.5350878 seconds.
Experiment Experiment1, Scenario 3x2, replication 10 completed at simulation time 100. Actual run time: 1.521087 seconds.
Experiment Experiment1, Scenario 3x1, replication 10 completed at simulation time 100. Actual run time: 1.5110864 seconds.

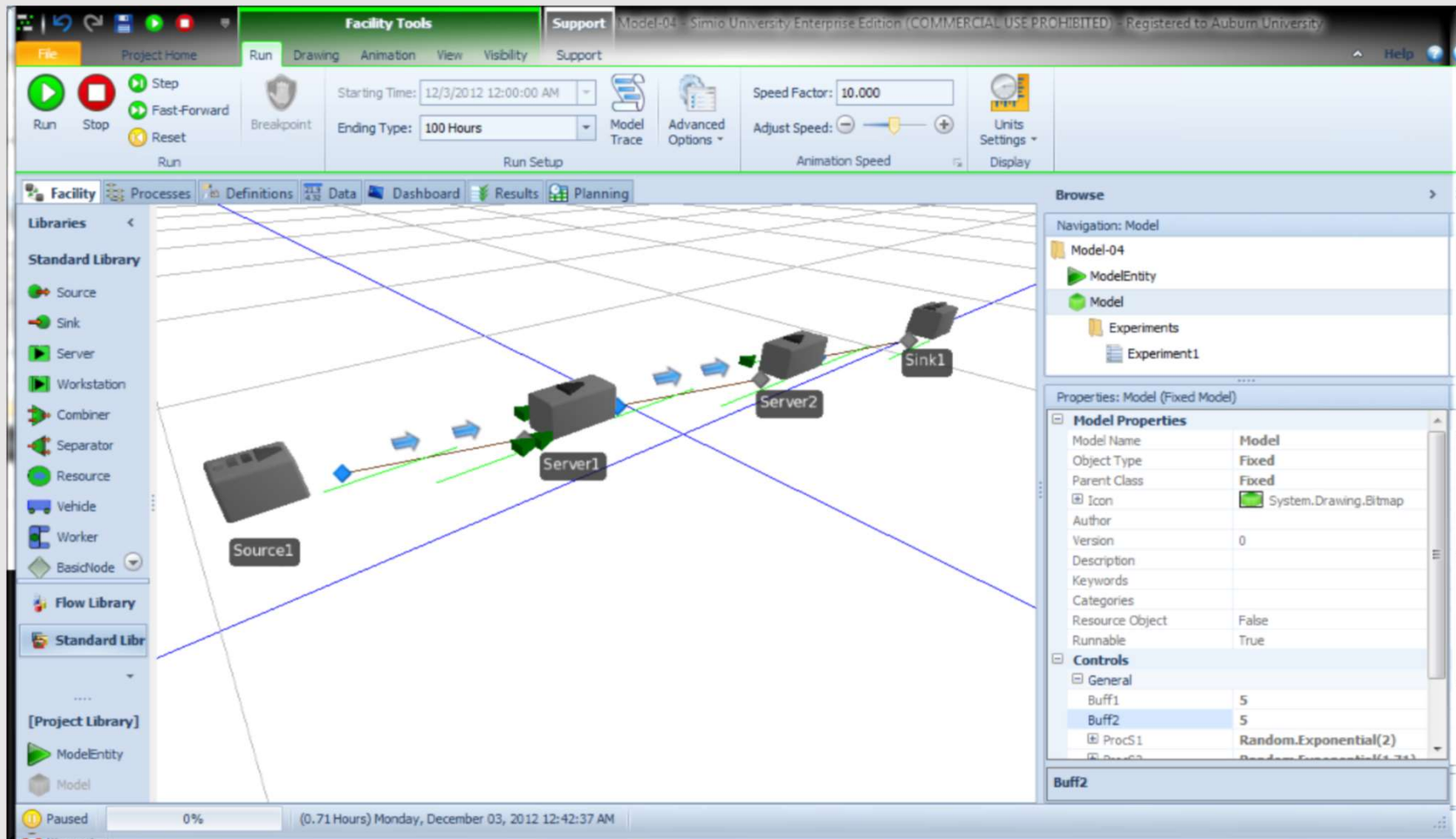
Stopped

实验练习

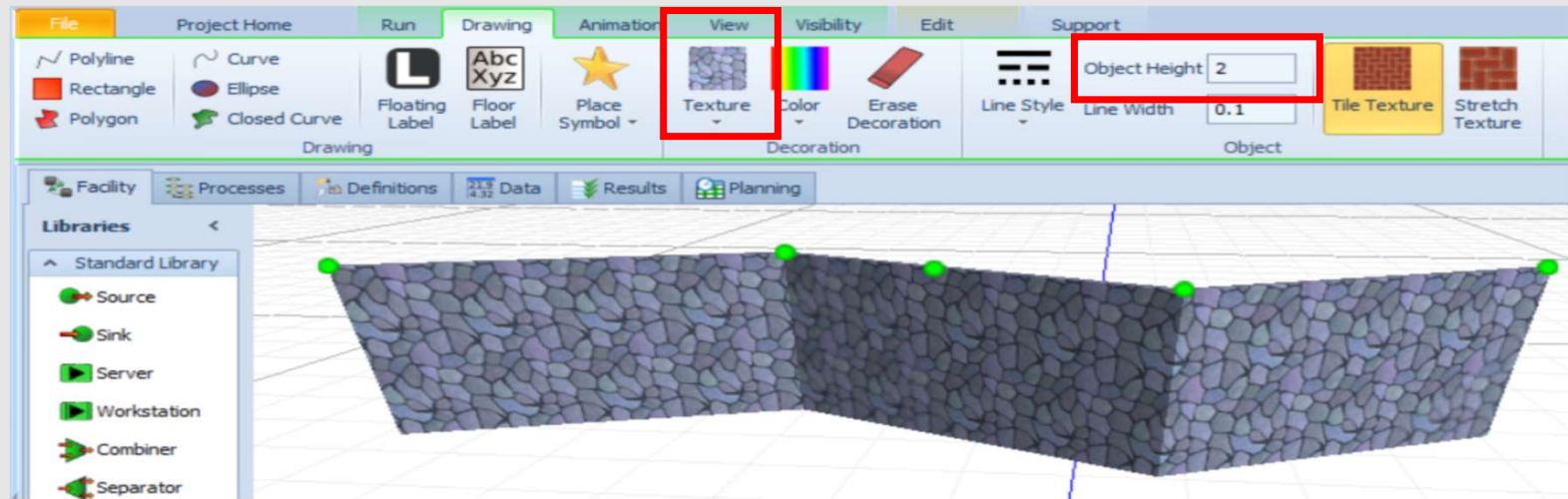
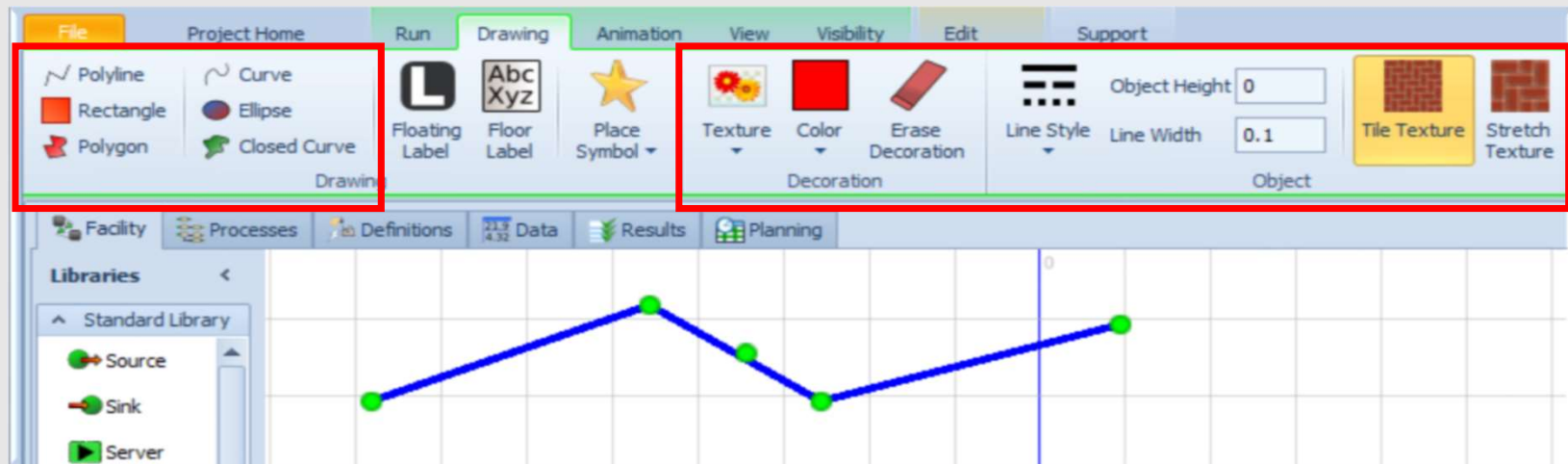


- ▶ 建立模型。
- ▶ 找到使产出率最大化的缓冲区配置，你只有7个单位的缓冲区容量（总量），包括Source的输出缓冲。
- ▶ Source1的到达间隔： **Interarrival Time: *Random.Exponential(2.4)***
- ▶ 更新两个工作站的服务时间如下：
 - ▶ Server1: ***Random.Triangular(0.5, 2, 3.5)***
 - ▶ Server2: ***Random.Uniform(1, 2.42)***
- ▶ 增加一个响应：
 - ▶ 止步率: ***Source1.OutputBuffer.NumberBalked***
- ▶ 重新运行仿真实验，注意和我们使用指数分布时候的产出率和缓冲区的差异。
- ▶ **挑战**: 显示一个仿真实验，使用原先的服务时间和修改后的服务时间。

Simio 三维模式



基本的绘图工具



状态标签（绑定）

The screenshot displays a simulation software interface. At the top, a menu bar includes 'File', 'Project Home' (highlighted with a red box), 'Run', 'Drawing', 'Animation', 'View', 'Visibility', and 'Symbols'. Below the menu bar is a toolbar with various icons for drawing and animation. The main workspace shows a simulation diagram with a green 'Server1' block and a grey status label '2'. A red arrow points from the 'Expression' field in the properties panel to the status label. The properties panel is titled 'Properties: InputBuffer.Contents (Status Label)' and contains the following information:

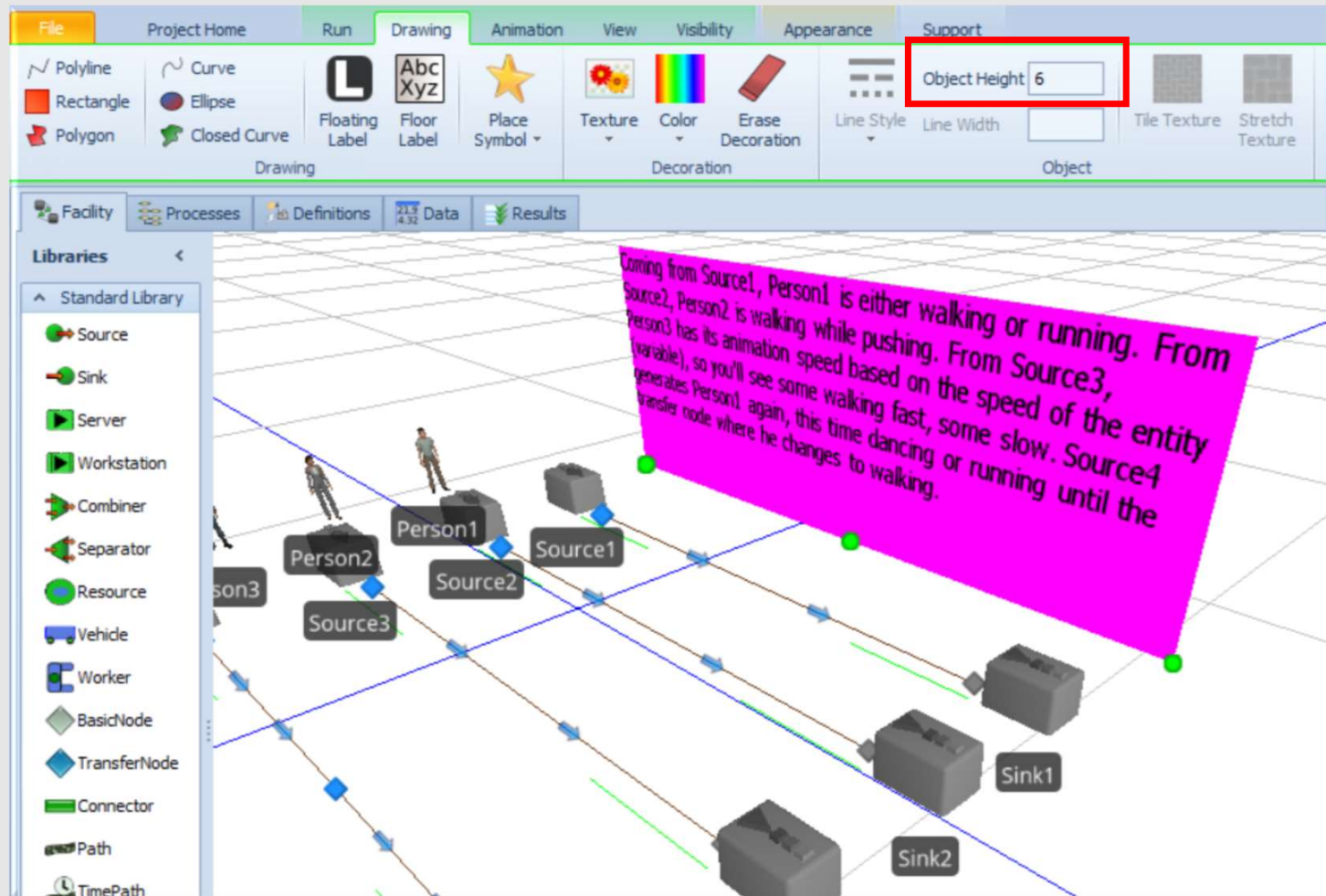
Properties: InputBuffer.Contents (Status Label)	
<input type="checkbox"/> Show Commonly Used Properties Only	
Appearance	
Back Color	170, 170, 170
Text Color	0, 0, 0
Animation	
Attached To	Server1
Expression	InputBuffer.Contents

地面标签（和实体绑定）

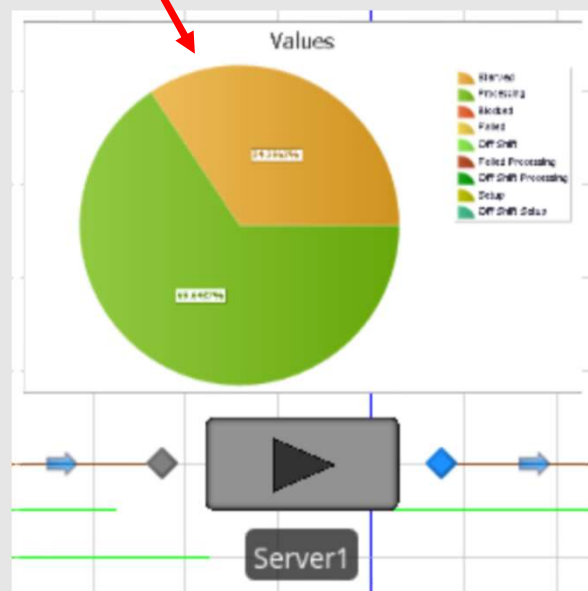
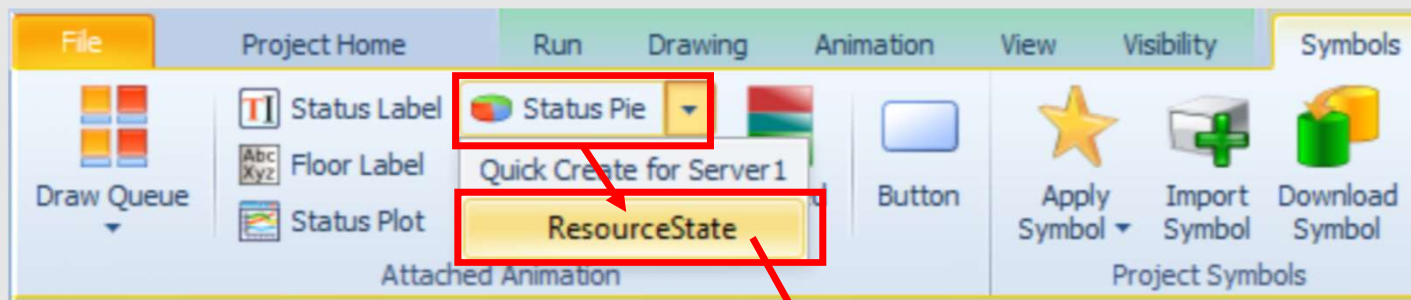
The image displays a simulation software interface with several key components:

- Top Panel:** Contains tabs for File, Project Home, Run, Drawing, Animation, View, Visibility, and Symbols. The 'Symbols' tab is active, showing various tool icons. The 'Floor Label' icon is highlighted with a red box.
- Libraries Panel:** Located on the left, it shows a 'Standard Library' with items like Source, Sink, Server, and Workstation. A 'PartA' entity is visible in the main workspace.
- Main Workspace:** Shows a grid with a 'PartA' entity (a blue rectangle) and several green circular entities. A label 'PartA' is placed on the grid.
- Configuration Panel:** A detailed view of the 'Floor Label' tool settings. It includes:
 - Text:** A text input field containing 'Abc Xyz'.
 - Size:** A 'Characters Per Line' dropdown set to 12.
 - Color:** A 'Background Color' selection box.
 - Simulation:** Radio buttons for 'Proportional' and 'Fixed'.
 - Label Text:** A text area containing instructions on using tags for formatting and expressions. It includes the example text: `Entity: {Name}` and `Created: {Math.Round(TimeCreated*60,2)}`.
- Inset Diagram:** A small diagram on the right shows a flow with a blue arrow, a green triangle, and a grey diamond, with labels 'Entity: PartA.26 Created: 4.2' and 'Entity: PartA.28 Created: 4.31'.

地面标签（作为广告牌/电视墙）



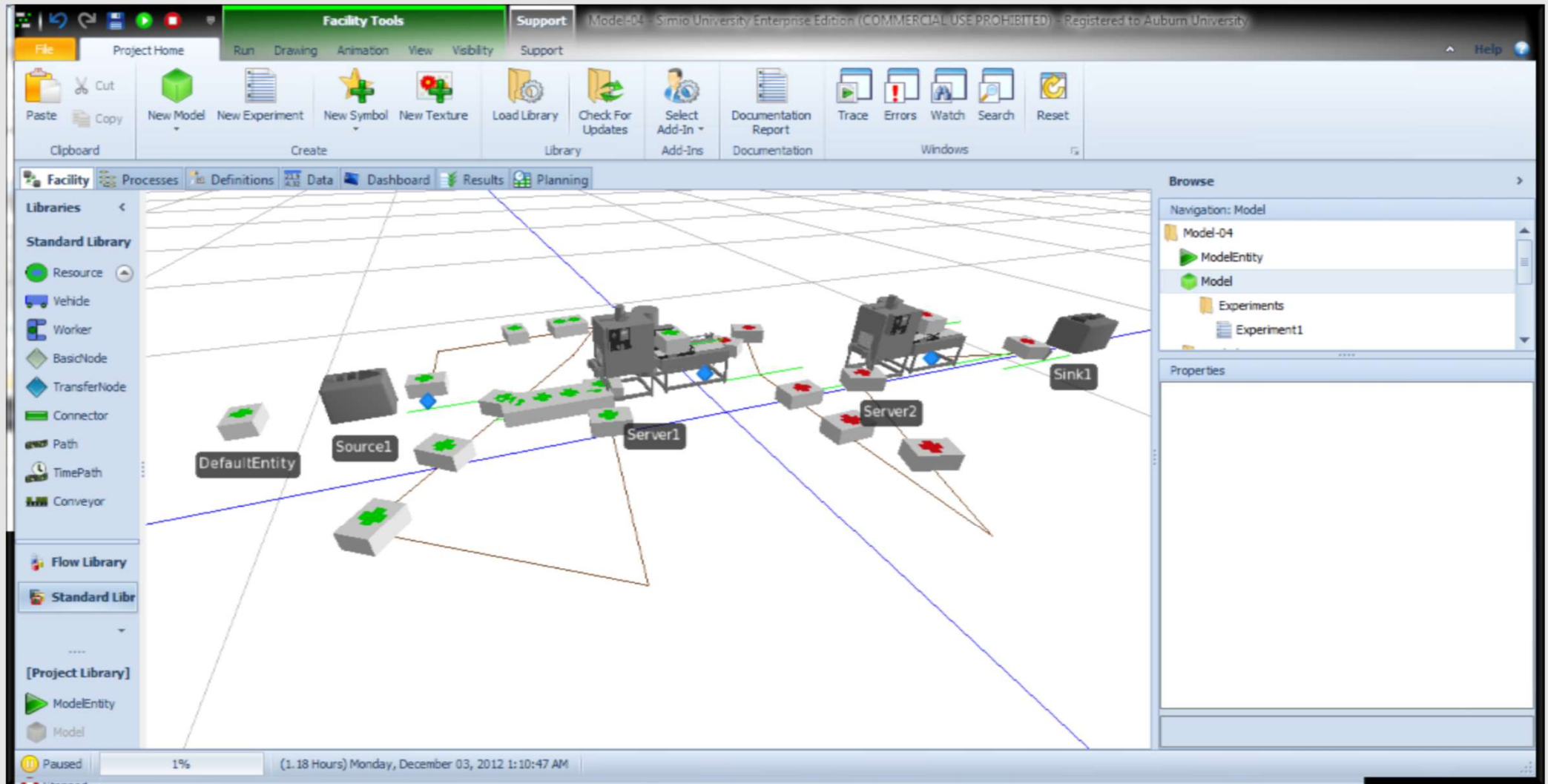
饼图（绑定服务器）



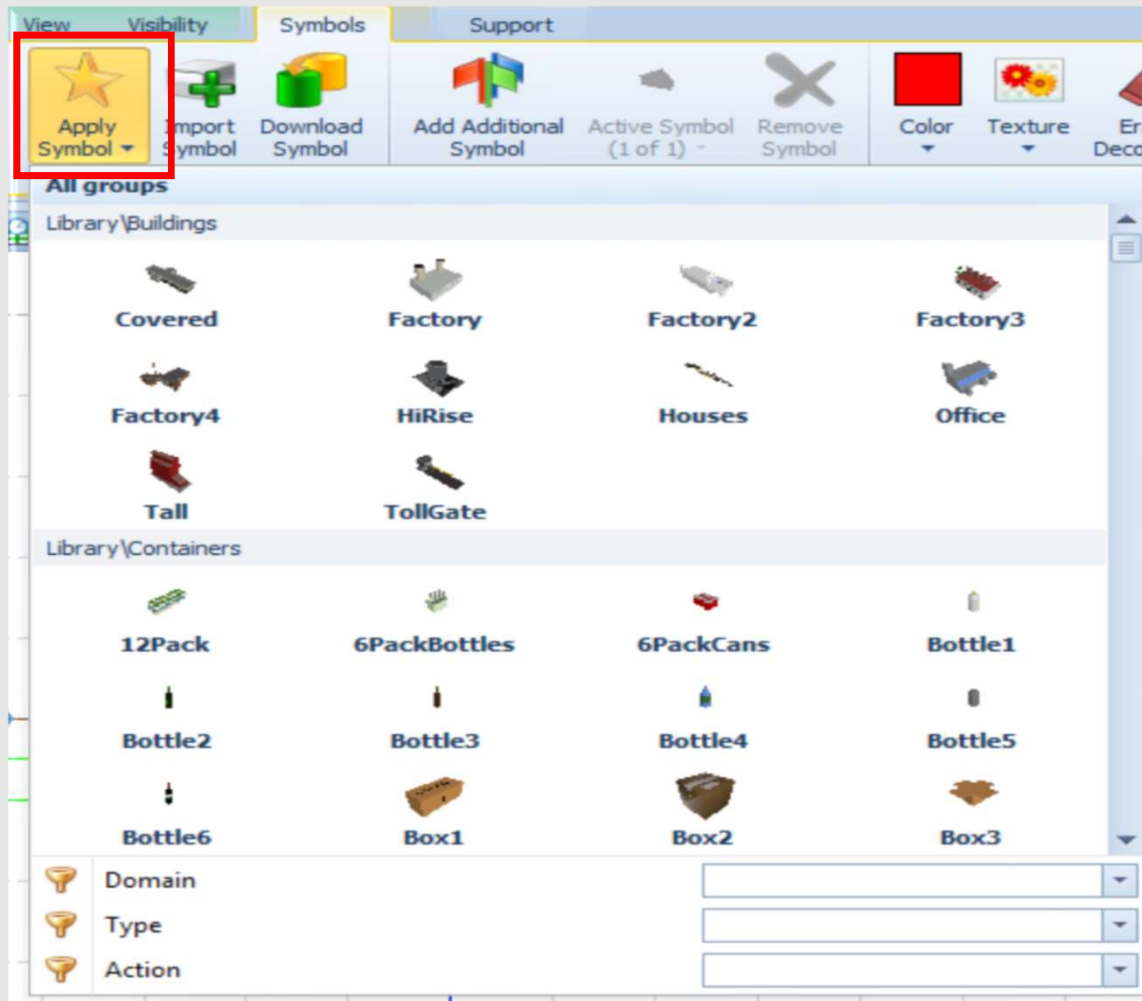
状态图(折线图)



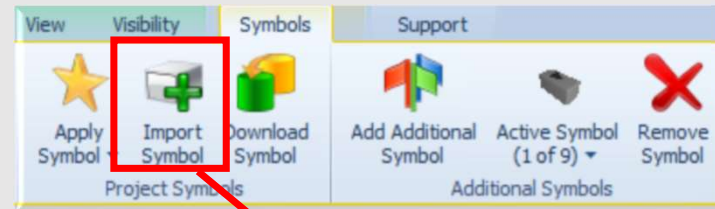
图形，状态赋值，3D仓库



图形选择-Simio图库



图形格式的导入



SketchUp Files (*.skp)
Autocad Drawing Exchange Format Files (*.dxf)
Autocad DWG Files (*.dwg)
Image File (*.jpg, *.png, *.bmp, *.gif)
Ogre Mesh File (*.mesh)
3DS Max Files (*.3ds)
AC3D Files (*.ac)
Collada Files (*.dae)
LightWave Files (*.lwo)
Milkshape 3D Files (*.ms3d)
Wavefront Files (*.obj)
Stanford Polygon Files (*.ply)
Stereolithography Files (*.stl)
FBX Files (*.fbx)
Blender Files (*.blend)

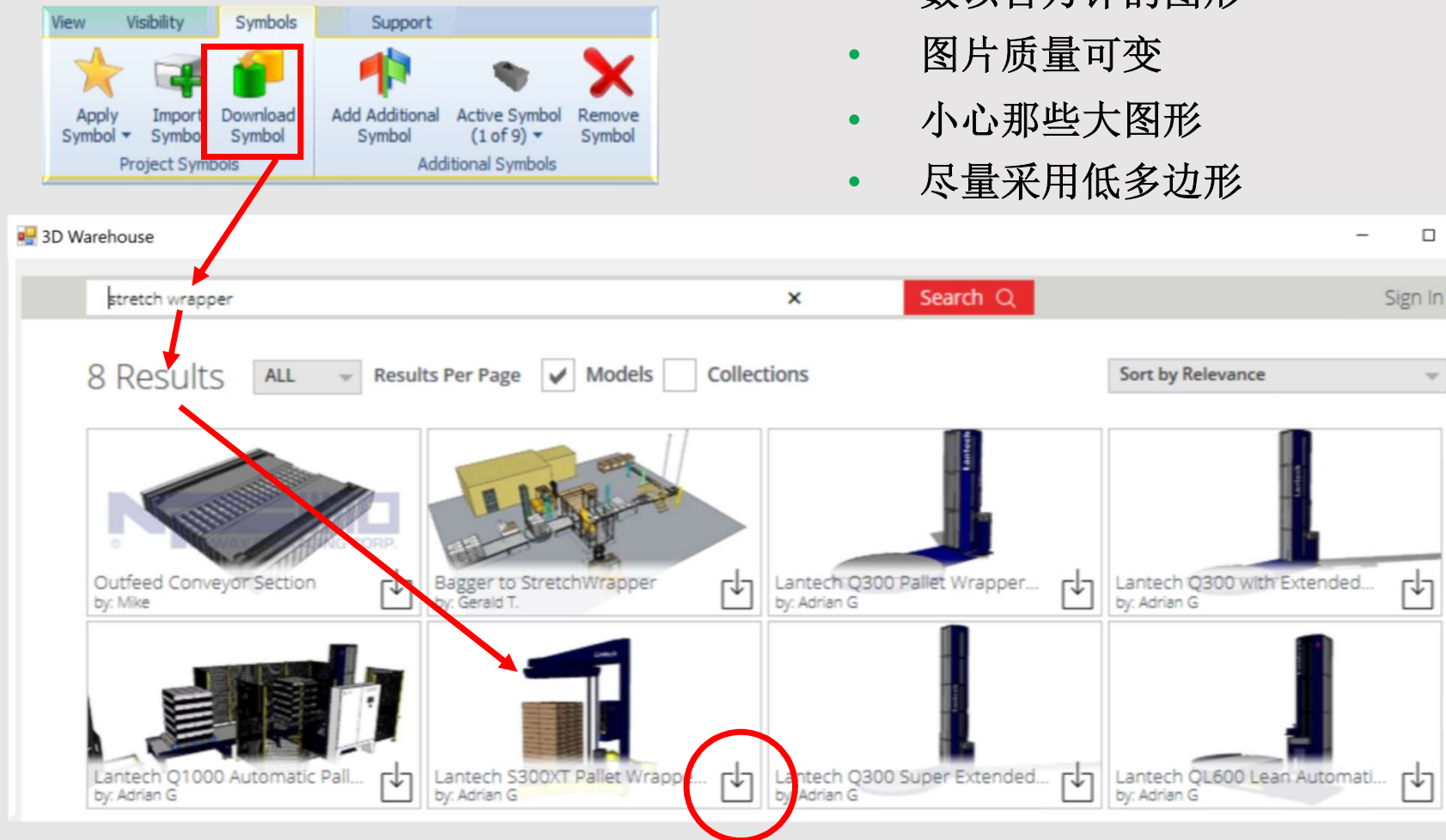
缩小文件的大小再保存

二维图可以做比较好的背景图

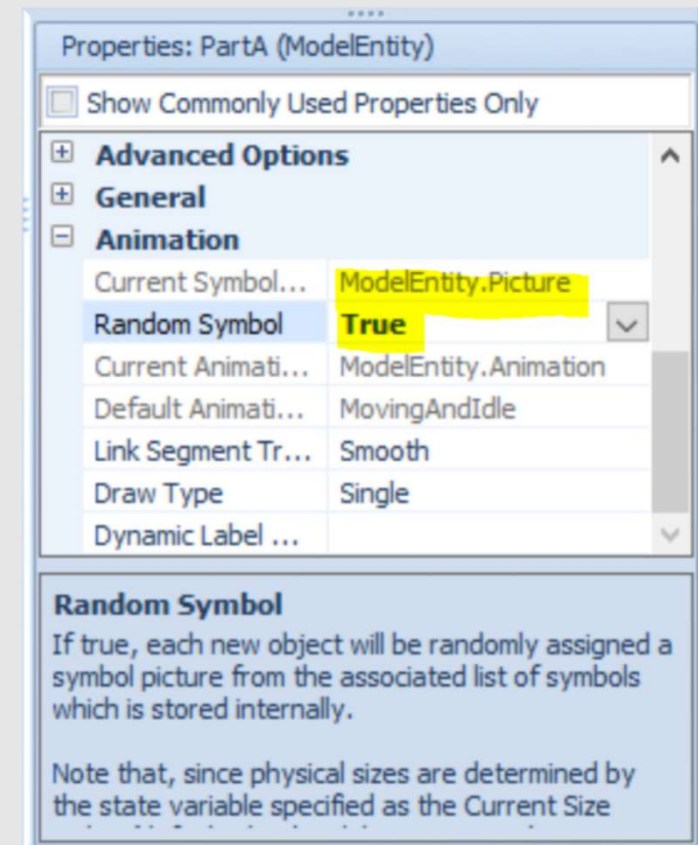
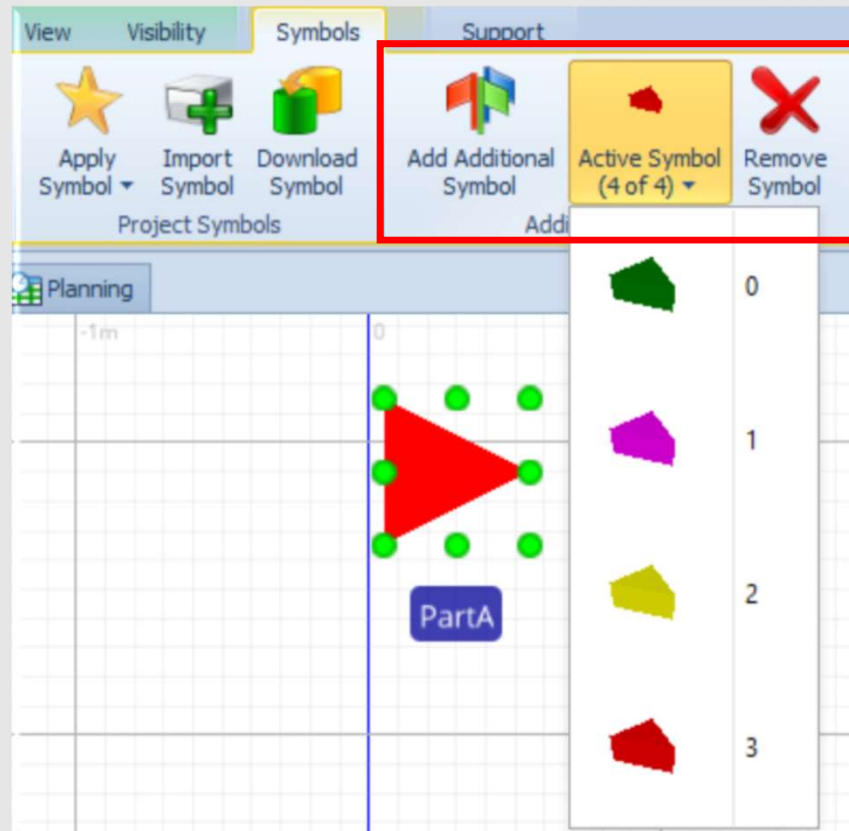
可以将2维用于形状

在线下载三维图形

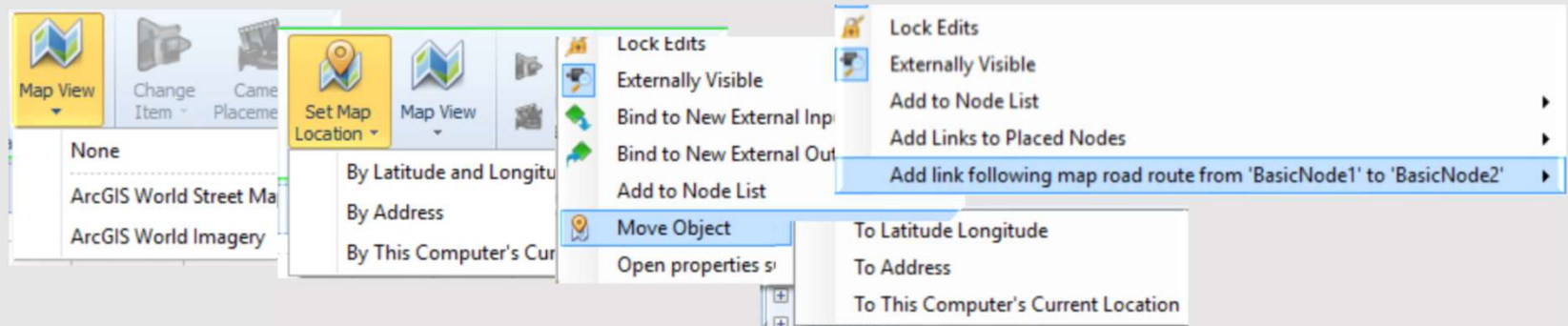
- 数以百万计的图形
- 图片质量可变
- 小心那些大图形
- 尽量采用低多边形



增加额外的图形



GIS地理信息系统

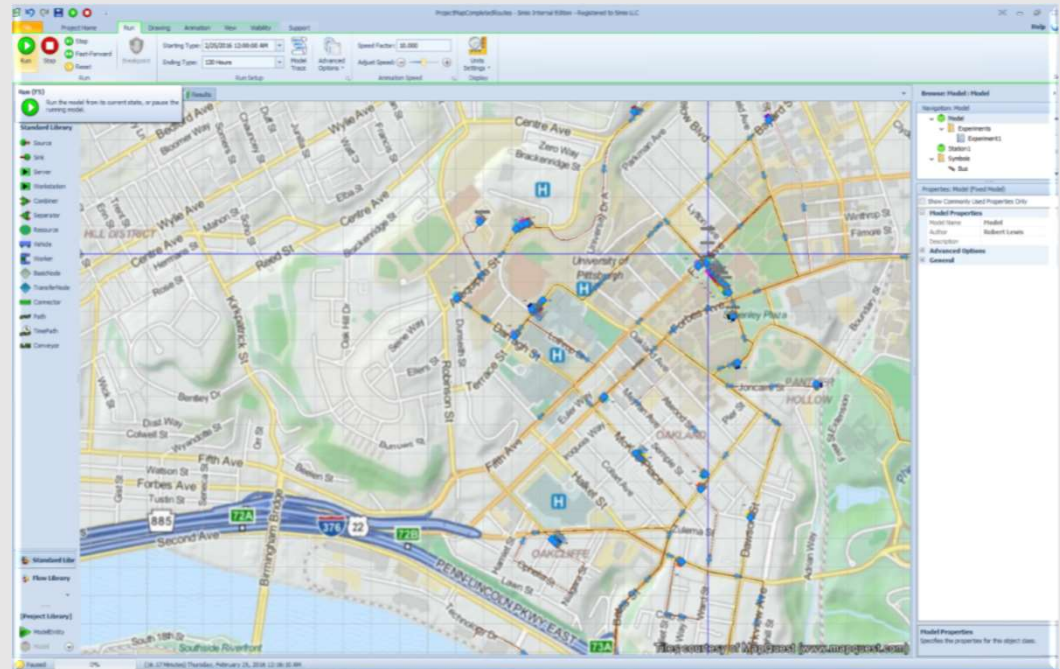


选择Map View (View 菜单下)

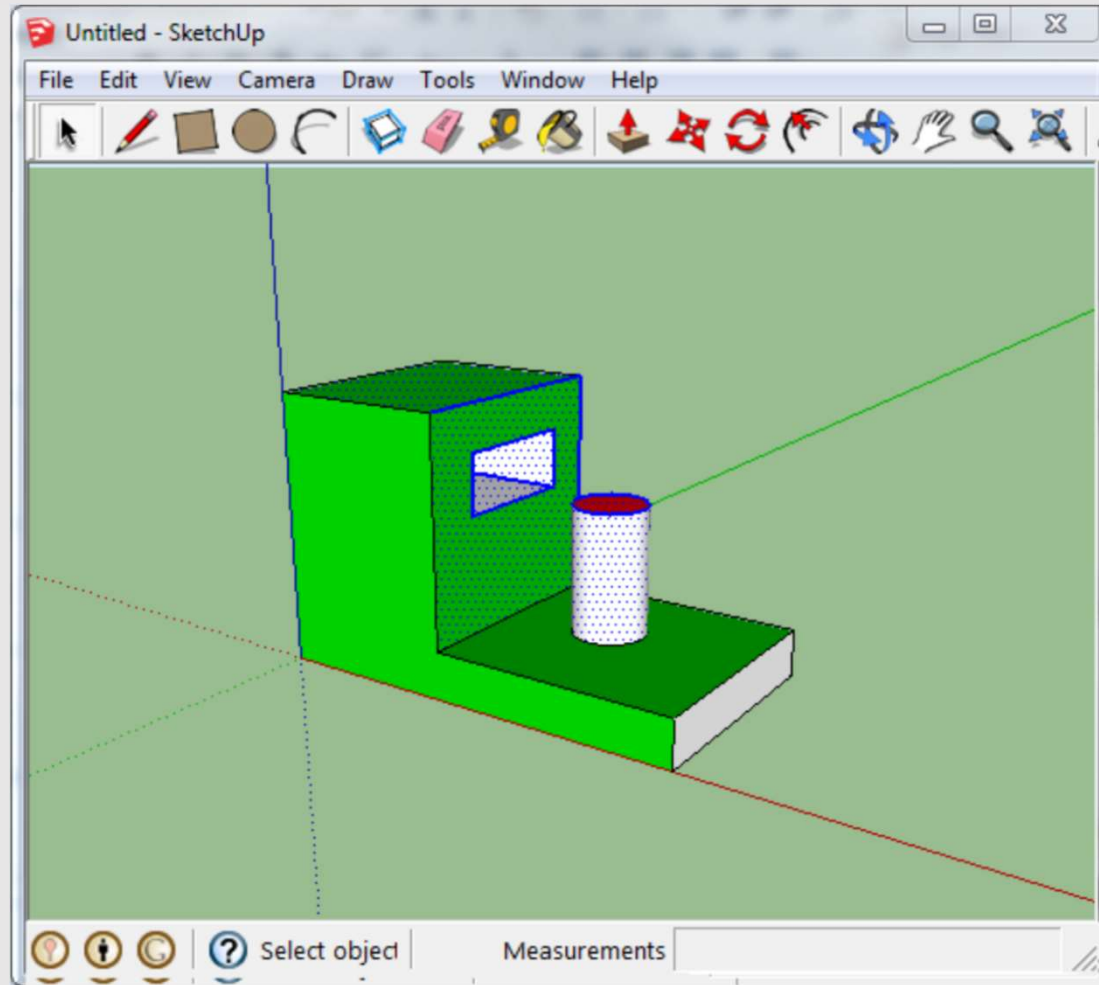
设定地图的位置

设定节点位置

高亮选中配对的节点，然后相连接。



创建3D图形-SketchUp软件



<http://www.sketchup.com/>

模型练习

- ▶ 给上述流水线模型增加动画元素，使其看上去更现实
 - 例如搜索某个领域，寻找适合的素材或制作3D素材，替换模型中的实体和服务。
- ▶ 增加以下信息的状态图：
 - ▶ 瞬时的系统内实体数
 - ▶ 系统内平均实体数
 - ▶ 实体在系统内的平均停留时间



第三章



Simio建模框架



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

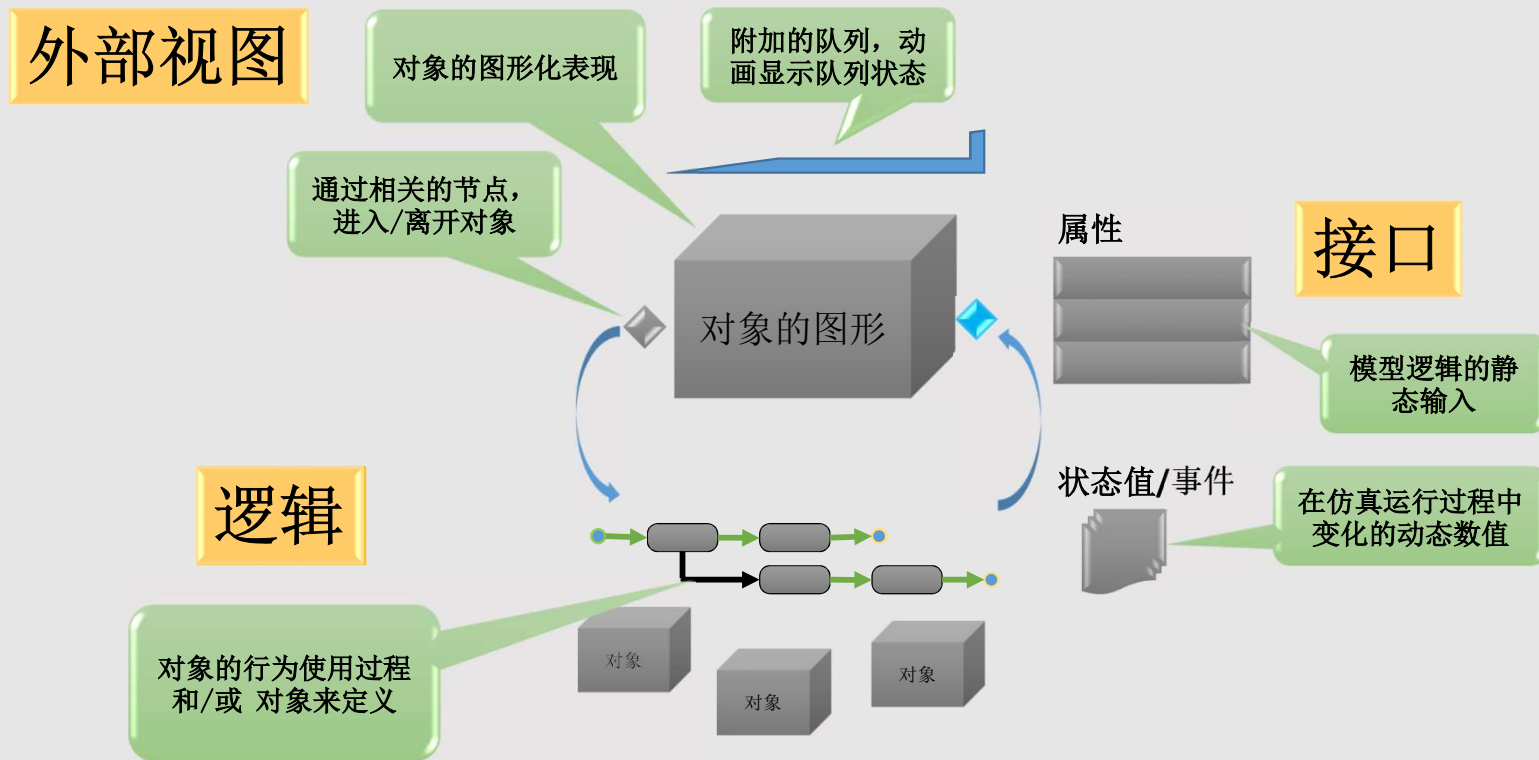
建模框架重点

- ▶ 面向对象的概念
- ▶ 对象类
- ▶ 对象作为资源
- ▶ **Station**（站点）元素
- ▶ 属性和状态变量

面向对象的概念

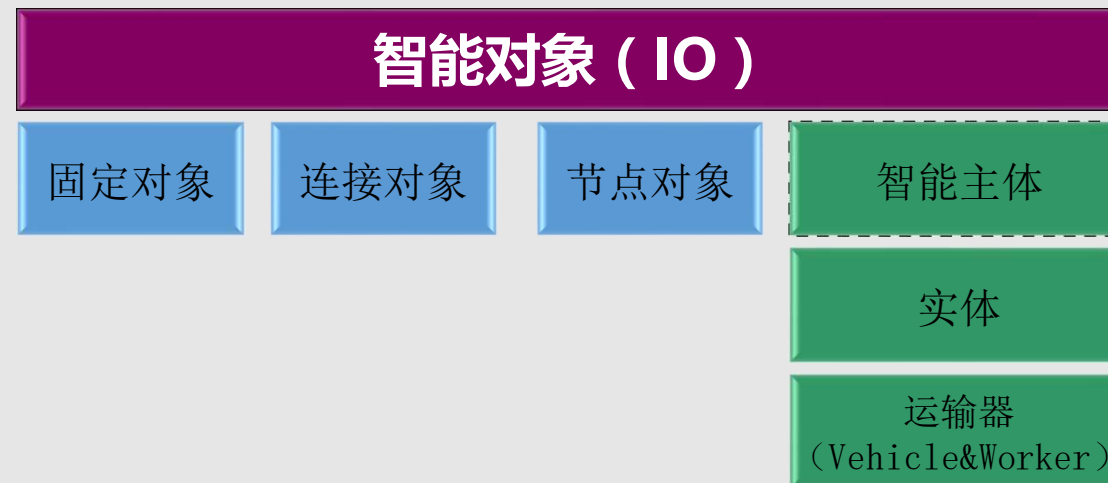
- ▶ 对象 是 对象定义 的实例。
- ▶ 对象被放入 模型 的设施 (Facility) 视图。
- ▶ 对象代表了系统的一个物理部分 - 比如客户, 工人, 设备, 汽车, 道路。
- ▶ 对象定义 在 对象库 或者 项目 中定义。
- ▶ 对象的行为是由它的定义部分所控制的, 如果定义改变, 那么那个定义的所有实例都随之改变。
- ▶ 对象定义是从 基础类 继承而来, 基础类定义其核心功能。

对象的剖析



对象类

- ▶ 所有的“对象定义”都从属于底层的类，它定义了对象的核心功能。
- ▶ 一个对象的模型在基类的基础上增加了特定的行为逻辑。



对象作为资源

任何对象都可以将其声明为“资源”

- 资源具有一个“能力”，可以被多个对象（实体）获取。
- 资源能力可以是固定值或随着计划而改变。

任何对象都可以获取其他对象的能力，将其作为“资源”使用。

- 当资源对象被释放后，它可以从其分配队列中选取第一个对象，重新被分配，或者根据某个规则，从所有等待的对象中动态选取一个。

更多关于资源

- ▶ 获取请求(Seize Request)被保存在资源对象的分配队列 (Allocation Queue)。
- ▶ 获取请求(Seize Request)可以被资源对象 (智能对象) 接受或者拒绝。
- ▶ 获取请求可以跟随访问请求(Visit Request)-这些是由资源对象来管理的。
- ▶ 资源在释放时可以被“预留”一段时间，方便重新抓取相同资源。



实体可以在网络上，自由空间里移动，或者访问某固定对象。

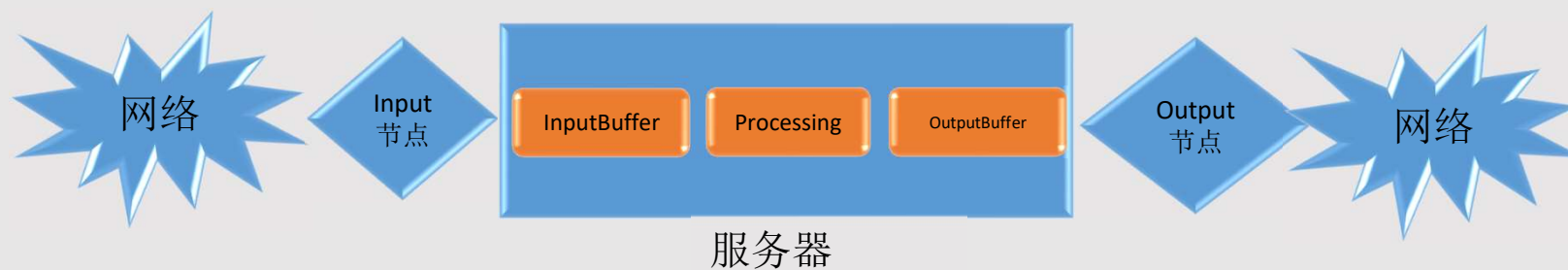
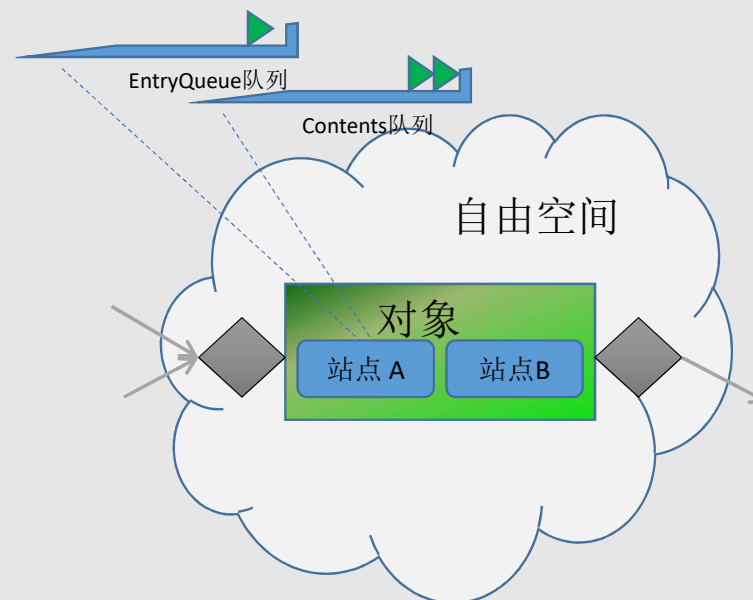
访问的实体位于对象内部的站点。

实体从对象的关联节点 进入/离开。

站点具有容量限制，能够限制过多实体进入站点。

实体在**EntryQueue**（进入队列）等待，并逐一被转入**站点**。一旦转入站点后他们停留在**Contents** 队列。

站点



属性和状态变量

属性是静态特征，在运行时不能被改变。

- 属性的例子有：加工时间，期望故障率，初始速度，期望的生产批量，每小时成本等。

状态变量是动态属性，可以在运行过程中动态改变（赋值）。

- 状态变量的例子有：加工计数，宕机次数，当前速度，批量，累积成本等。

属性



属性是“强类型”，它有很多种数据类型：

标准 – 例如整数Integer, 实数Real, 表达式Expression, 布尔型Boolean ... (21)

元素 (Element) – 物料元素, 统计元素, ... (20)

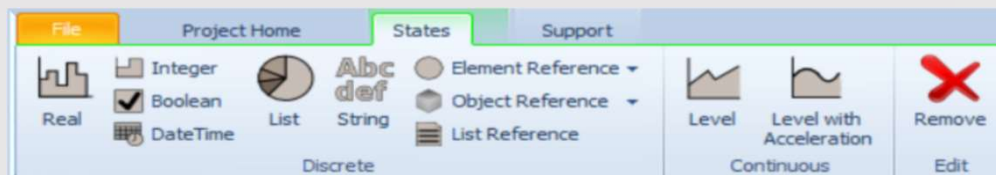
对象 (Object) – 通用或者特定的对象引用(7)

重复组 (Repeat Group) – 任何上述类型的重复集合

属性虽然在运行时不能改变定义，但

一个表达式属性可以包含随机变量和动态的状态变量。虽然属性是“静态”的,但它在每次被评估时,可能返回不同的值。

状态变量



▶ 状态变量可以取下面的这些类型：

▶ 离散（Discrete）

- 在离散的时间点上被指定新的值
- 可以采用标量或者数组（可扩展到10维）
- 类型有 *Real*（实型），*Integer*（整型），*Boolean*（布尔），*DateTime*（日期时间），*List*（列表），*String*（字符串），*Element Reference*（元素引用）和 *Object Reference*（对象引用）

▶ 连续变化（Level）

- 根据变化率随着时间推进连续地变化
- 变化率不为0

▶ 变化率及加速度（Level with Acceleration）

- 根据某个变化率和加速度，随着时间推进连续地变化
- 值、变化率、加速度可以离散地改变

属性和状态变量

- ▶ 任何对象 可以拥有预定义的属性和状态值，和用户添加的属性和状态值。
- ▶ 实体对象的状态变量和属性有时候被称作实体的“特性”-Attributes 。
- ▶ 模型的状态变量和属性参数有时候被称作“全局变量”。
- ▶ 状态变量可以在其他对象定义的内部设定，比如：
 - ▶ 服务器上产生的应付费用
 - ▶ 车辆对象的移动总时间

Simio操作符

- ▶ == (等于)
- ▶ > (大于)
- ▶ < (小于)
- ▶ >= (大于等于)
- ▶ <= (小于等于)
- ▶ && (与)
- ▶ || (或)
- ▶ ! (非), 例如: !IS.Red (不是红色)
- ▶ != (不等于)
- ▶ 布尔型 (True/False): 真/假
- ▶ IS.InstanceName (是否是某个实例) IS.Worker1[1]
- ▶ IS.DefinitionName (是否是某类对象定义) IS.Worker

数学符号 / 函数

▶ 运算操作符

`+, -, *, /`

`^` (幂), (2^3 等于8)

▶ Math.(许多函数):

`Math.If()`

`Math.Round()`

▶ DateTime.(许多函数):

`DateTime.Month()`

`DateTime.DayOfWeek()`

`DateTime.ToString(TimeNow, "YYYY/MM/DD hh:mm:ss")`

▶ 查询Simio帮助文档 (F1):

搜索 `Functions in Simio - Automatic`

标准库对象

- ▶ 固定对象
 - ▶ Source, Server, Sink, Combiner, Separator, Resource, WorkStation
- ▶ 连接对象
 - ▶ Connector, TimePath, Path, Conveyor
- ▶ 节点对象
 - ▶ BasicNode, TransferNode
- ▶ 运输器对象
 - ▶ Vehicle, Worker

模型实体 (ModelEntity)

- ▶ 默认的实体对象是模型实体 ModelEntity 类的一个实例，新建模型时被自动加入到项目文件中。第一个被生成的 ModelEntity 类型实例叫 “DefaultEntity”。
- ▶ 默认的实体对象在项目库中定义，比在标准库中定义更加方便，用户可以修改默认实体的定义，添加图形、状态变量、属性和逻辑。
- ▶ 当您需要编辑默认实体的尺寸和外张图片时，才需要将其拖动到 Facility 视图。

第三章总结

- ▶ 对象定义（**Object Definitions**）是在项目中定义或者对象库中定义。
- ▶ 对象是：
 - 对象定义的实例（ Instances ）。
 - 从五种基础类中衍生而成。
 - 可以被其他对象抓取/释放的资源。
 - 智能 – 智能对象可以和其他对象进行互动。
- ▶ 标准对象库提供了一系列通用型的对象定义，它们可以适用于广泛的应用需求。



第四章



Simio固定对象



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

大纲

- ▶ 标准对象库固定对象的介绍
- ▶ **Source**（实体发生器），**Sink**（销毁器），**Server**（服务器）
 - 模型练习 – **Agony**机场购票
- ▶ **Combiner**（合并器），**Separator**（分离器）
 - 模型练习 – 合并和分离
- ▶ **Resource**（资源），**Workstation**（工作站）
- ▶ 物料元素, 切换
- ▶ **Balking**（止步），**Reneging**（中途退出）
 - 练习– 止步，中途退出
- ▶ 任务序列基础知识
 - 练习– 任务序列

固定对象

基础对象类型,一般用于建立对象和模型。

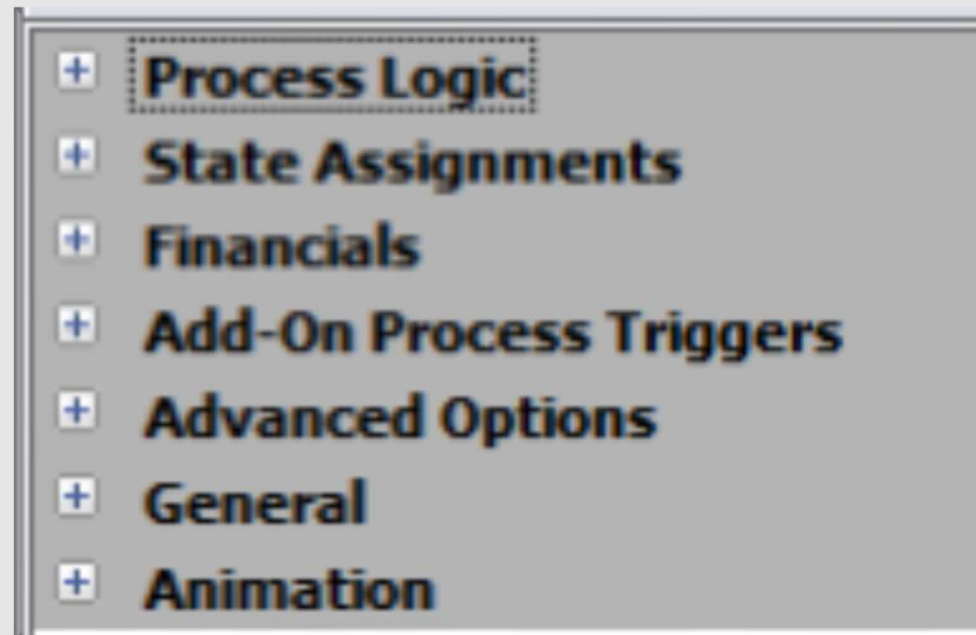
在运行过程中保持静态的位置。

也许使用资源能力。

我们很多库对象都是基于固定对象类。

通用的对象属性组

- ▶ 大多数库对象包含通用的属性组—有一些是显而易见的，另一些是更为高级的：



实体发生器



- ▶ 动态实体从发生器产生，在OutputBuffer站点等待被转移到输出节点的网络上。
- ▶ 属性包括了 *Entity Type* (实体类型), *Arrival Mode* (到达模式), *Entities per Arrival* (每次到达实体数) 和 *Stopping Conditions* (停止条件)。
- ▶ 到达模式包括: *Interarrival Time* (到达间隔), *Time Varying Arrival Rate* (时变的到达率), *On Event* (按事件产生), *Arrival Table* (到达时间表)。
- ▶ 根据不同的到达模式来创建动态实体。
- ▶ 实体图像可以使用随机的图形，只需要给实体的实例增加额外的图像，并设定属性 *Random Symbol* 为 *True*。
- ▶ 使用数据表产生多种不同的实体类型。
- ▶ 能产生批量到达，限制产生和搜集统计数据。
- ▶ 没有输入节点，只有一个输出节点。



实体销毁器



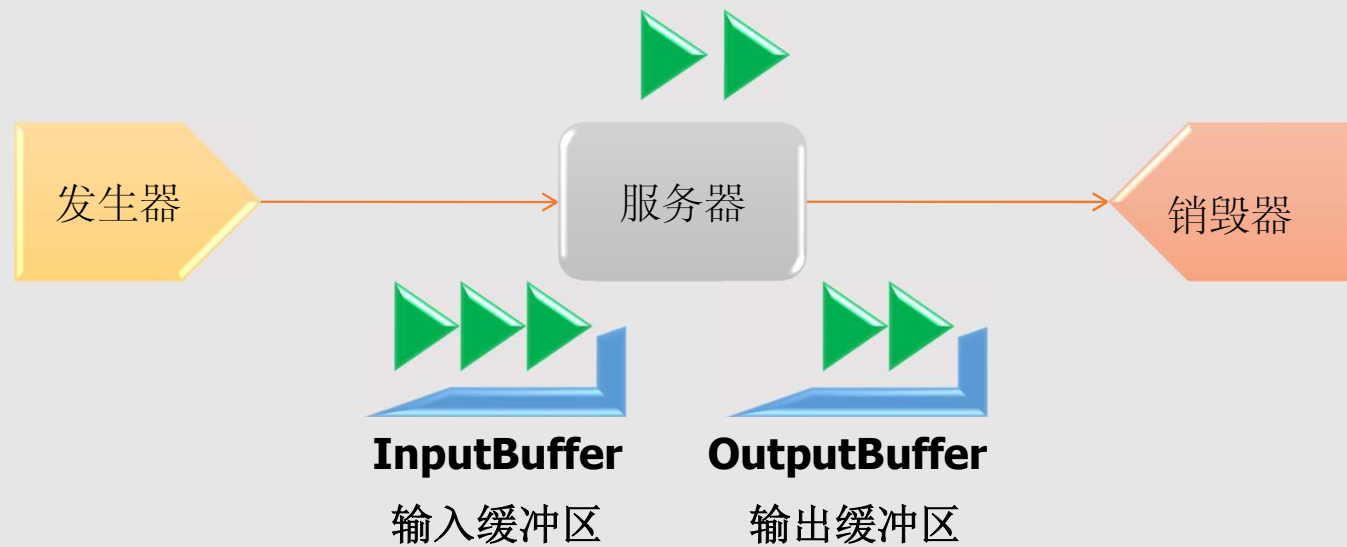
- ▶ 动态实体从Sink的InputBuffer站点离开系统。
- ▶ 自动搜集实体离开系统的数据 (*NumberEntered, NumberExited*), 和实体在系统中的时间*TimeInSystem*。
- ▶ 有额外的状态赋值。

服务器



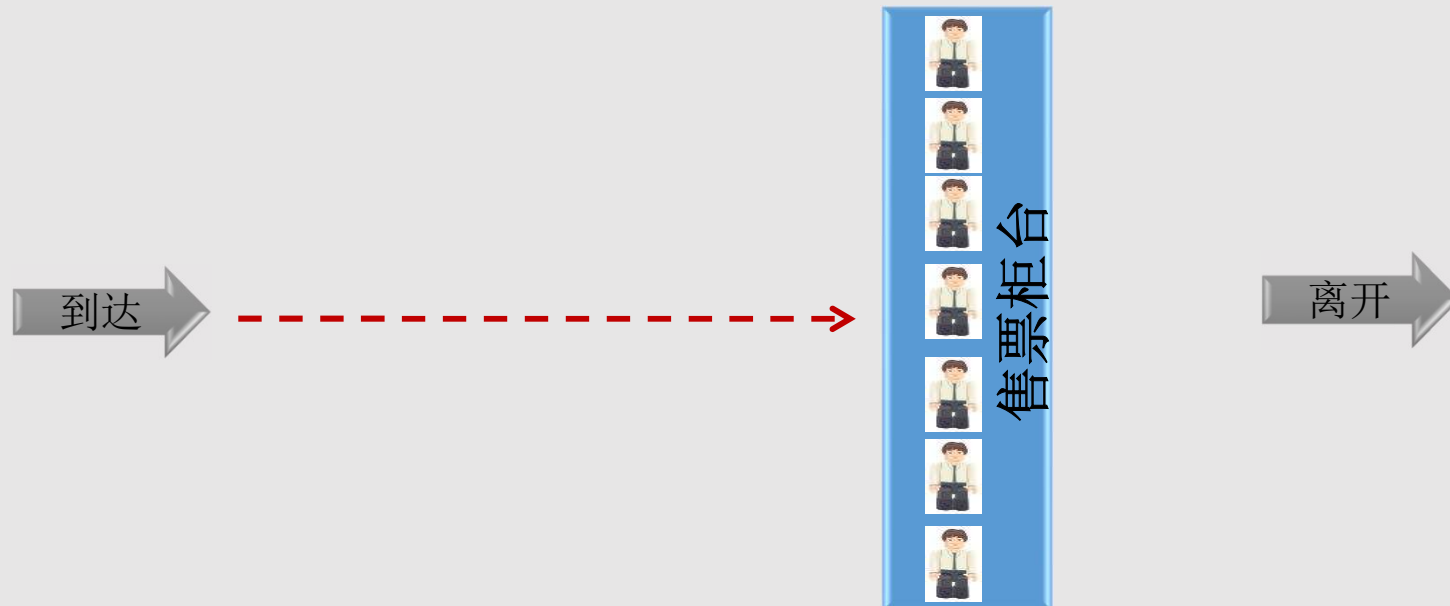
- ▶ 到达的实体占用/释放服务器。
- ▶ 服务器的分配队列 (AllocationQueue) 是排序的。选择的规则是按照队列中的第一个或者动态选择规则 (*Dynamic Selection Rule*)。
- ▶ 服务器具有固定的处理能力，或者服从某个工作计划，都有 OffShift 规则 (完成工作或者暂停工作)。
- ▶ 加工类型 (Process Type) 可以是特定时间或任务序列。
- ▶ 包含了稳定性 (比如故障)，服务器的失效模式有 *Calendar Time Based* (基于日历时间的)，*Processing Time Based* (基于加工时间的)，*Processing Count Based* (基于加工个数的) 和 *Event Count Based* (基于事件计数的)。
- ▶ 加工过程中用到附属资源，附属资源可以被预留并在后面重复利用。

实体发生器-服务器-销毁器



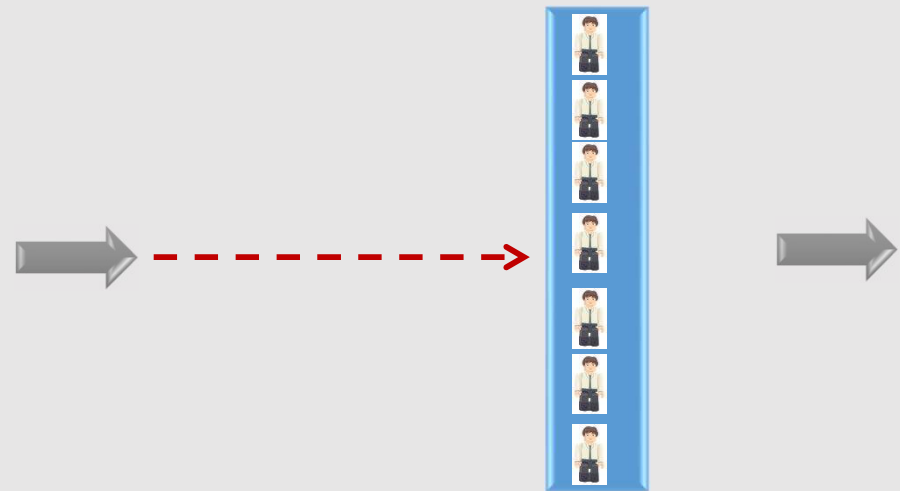
练习-Agony机场购票模型

- ▶ Agony机场的售票大楼有一个售票区域，需要了解每天需要多少票务人员服务。



练习-Agony机场购票模型

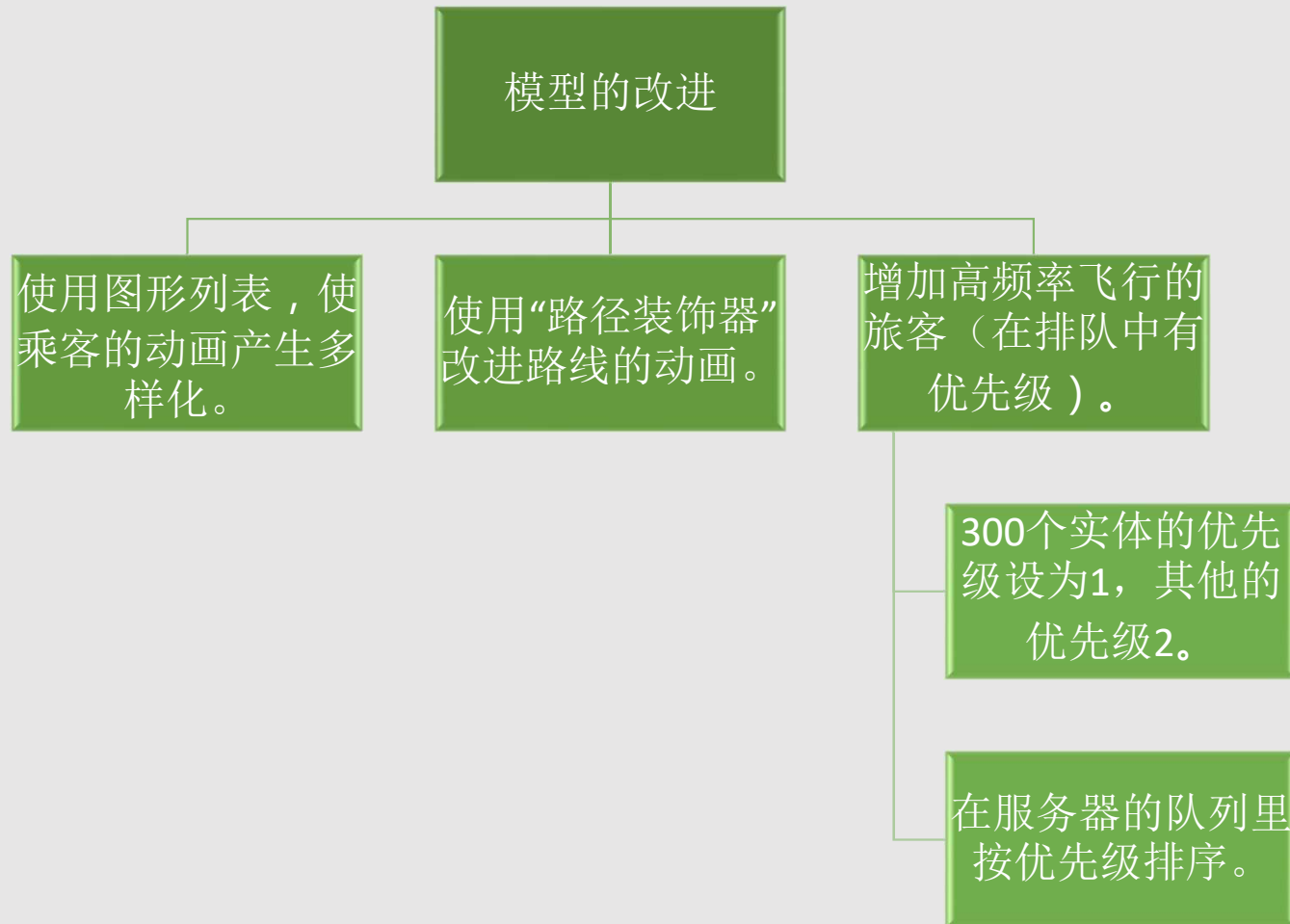
- ▶ 一天最多来800个乘客，平均每1.2分钟（指数分布）的到达间隔，需要花费1-3分钟（均匀分布）走到售票处。
- ▶ 售票服务时间大概花1-10分钟（均匀分布）。
- ▶ 在售票处分别用5，6，7个票务人员来衡量旅客在系统的停留时间，票务人员的利用率。
- ▶ 运行模型直到所有实体被处理。



Agony练习的提示

- ▶ 使用TimePath来模拟时间已知的移动过程。
- ▶ 用“服务器”代表售票过程。
- ▶ 设定模型Ending Type为无限（Infinity）。
- ▶ 鼠标右击Initial Capacity属性创建一个新的引用属性，用于指定服务器的服务能力。
- ▶ 建立仿真实验，以实体TimeInSystem属性和售票人员利用率为响应变量。
- ▶ 分别建立售票人员为5，6和7个的不同方案。

Agony机场模型的改进



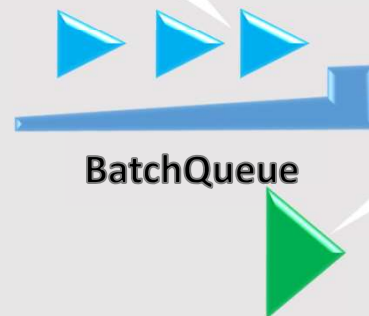
合并过程

一组成员实体可以被合并，然后被父实体携带。

合并的成员可以从合并批次中单独拆分出来。

父实体也能被合并。

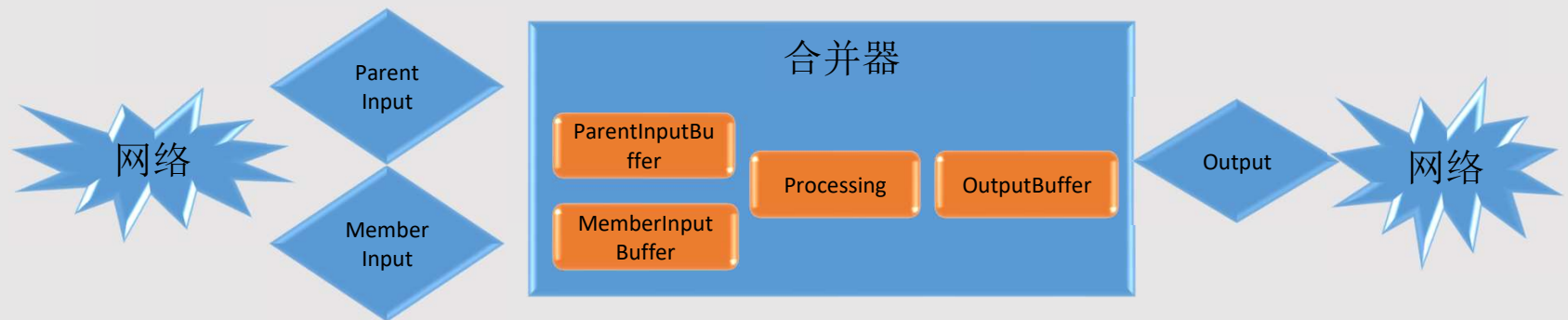
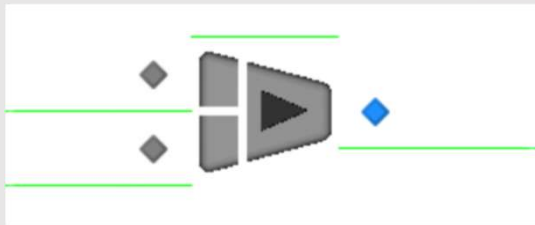
成员实体被合并



父实体携带在合并队列中的成员实体

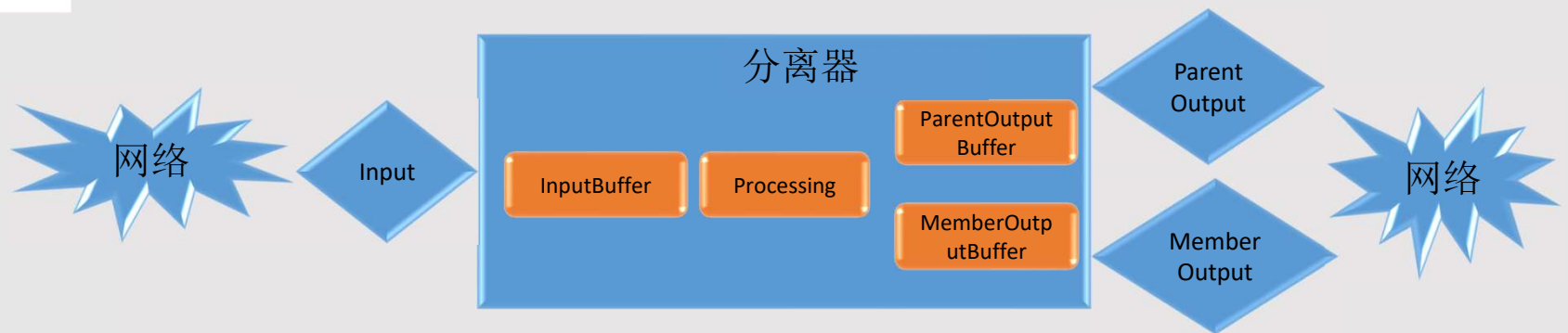
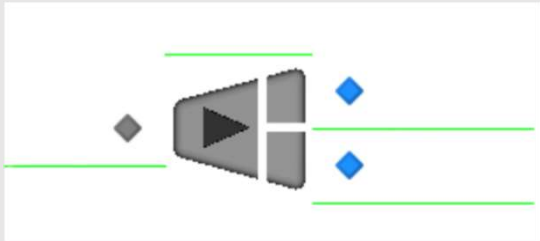
合并器

- ▶ 和服务器的相似功能，是具有合并逻辑的服务器。
- ▶ 合并器为父实体和成员实体提供单独的进入点
- ▶ Simbits模型：
 - ▶ Regenerating Combiner
 - ▶ Reneging Combiner

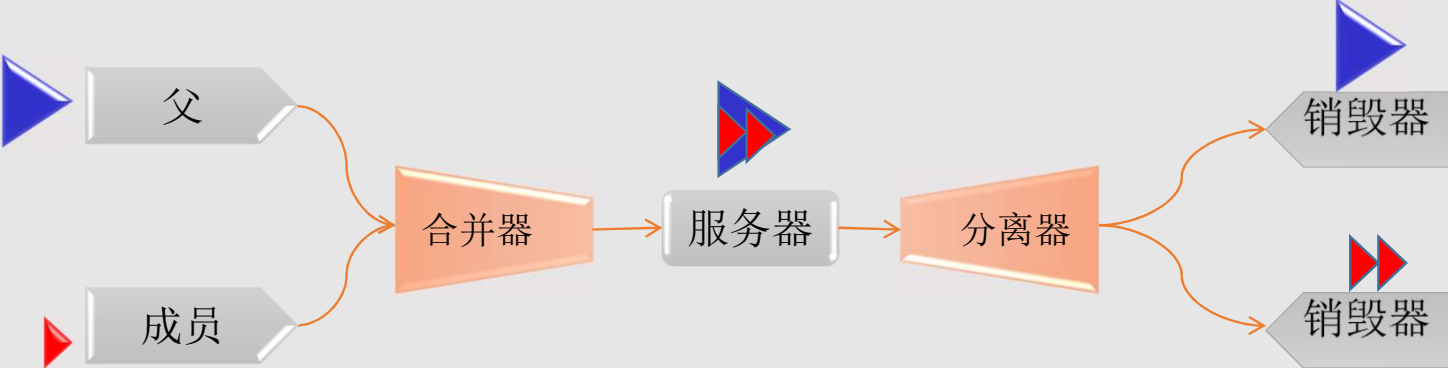


分离器

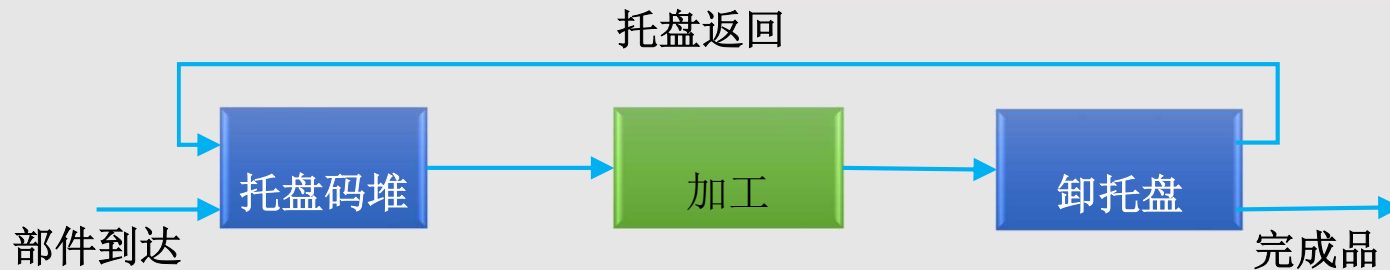
- ▶ 分离器为离开的父实体和成员实体提供了各自的输出节点。
- ▶ 允许成员实体从“父”实体上分拆出来，成员实体也能从父实体和其他实体类型上复制。
- ▶ 和服务端相似的特点，是具有分离/克隆逻辑的服务器。



合并然后分离



练习-合并和分离



▶ 部件到达流程

- ▶ 到达间隔：均匀分布（0.75~1.25）分钟（平均每分钟1个）
- ▶ 部件加工：对于托盘上的每一个实体的加工时间：指数分布，均值0.9分钟（每分钟1.11个部件）

▶ 码堆机和拆托盘机的处理时间：每托盘是均匀分布（10~20秒）

▶ 托盘返回

- ▶ 15分钟

▶ 模型运行100小时

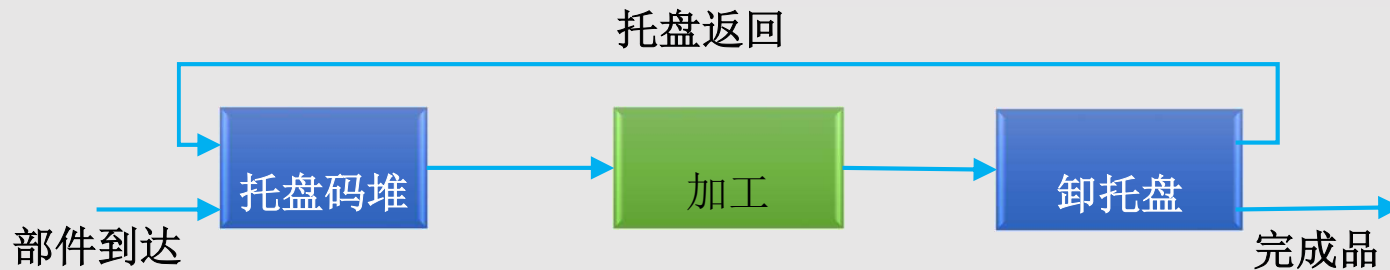
▶ 决策：

- ▶ 托盘的数量（固定数量的托盘周转）
- ▶ 托盘的装载能力产生的成本（装2个，4个，6个的成本分别是\$200，\$400，\$600）
- ▶ 最大化利润，如果每个完成部件的利润为\$1

▶ 已知要求：

- ▶ 最小产出：
每小时55个部件
- ▶ 最大的平均系统停留时间：1小时

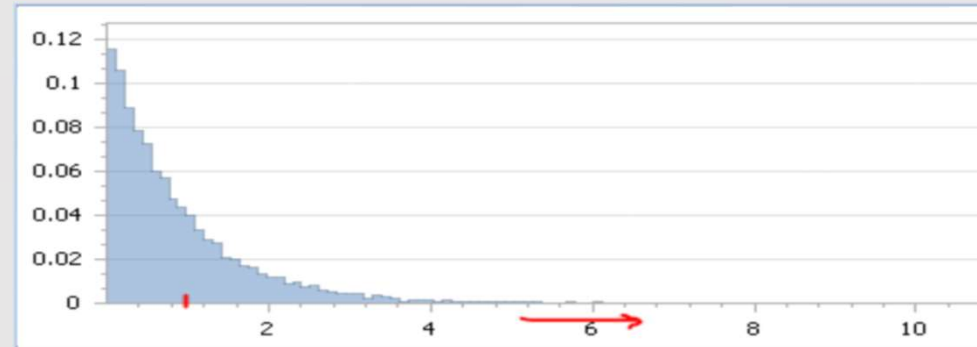
练习思考



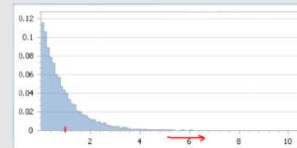
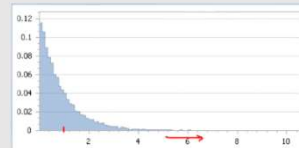
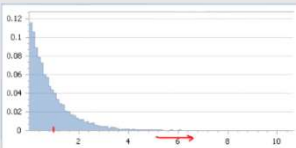
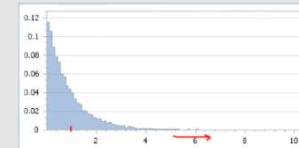
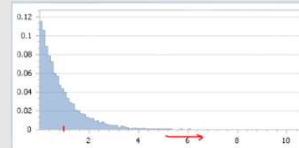
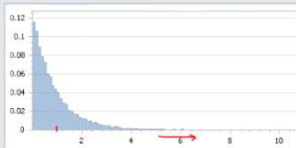
- ▶ 您要实验哪些控制变量?
 - ▶ 您如何衡量结果?
 - ▶ 如何确定是“最佳”的结果?
 - ▶ 部件如何到达?
 - ▶ 托盘（载具）如何到达?
 - ▶ 您如何停止模型的运行?
 - ▶ 您如何计算成本?
 - ▶ 您如何计算利润?
 - ▶ 加工延迟是多久?
- ▶ 决策:
 - ▶ 托盘的数量 (固定数量的托盘周转)
 - ▶ 托盘的装载能力产生的成本（装2个, 4个, 6个的成本分别是\$200, \$400, \$600）
 - ▶ 最大化利润，如果每个完成部件的利润为\$1
 - ▶ 已知要求:
 - ▶ 最小产出:
每小时**55**个部件
 - ▶ 最大的平均系统停留时间: **1** 小时

6个样本累加 不等于 一个样本乘以6!

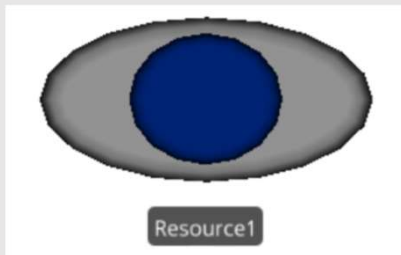
Random.Exponential(.9)*
PalletCapacity



- $\text{Expo}(.9) + \text{Expo}(.9) + \text{Expo}(.9) + \text{Expo}(.9) + \text{Expo}(.9) + \text{Expo}(.9) \dots$ 或
- **Random.Erlang(.9*PalletCapacity, PalletCapacity)** 或
- **Math.SumOfSamples(Random.Exponential(.9), PalletCapacity)**

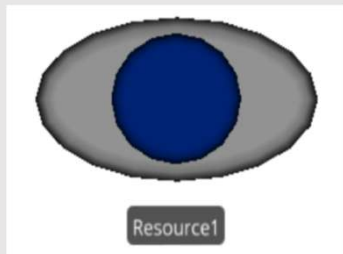


资源对象



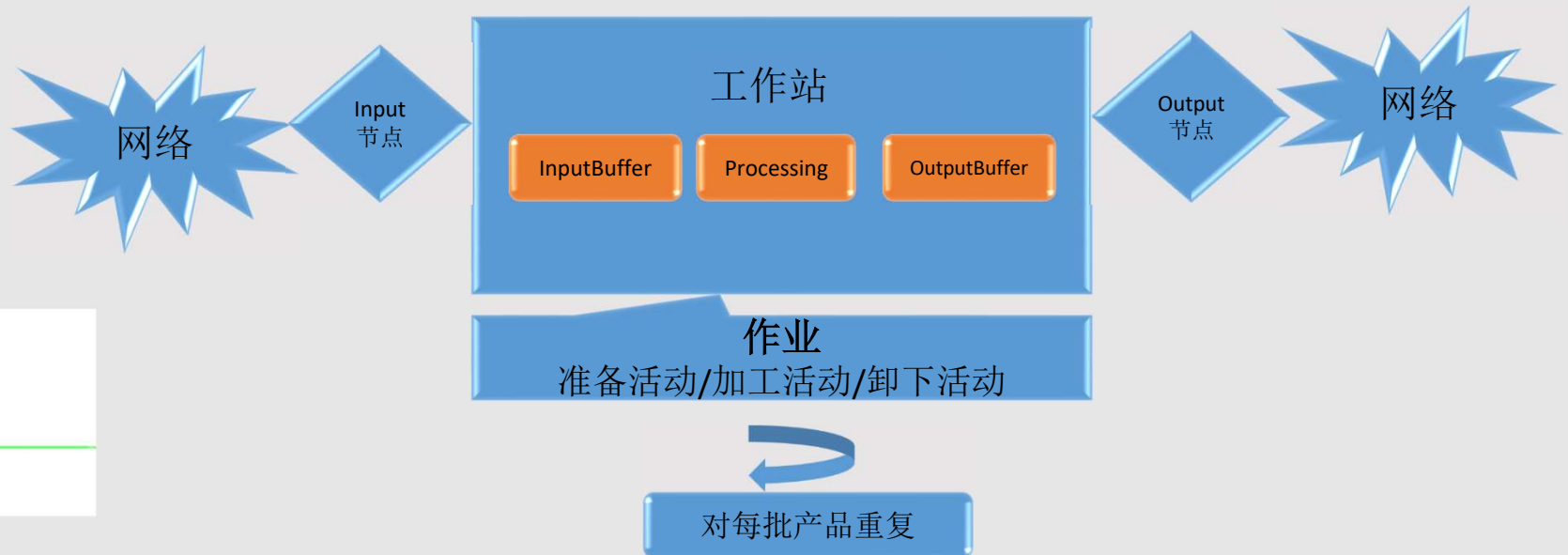
- ▶ 提供了一个原型的对象类,用于约束实体的流动。
- ▶ 不允许透过对象流动,但是可以被获取或者释放。
- ▶ 包含了稳定性特征(比如故障)。
- ▶ 能搜集关键处理数据。

智能资源



- ▶ 静态的排列&选择
 - ▶ 基于状态值或者表达式的HVF/LVF
 - ▶ 例如：优先级，交付期，严重度
- ▶ 动态的排列&选择
 - ▶ 每次资源寻找工作的时候被评估
 - ▶ 例如：关键比率（即到截止期还剩余的时间/剩余的工作量），序列相关的准备时间。
- ▶ 评价分配（Evaluating Seize Request）
附加过程
 - ▶ 资源可以选择处于‘闲置状态’。

工作站



- ▶ 包含“物料和其他约束”和准备时间/加工时间/拆卸时间多个阶段的服务器。
- ▶ 包含每个阶段所需要的附属资源 (Secondary Resource)。
- ▶ 实体使用 *Operation Quantity* 和 *Processing Batch Size* 作为生产的批量，加工活动每次加工一个批次。
- ▶ “物料和其他约束”包括物料消耗/生产，最大生产周期 (*Maximum Makespan*) 等。
- ▶ 服务器中的任务序列功能更为灵活，可以替代工作站的大部分功能。

物料消耗 / 生产

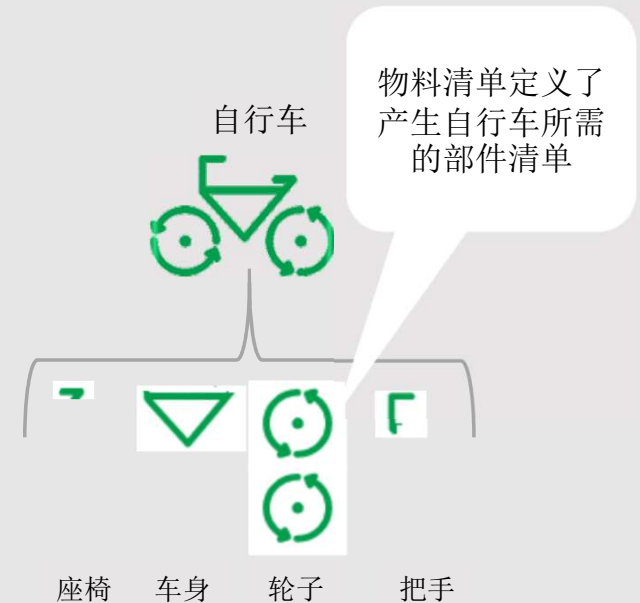
- ▶ 物料元素是用于表示离散的可消耗的资源，不需要动画表现。
- ▶ 物料可以被消耗（**Consume**）或生产（**Produce**）。
- ▶ 典型的，消耗的材料被指定为物料清单（**Bill of Material**）中的物料,生产的物料定义为“物料”元素。

SimBits模型:

WorkstationWithMaterialConsumption

WorkstationWithMaterialConsumptionAndReplenish*

ServersUsingTaskSequenceWithDataTablesJobShop



准备时间 / 切换时间

- ▶ 切换时间是根据作业属性 (Operation Attribute) 动态计算 (一般是List类型的值)
- ▶ ChangeOverLogic元素 (在Definitions标签页)
 - ▶ ChangeOver Matrix(切换矩阵)基于String值的List。
 - ▶ 灵活的From/To值类型, 可以是Specific, Any Change, Range。
- ▶ Setup效率。
- ▶ 同步的Setup支持。

从小到中, 准备时间为6.2

From/To	小	中	大
小	0.0	6.2	11.7
中	15.3	0.0	4.8
大	21.3	14.6	0.0

切换矩阵

Properties: ChangeoverLogic1 (Changeover Logic Element)

Show Commonly Used Properties Only

Basic Logic

Setup Transitions 1 Row

Setup Efficiency Multiplier

Assume Concurrent Setups If

General

切换逻辑元素

Setup Transition	
Operation Attribute	
Use Changeover Matrix	False
From Value Type	Range
Range Start	0.0
Range End	0.0
To Value Type	Any
Setup Time	0.0

Balking（止步）和Reneging（中途退队）

- ▶ **Balking:** 当实体到达时，就决定不进入本来要去的目的地。
- ▶ **Reneging:** 当实体进入了队列或者站点之后，它决定放弃服务过程，直接离开。
- 小组讨论，因为它们有近似的概念和行为。
- 在面向服务的系统应用中最为常见，如银行排队，呼叫中心。

***Simio**所有版本都支持这两类排队行为，自**9.149**版本增加了更为方便的功能。

Balking (止步)

当实体到达时，就决定不进入本来要去的目的地。

- ▶ 没有可以等待的空间
- ▶ 队列，站点或系统的容纳达到了极限
- ▶ 实体自己决定不等待

两个关键要素:

- ▶ 在什么条件下“止步”发生?
- ▶ 止步的实体发生了什么?

Reneging (中途退队)

当实体进入了队列或者站点之后，它决定放弃服务过程，直接离开。

- ▶ 达到了等待的容忍限度
- ▶ 一个触发时间发生了
- ▶ 实体自己决定要移动到其他地方

两个关键要素:

- ▶ 在什么条件下“中途退队”发生?
- ▶ 退出队列的实体发生了什么?

在哪里指定?

- ▶ 标准库对象的输入或输出缓冲区
- ▶ 站点元素中 (和过程逻辑一起使用)

Buffer Logic	
Input Buffer	
Capacity	Infinity
Balking & Reneging Options	
Balk Decision Type	Conditional
Balk Condition Or Probability	0.0
Balk Node Name	None (Destroy Entity)
Reneged Triggers	0 Rows

None
Blocked
Conditional
Probabilistic

Reneged Trigger	
Trigger Type	Time Based
⊕ Wait Duration	0.0
Reneged Decision Type	Always
Reneged Node Name	None (Destroy Entity)

改变状态

- ▶ 当实体“止步”或“退队”后，您可以修改那个实体的状态变量或任何系统状态变量（State Assignments的On Balking, On Reneging）。

Properties: Server1 (Server)

Show Commonly Used Properties Only

+ Table Row Referencing

- State Assignments

On Entering	0 Rows
Before Processing	0 Rows
After Processing	0 Rows
Before Exiting	0 Rows
On Balking	0 Rows
On Reneging	0 Rows

+ Secondary Resources

+ Financials

+ Add-On Process Triggers

On Balking
Optional state assignments when an entity is balking at entering an input or output buffer of the object.

On Reneging
Optional state assignments when an entity is reneging from an input or output buffer of the object.

Balking/Reneging的统计数据

▶ 函数:

- ▶ ObjectName.BufferName.NumberBalked

 - ▶ Server1.InputBuffer.NumberBalked

- ▶ ObjectName.BufferName.NumberReneged

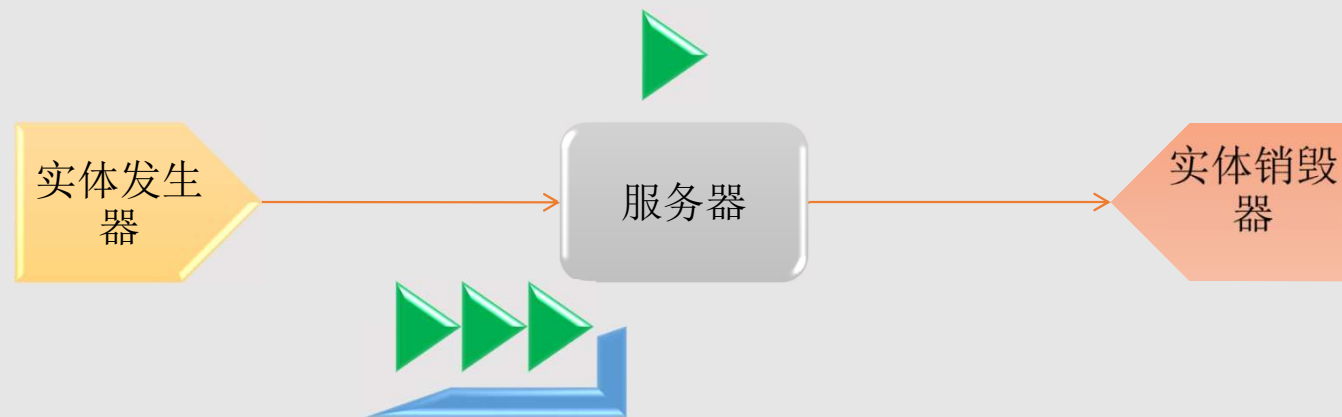
 - ▶ Server1.InputBuffer.NumberReneged

▶ 数据透视表输出

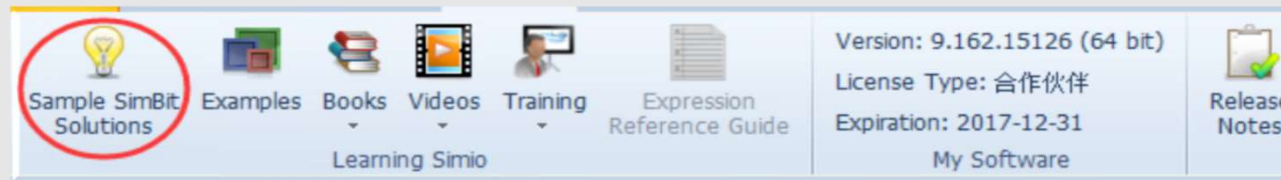
Object N...	Data S...	Category	Data Item	Statistic	Average Total
Server 1	InputBuffer	Content	NumberInStation	Maximum	7.0000
			TimeInStation	Average (Hours)	0.0914
		Maximum (Hours)		0.5846	
		Minimum (Hours)		0.0000	
		Throughput	NumberBalked	Total	4.0000
			NumberEntered	Total	378.0000
			NumberExited	Total	378.0000
NumberReneged	Total		51.0000		
Source 1	OutputBuffer	Throughput	NumberBalked	Total	328.0000
	Processing	Throughput	NumberEntered	Total	627.0000
			NumberExited	Total	627.0000

练习-Balking/Reneging

- ▶ 创建一个**Source-Server-Sink**模型，服务器的输入缓冲区的能力为4。
- ▶ 实体到达时如果队列满了，那么它将止步（被销毁）。
- ▶ 实体在队列里等待不超过**0.5**分钟，超过就中途离开队伍（被销毁）。
- ▶ 确定止步的数量和退队的数量。



Simbits: 止步和退队



- ▶ **BalkingOnSourceBlockingOnServer**
- ▶ **ChangingQueuesWhenServerFails**
- ▶ **ServerQueueWithBalkingAndReneging**
- ▶ **MultiServerSystemWithJockeying**

任务序列基础

Task Sequences Basics

赋予Server超能量!



任务的介绍

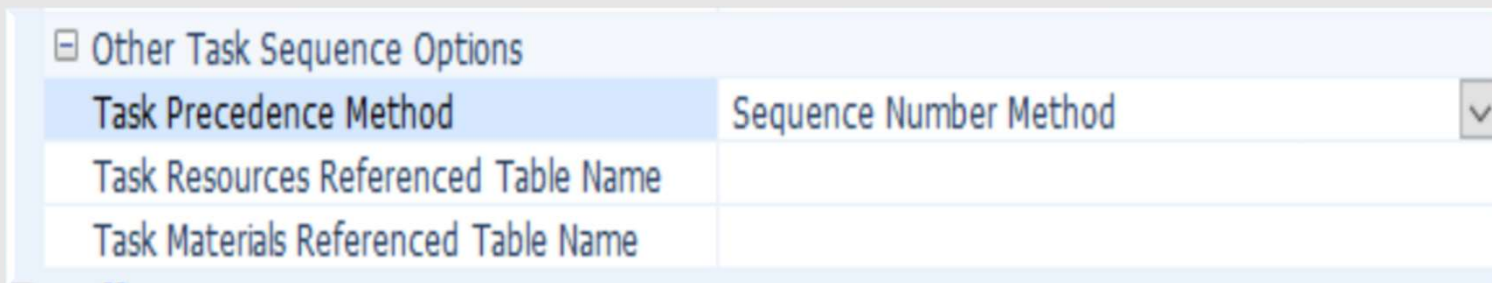
- ▶ 服务器（和合并器，分离器）模拟了一个具有有限资源的加工设备。
- ▶ 加工时间可以用表达式定义，或者定义成一系列要操作的任务。
- ▶ 服务器的Process Type（加工类型）属性：
 - ▶ Specific Time（特定时间）：输入加工时间的表达式
 - ▶ Task Sequence（任务序列）：执行任务的重复组。



- ▶ 任务可以使用附属资源和物料。

任务的介绍

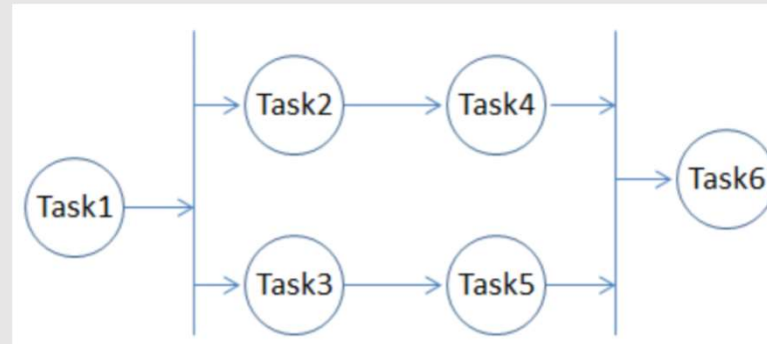
- ▶ 任务序列是由任务顺序方法控制的。
 - ▶ Sequence number（序列号）–最适合简单的任务序列
 - ▶ Immediate Successor（紧后活动）
 - ▶ Immediate Predecessor（紧前活动）
- ▶ 任务可以用关系型数据表中的数据驱动。



The screenshot shows a software interface with a table titled "Other Task Sequence Options". The table has two columns: "Task Precedence Method" and "Sequence Number Method". The "Sequence Number Method" column has a dropdown arrow. Below the table, there are two rows with labels: "Task Resources Referenced Table Name" and "Task Materials Referenced Table Name".

Other Task Sequence Options	
Task Precedence Method	Sequence Number Method <input type="button" value="v"/>
Task Resources Referenced Table Name	
Task Materials Referenced Table Name	

任务顺序方法



使用序列号

Sequence Number	Task Name
10	Task1
20.1	Task2
20.2	Task3
30.1	Task4
30.2	Task5
40	Task6

使用紧前活动

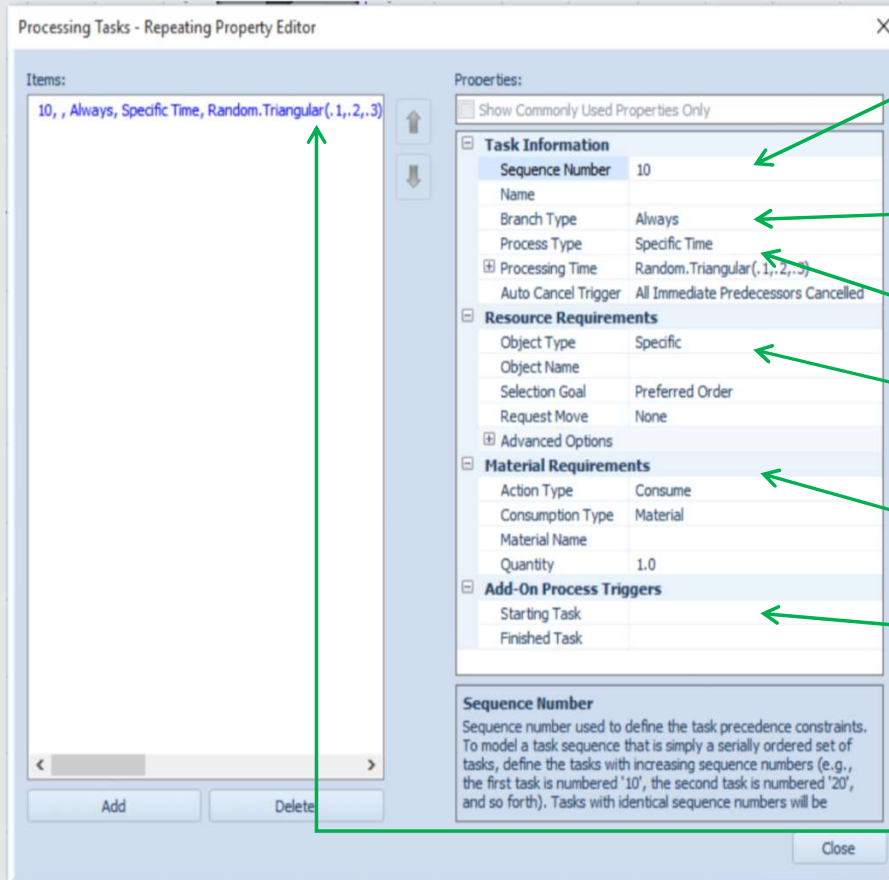
ID Number	Task Name	Immediate Predecessors
1	Task1	
2	Task2	1
3	Task3	1
4	Task4	2
5	Task5	3
6	Task6	4,5

使用紧后活动

ID Number	Task Name	Immediate Successors
1	Task1	2,3
2	Task2	4
3	Task3	5
4	Task4	6
5	Task5	6
6	Task6	

任务序列

► 服务器加工任务



Process Type	Task Sequence
Processing Tasks	1 Row

Task Number（任务号）确定任务的执行是串行还是并行的。简单序列是10, 20, 30, 40。

Branch Type（分支类型）允许我们执行选择的任务。

Process Type（加工类型）确定加工逻辑在哪里被定义。

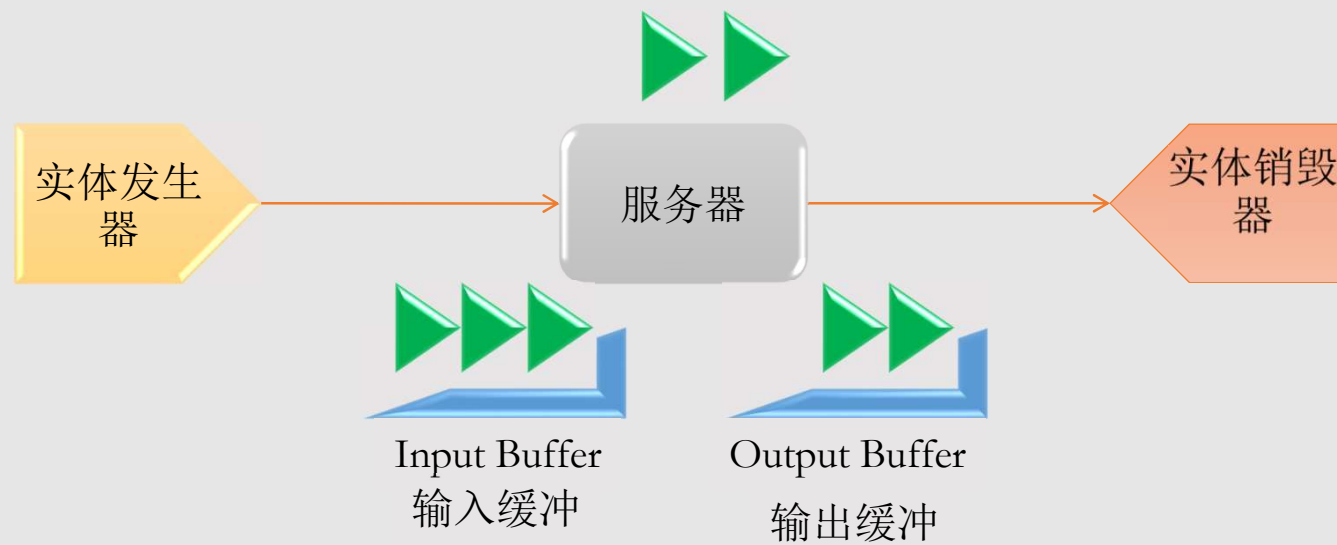
Object Type（对象类型）定义执行该项任务所需的任何资源。

Material Requirements（物料需求）定义了该任务所需的任何物料。

Add-on Processes（附加过程）可以在任务执行之前或之后，被直接执行。

Items（序列项目）重复组提供了串行和并行任务的集合。

Source-Server-Sink模型使用任务

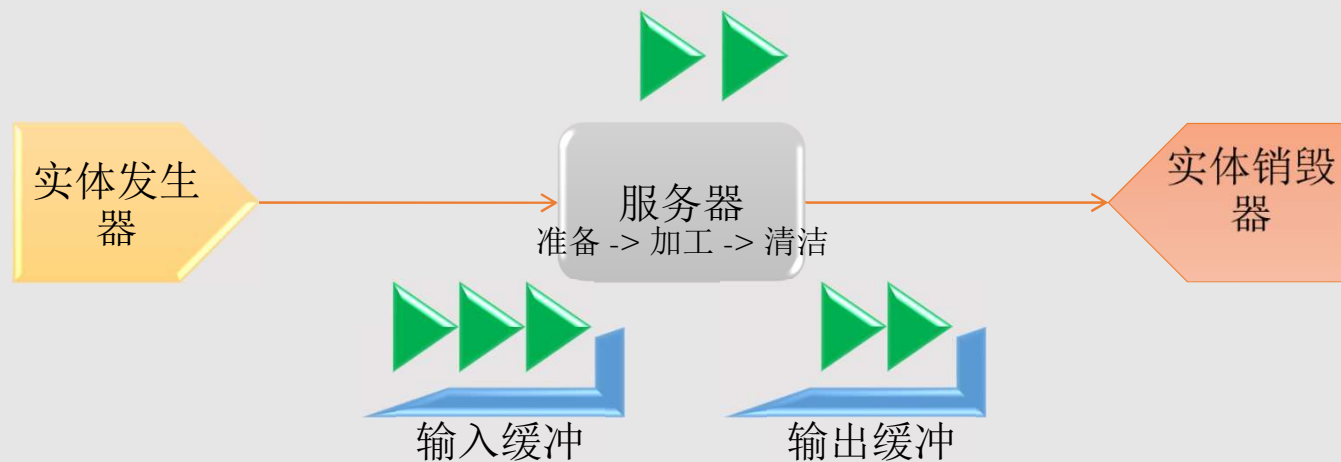


任务的分支类型

- ▶ 确定该任务是否要被执行。
- ▶ 四种分支类型 (**Branch Types**)
 - ▶ **Always** (永远)
 - ▶ **Conditional** (按条件) – 如果表达式为TRUE
 - ▶ **Probabilistic** (按概率) – 互斥的选择项
 - ▶ **Independent Probabilistic** (独立概率) – 相互独立的可能性。
- ▶ 默认某任务的前置任务没有被执行的话, 那么该任务也不会被执行。

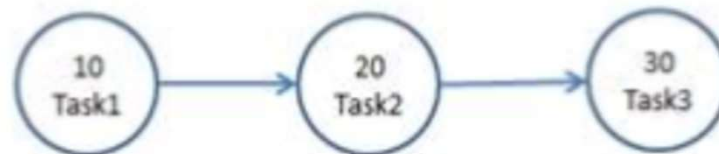
任务序列练习1

- ▶ 建立一个Source-Server-Sink模型，所有属性都使用默认值，除了：
 - ▶ 指定服务器能力为3
- ▶ 指定加工过程为3个串行的任务，代表准备，加工和清洁工作，设定为默认值，除了...
 - ▶ 准备和清洁任务需要工人
 - ▶ 清洁任务只有40%的时候需要
 - ▶ 有两个工人是可用的 (`Worker>Population>InitialNumberInSystem`)
- ▶ 挑战:让工人访问准备工作和清洁工作的节点。



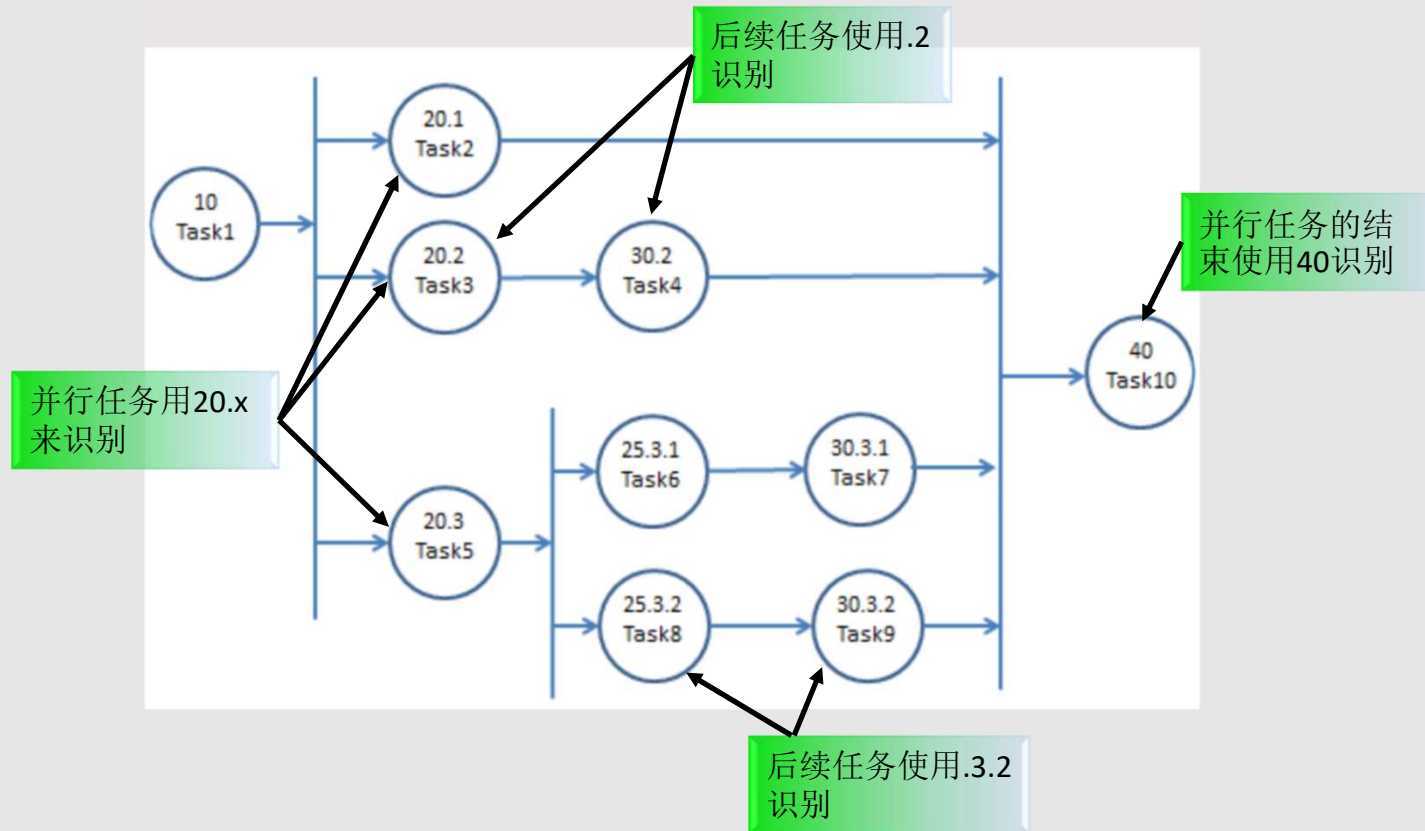
直线任务

Sequence Number	Task Name
10	Task1
20	Task2
30	Task3



更加复杂的任务序列

▶ 序列号“...YY”



任务序列 Simbits

▶ **ServerUsingTaskSequenceWithWorkers**

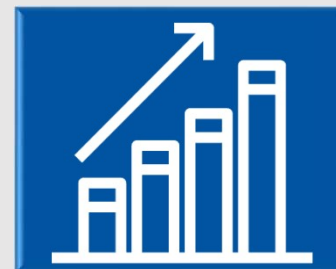
- 4个串行的任务

▶ **TaskSequenceAndWorkerInServer**

- Server 上的同步任务，使用1名工人
- 任务将工人移动到新的位置，然后回来
- 定义Server 中的任务

▶ **TaskSequenceAndWorkerInTable**

- 在Table 中定义任务



第五章



动态对象移动



(0510) 82132584



无锡迅合信息科技有限公司©



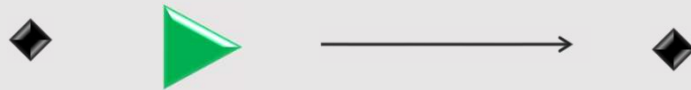
<http://www.simio-china.com/>

大纲

- ▶ 自由空间,连接对象,节点对象,网络
- ▶ **Connector** (连接线), **Path** (路径), **TimePath** (时间路径)
 - 练习 – 双向路径
- ▶ 路由决策
 - 连接权重 (**Link Weights**), 按序列进行路由,动态路径
 - 练习 – 连接权重
- ▶ 传送带 – 累积式, 非累积式
- ▶ 车辆对象 – 按需 (**On-Demand**), 固定线路(**Fixed Route**)
- ▶ 工人对象 – 静态任务, 非静态任务
 - 练习 – **Agony**机场和电车

自由空间运动

- ▶ 在模型中，实体可以使用自由空间移动功能，从一个位置移动到另一位置。
- ▶ 实体将以某个速度运动以直线的方式到达目的地，而无需使用任何预先定义的网络，也无需任何物料搬运工具。
- ▶ 实体的自由运动轨迹可以人为定义和控制（方向，角度，加速度，减速度等）。

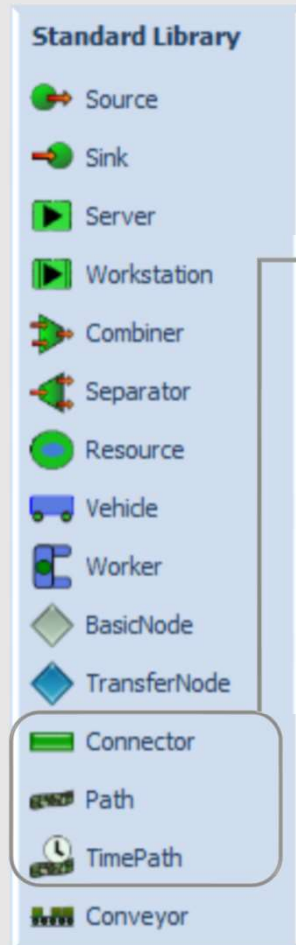


Tip 设定实体的网络为 'Free Space Only' 或 'Network if Possible' ，开启实体在自由空间移动模式。



Tip 在过程逻辑中可以使用Travel步骤实施高级的自由空间运动，它支持多个选项，包括和方向相关的设定，加速度和减速度等。

使用连线和路径进行移动



- ▶ **Connector** (连接线) – 用于定义一个直接的，0时间通过，0移动距离的连接线，连接两个位置。
- ▶ **Path** (路径) – 用于定义两个位置之间的通路，移动时间是由路径的长度和实体的移动速度决定的。允许或者不允许实体在路径上通过。
- ▶ **TimePath** (时间路径) – 用于定义两个位置之间的通路，移动总时间是明确指定的。



模型系统中的路径可以组合成网络 (Networks)。



可以使用Path decorator (路径修饰器) 美化任何路径或传输带。

连接线，时间路径和路径

Connector

- ▶ 0时间通过
- ▶ 选择权重

Time Path

- ▶ 移动时间
- ▶ 通路的容量
- ▶ 单向/双向
- ▶ 选择权重

Path

- ▶ 每个实体移动的速度，及限速
- ▶ 允许超越或不允许超越
- ▶ 根据实际比例尺
- ▶ 通路的容量
- ▶ 单向/双向
- ▶ 选择权重

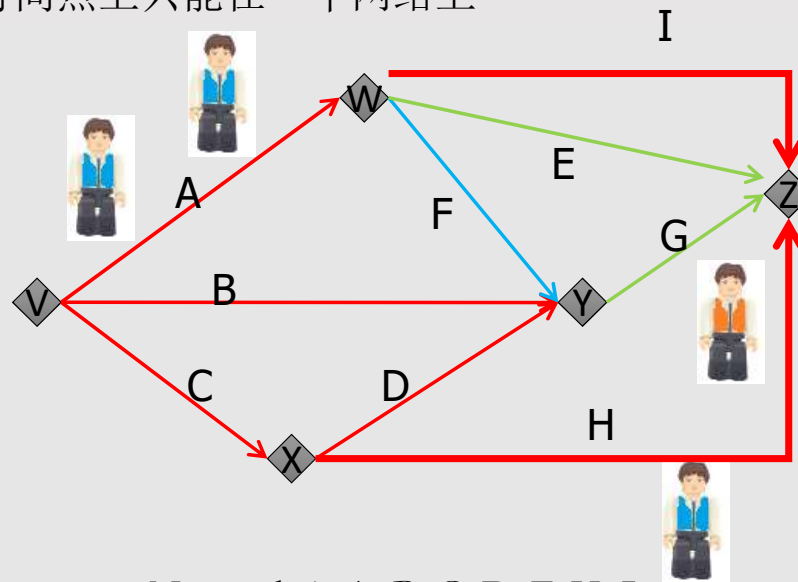
连接，节点和网络



- ▶ 连接对象（**Link**）是两个节点之间的通路。
- ▶ 一个节点（**Node**）可以有任意数量的进入连接和输出连接，它可以是对象的进入/离开点。
- ▶ 连接对象可以被加入一个或多个网络（**Networks**）。
- ▶ 实体/运输器可以移动：
 - ◆ 在由连接和节点组成的网络上
 - ◆ 进入和离开其他对象
- ▶ 在网络上的任意两个节点之间有最短路径。
- ▶ 从节点出发的路径选择可以采用最短路径或者按连接权重。
- ▶ 目的地可以是：
 - ◆ **Continue**（继续），**Specific**（特定），**Select from List**（从列表中选择），或**By sequence**（按照序列表）。

网络 (Network)

- ▶ 实体通过Link构成的网络,移动到目的地
- ▶ 连接(Link)模拟实体在固定的路径上从一点到另外一点的移动.
- ▶ 节点模拟2个或2个以上连接之间的交叉点
- ▶ 连接可以在多个网络中
- ▶ 实体在一个时间点上只能在一个网络上



Network 1: A, B, C, D, F, H, I

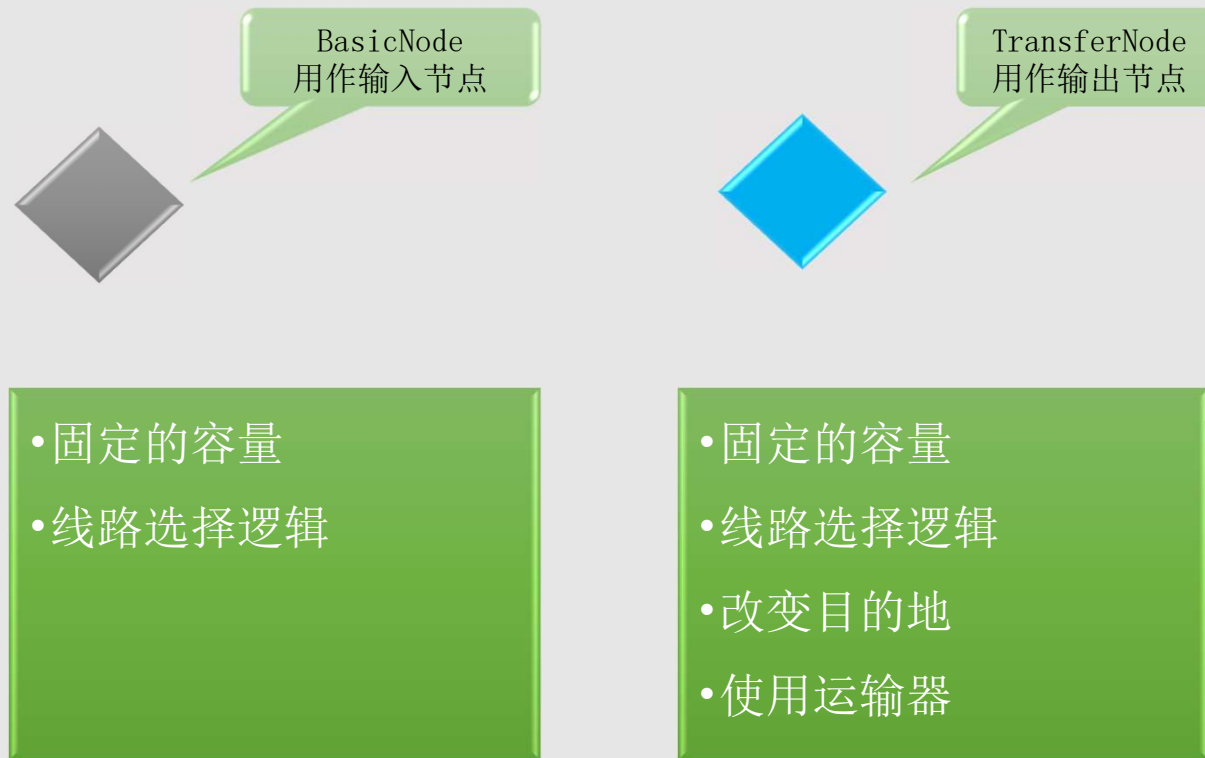
Network 2: E, F, G

节点对象

- ▶ 节点定义了一个或多个连接的起点和终点
- ▶ 节点可以模拟多个进出的连接形成的交点.
- ▶ 节点可以定义关联的固定对象的进/出点.
- ▶ 节点有一个停泊区,实体可以停留在这里而不会阻塞网络的交通
- ▶ 节点对象没有大小, 实体由于实体长度的关系是有通过时间。



标准库节点对象



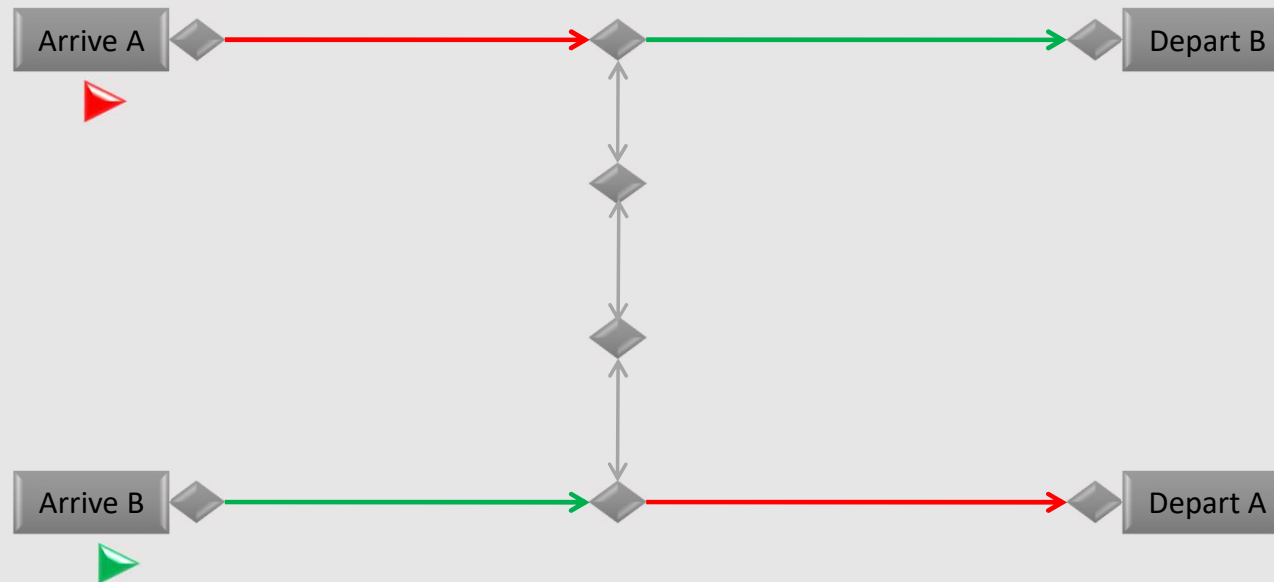
双向路径

- 双向路径允许一个时刻往一个方向移动。

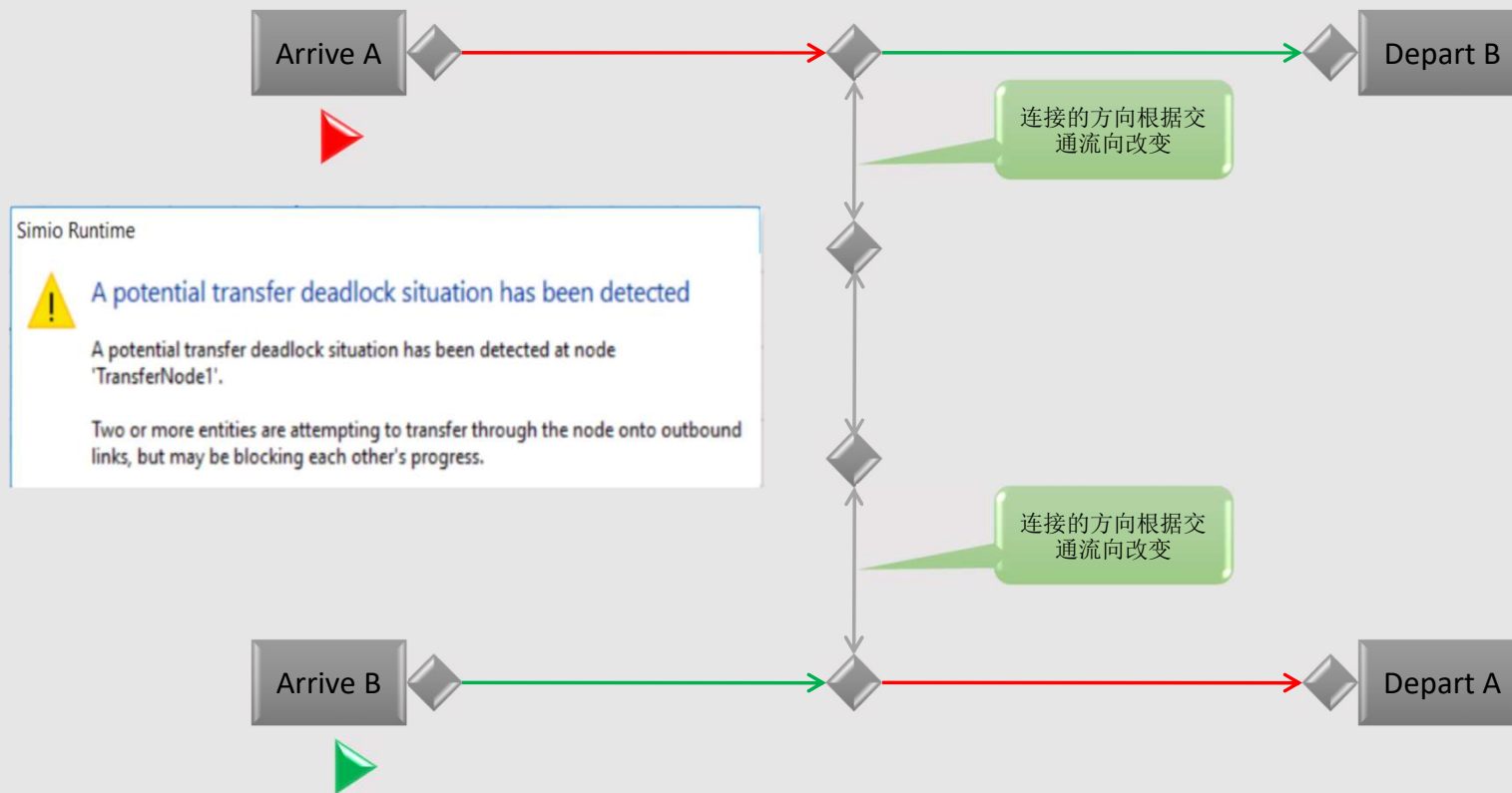


练习-双向路径

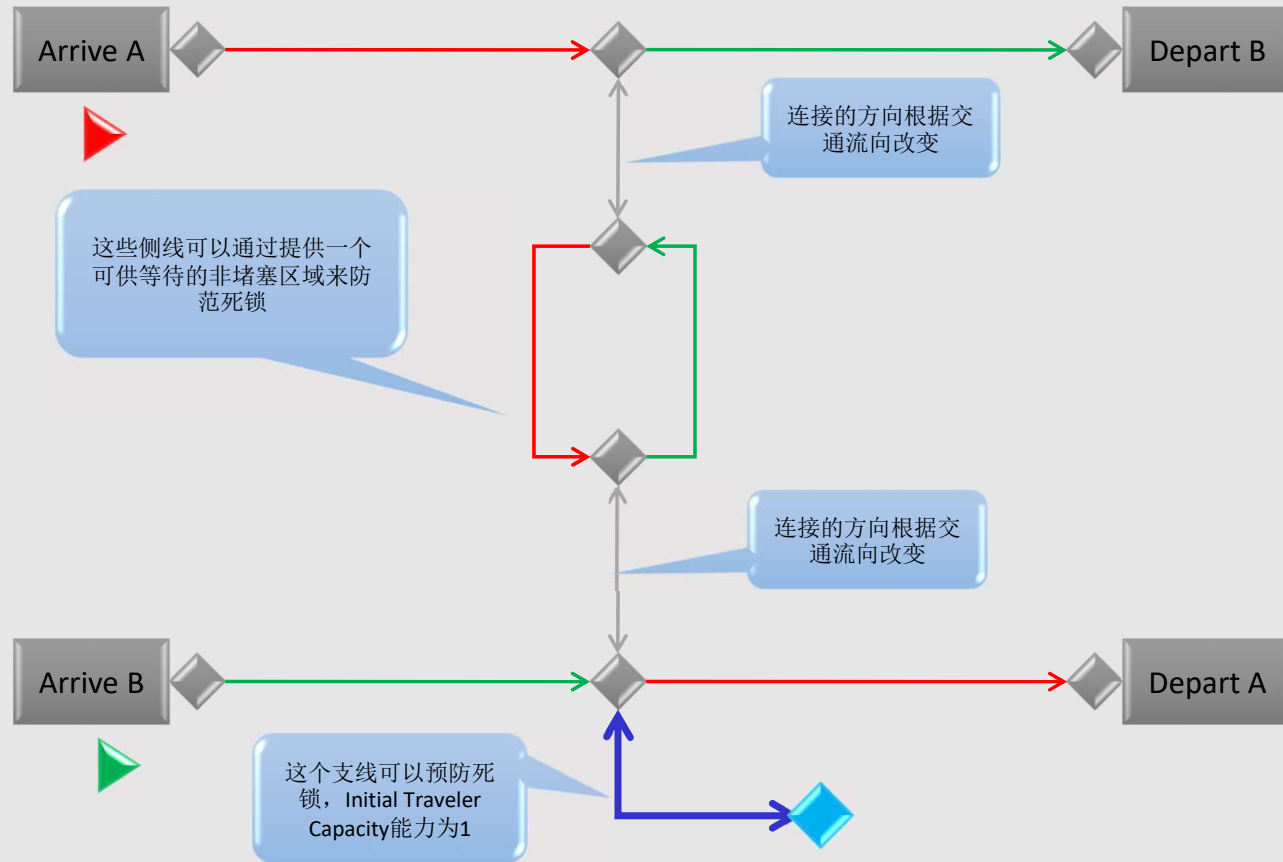
- 创建一个交错的路网模型，连线的方向如下 (将灰色路径的类型设成双向(*Bidirectional*))。
- 在Source的输出节点上设置 Entity Destination Type为*Specific*, Node Name为对应Sink的输入点 (ArriveA去DepartA)。
- 在运行模型前你可以预测会发生什么问题么?



例子：双向路径

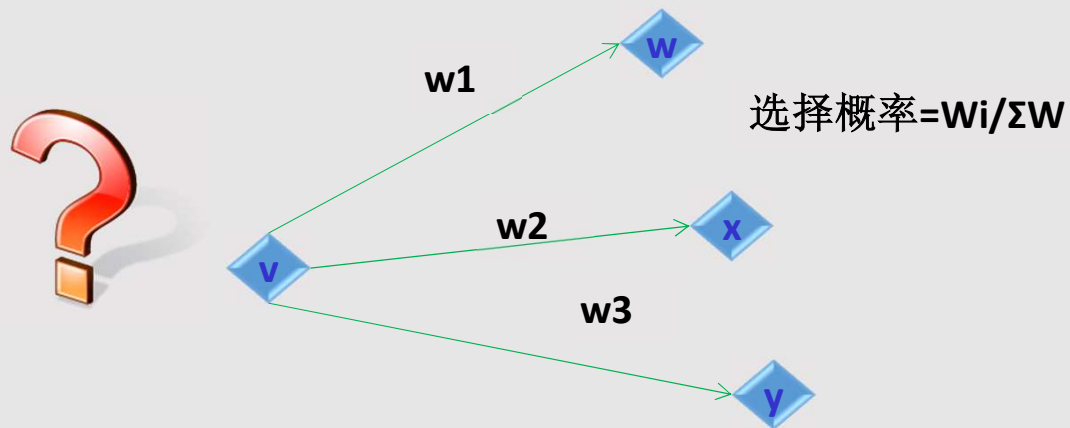


例子：双向路径

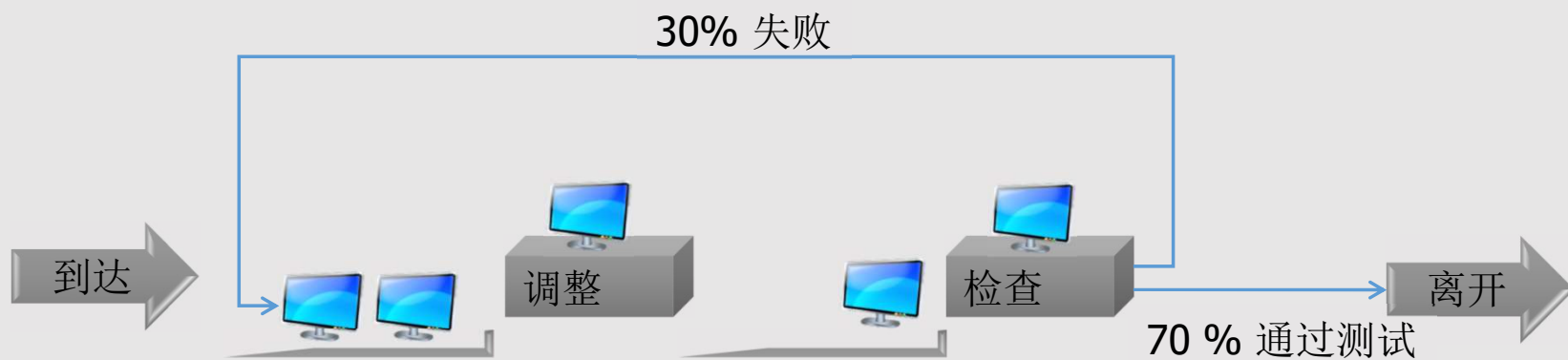


节点 / 线路选择

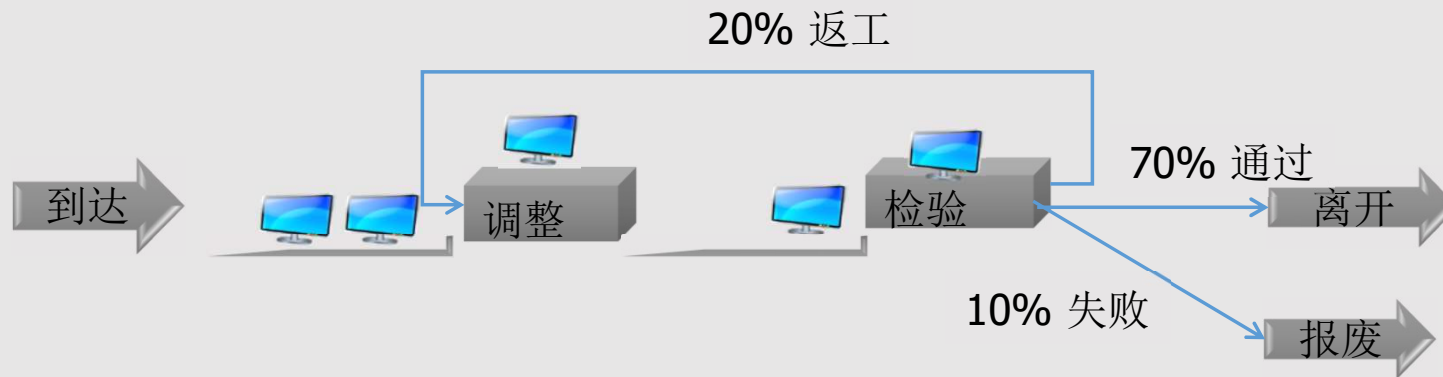
- ▶ 最短路径:选择到达目的地的最短路径
- ▶ 通过Link权重:通过给每个连接设定权重,按照这个权重随机地选择路径



例子：按照Link Weight选择路径



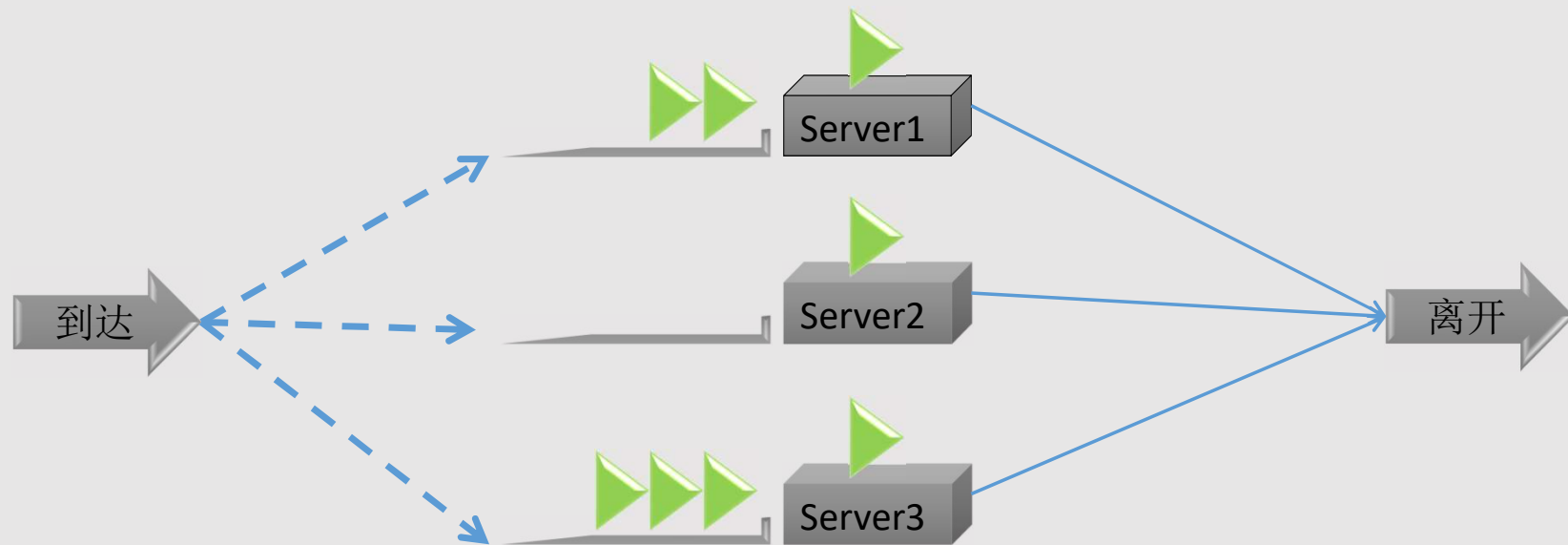
练习：Link Weight



- ▶ 重新构建之前的例子,添加一个报废的垃圾筒。
- ▶ **20%**的电视机需要返工, **10%**检验失败。
- ▶ 增加一个状态标签, 用于记录报废的数量。
- ▶ **挑战1:** 增加一个地板标签, 显示当前时间, 合格产品, 报废数量。
- ▶ **挑战2:** 部件在**8:00am** 才被报废。
 - 回忆一下逻辑操作符。

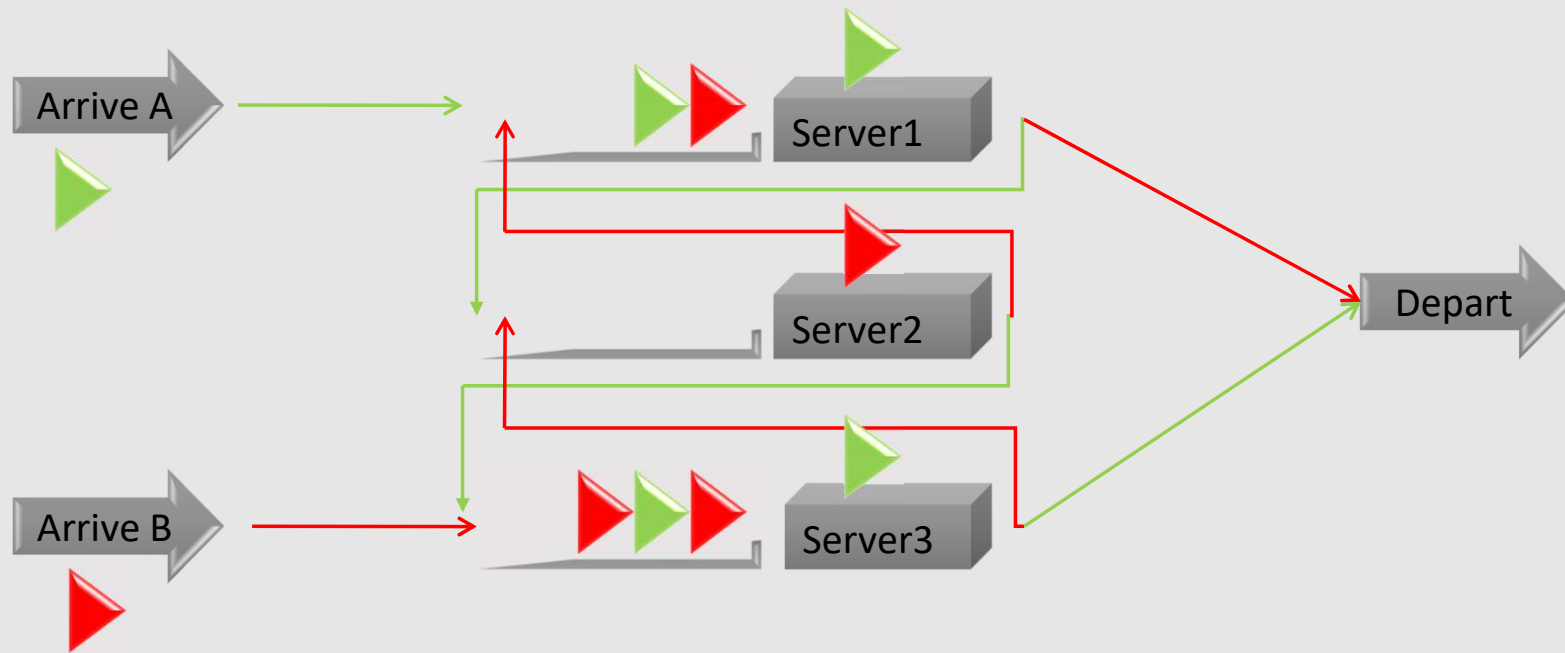
例子: 动态路径选择

▶ 按照最均衡化的方式投放



例子：根据序列列表选择路径

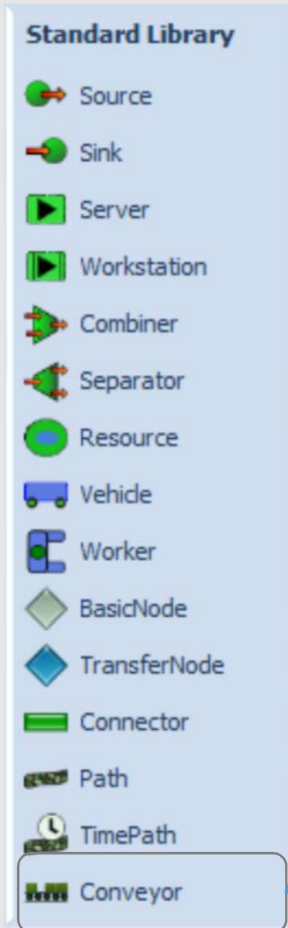
- ▶ PartA的加工线路： *Server1* → *Server2* → *Server3*
- ▶ PartB的加工线路： *Server3* → *Server2* → *Server1*



传送带类型



使用传送带运输物料



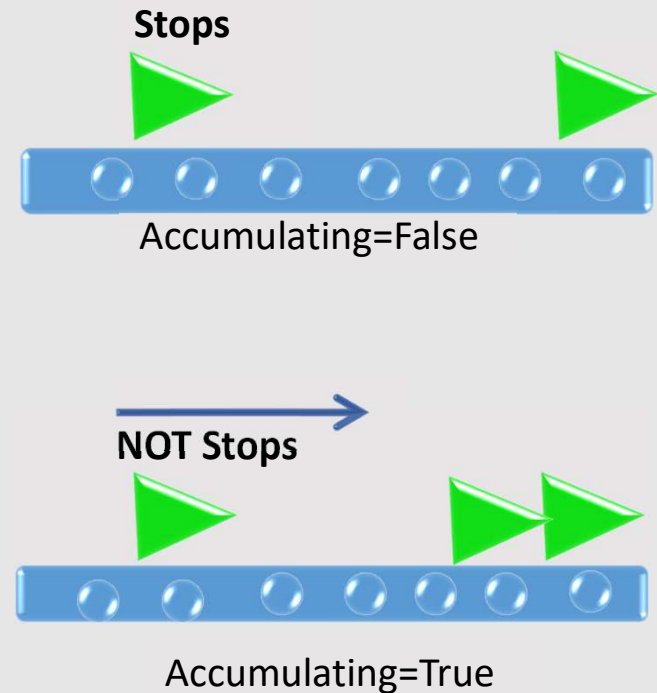
Conveyor 传送带- 可以在两个位置之间移动实体，使用累积式或者非累积式的传送装置。

- 累积式: 如果传送带末端有阻塞，允许实体可以在传送带末端进行堆积。
- 非累积式: 如果在传送带末端有阻塞，那么所有在传送带上的物体移动都被停止。

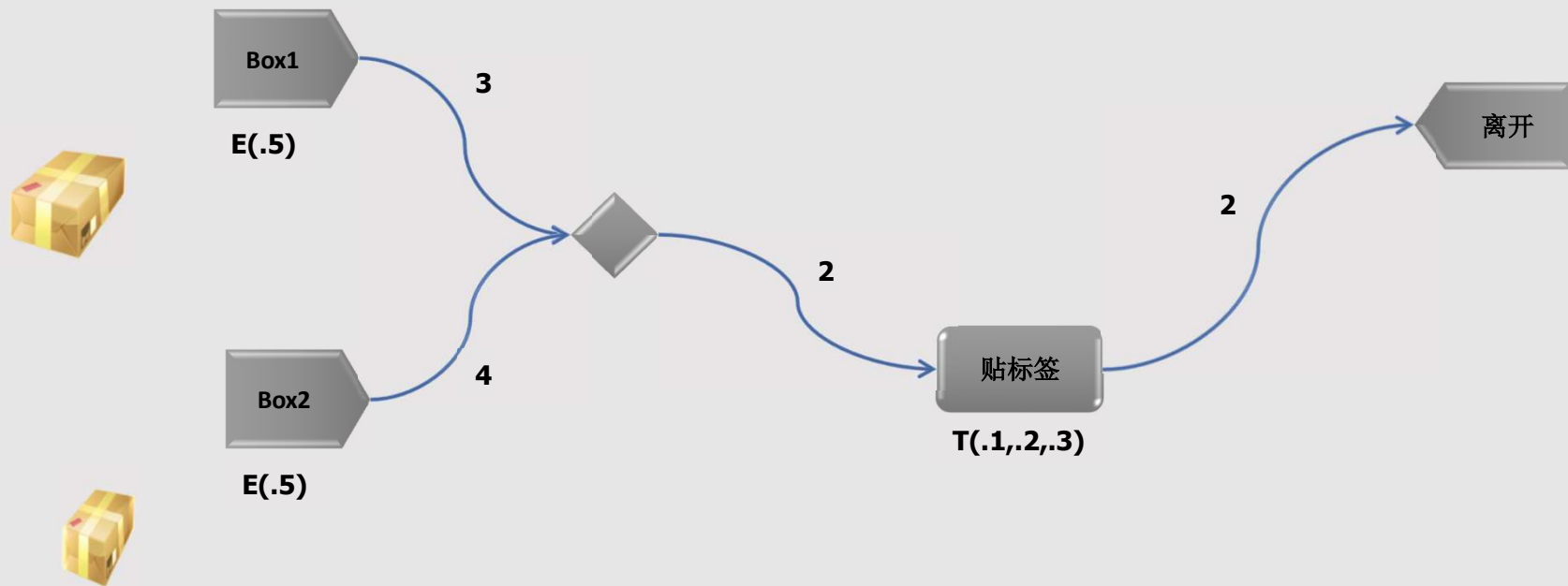
Tip 传送带可以有固定间隔的分区，比如，要模拟一个 Power & Free Conveyor（积放式输送链）在固定间隔上有吊具（挂钩）。

传送带对象

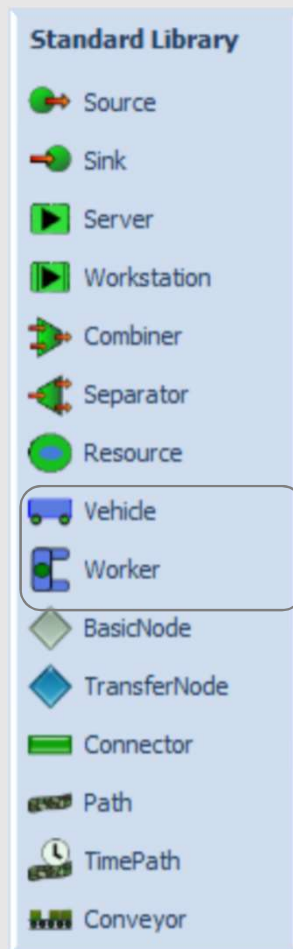
- ▶ 实体的移动由传送带控制。
- ▶ 当移动阻塞时，传送带停止。然后以**DesiredSpeed**重新启动。
- ▶ 累积式选项允许阻塞的实体滑动，未阻塞的实体继续移动。
- ▶ 实体的排列可以在任何位置，或者在等距的单元内。
- ▶ 也包含移动的容量（**Traveler Capacity**），“根据比例”（**Drawn to scale**），“选择权重”（**Selection Weight**）等选项。



传送带汇合



物料搬运系统-车辆/工人



车辆- 用于模拟装载，运输，卸载其他实体的搬运工具。

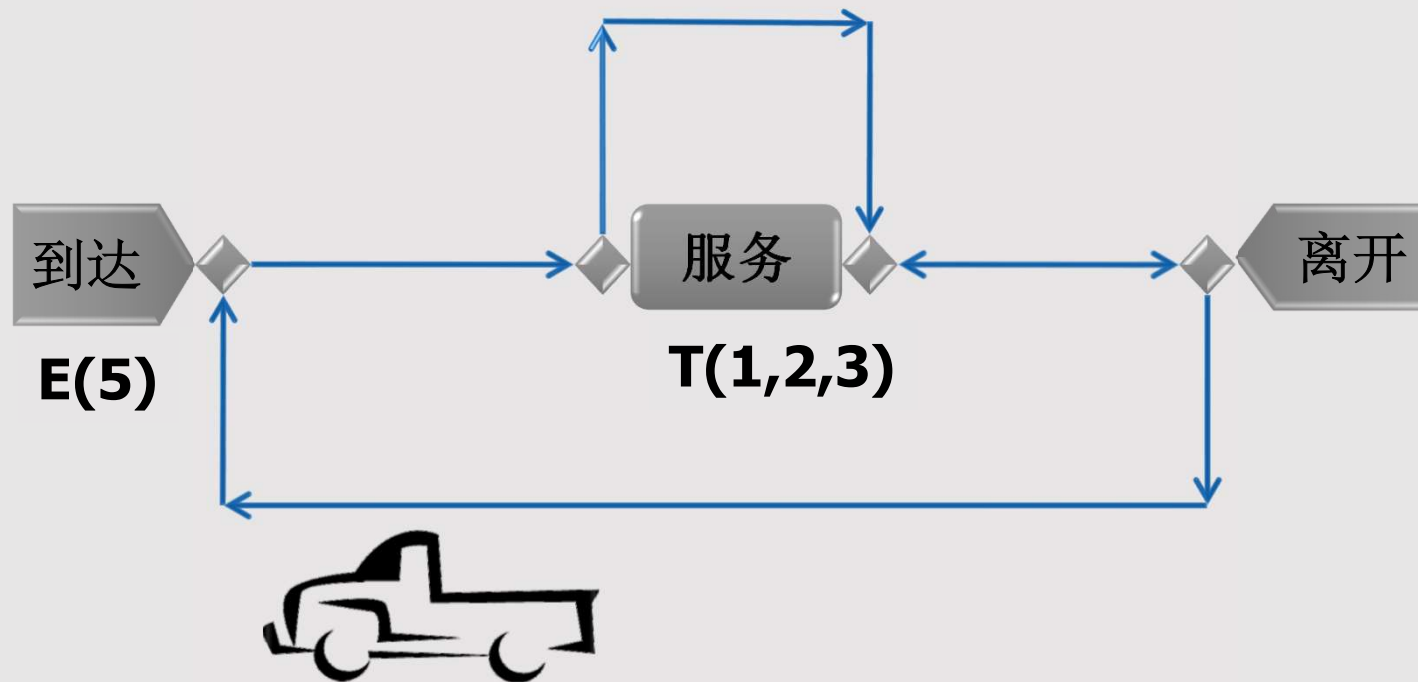
（具有稳定性逻辑，即车辆故障）

工人- 用于模拟一个劳动力资源，其执行的任务包括装载，搬运，卸下其他实体。



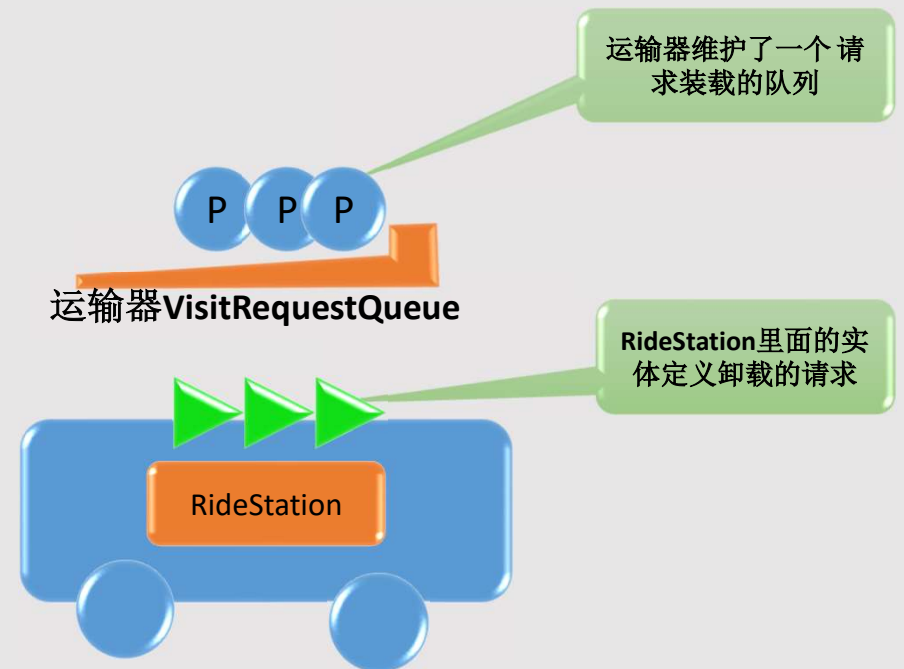
Tip Simio标准库对象属性面板上提供了附加过程的支持。附加过程（**Add-on Process**）是一个额外的，用户自定义的过程子程序，当和对象相关的某种特定事件发生时被执行。（例如：每当车辆或工人在评估是否要运输某个东西时，或正在装载/正在卸载时执行）。

按需装载（出租车模式）



车辆对象 (Vehicle)

- ▶ 运输器是实体一种, 它可以装载, 卸下和运输其他实体。
- ▶ 运输器具有初始节点属性和在系统中初始数量。
- ▶ 运输器具有期望速度, 装载能力, 装载/卸载时间。
- ▶ Routing Type指定运输器的路径选择策略: 可以是Fixed Route(固定线路) 或On Demand(按需发送)。
- ▶ 装载可以预先被预留, 或者临时被请求。
- ▶ Task Selection Strategy(任务选择策略)用于从VisitRequestQueue中选择下一个装载的实体或者RideStation中选择下一个要被卸载的实体。
- ▶ 运输器可以协商装载/预留的任务。实体或者运输器都可以拒绝装载或者被预留。
- ▶ Idle Action指定小车闲置时候的行为。



选择装载

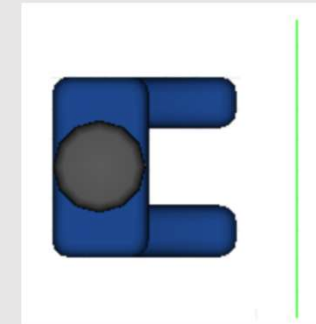
- ▶ 转移节点上的 *Ride on Transporter* 用于发起一个装载请求。
- ▶ 实体可以指定一个特定的运输器，或者根据 *Preferred Order*，或 *Smallest Value/Largest Value* 从本地可用的运输器列表中选择一个最佳运输器。
- ▶ 如果没有本地可用的运输器，那么实体可以预留某个车子进行装载，选项有 *Reserve Closest*，*Reserve Best*，或 *First Available at Location*；这将在运输器的 *VisitRequestQueue* 中放入一个装载请求。
- ▶ 实体在转移节点的 *RiderPickupQueue* 中等待被装载。
- ▶ 运输器和实体都必须接受预留请求或者装载请求。

等待装载的实体在转移节点的 *RidePickupQueue* 中等待



工人(Worker)

- ▶ 被获取/释放时，工人可以执行静态的任务-也可以先移动到实体的地点，然后执行任务。
- ▶ 工人也能够像车辆对象那样，拾取、携带或放下实体。
- ▶ 工人可以有固定的能力，或遵循工作计划。
- ▶ 工人具有*Desired Speed*（期望速度），*Ride Capacity*（装载容量）和*Load/Unload Time*(装载/卸载时间)等属性。
- ▶ 工人具有*Evaluating Transport Request*和*Evaluating Seize Request*逻辑，用于接受或拒绝不同的任务。



RideStation.Contents

车辆 vs 工人

Properties: Vehicle1 (Vehicle)	
<input type="checkbox"/> Show Commonly Used Properties Only	
[-] Transport Logic	
Initial Ride Capacity	1
Task Selection Strategy	First In Queue
[+] Load Time	0.0
[+] Unload Time	0.0
Park to Load/Unload	False
Minimum Dwell Time Type	No Requirement
[-] Travel Logic	
[+] Initial Desired Speed	2.0
Initial Network	Global
Network Turnaround Method	Exit & Re-enter
[-] Routing Logic	
Initial Priority	1.0
Initial Node (Home)	
Routing Type	On Demand
Idle Action	Park At Node
Off Shift Action	Park At Node
[-] Resource Logic	
Capacity Type	Fixed
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Park While Busy	False
[+] Reliability Logic	
[+] Financials	
[+] Add-On Process Triggers	
[+] Population	
[+] Advanced Options	
[+] General	
[+] Animation	

Properties: Worker1 (Worker)	
<input type="checkbox"/> Show Commonly Used Properties Only	
[-] Resource Logic	
Capacity Type	Fixed
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Park While Busy	False
[-] Travel Logic	
[+] Initial Desired Speed	2.0
Initial Network	Global
Network Turnaround Method	Exit & Re-enter
[-] Routing Logic	
Initial Priority	1.0
Initial Node (Home)	
Idle Action	Park At Node
Off Shift Action	Park At Node
[-] Transport Logic	
Initial Ride Capacity	1
Task Selection Strategy	First In Queue
[+] Load Time	0.0
[+] Unload Time	0.0
Park to Load/Unload	False
Minimum Dwell Time Type	No Requirement
[+] Financials	
[+] Add-On Process Triggers	
[+] Population	
[+] Advanced Options	
[+] General	
[+] Animation	

获取 vs 运输

获取和运输是两种不同的工作

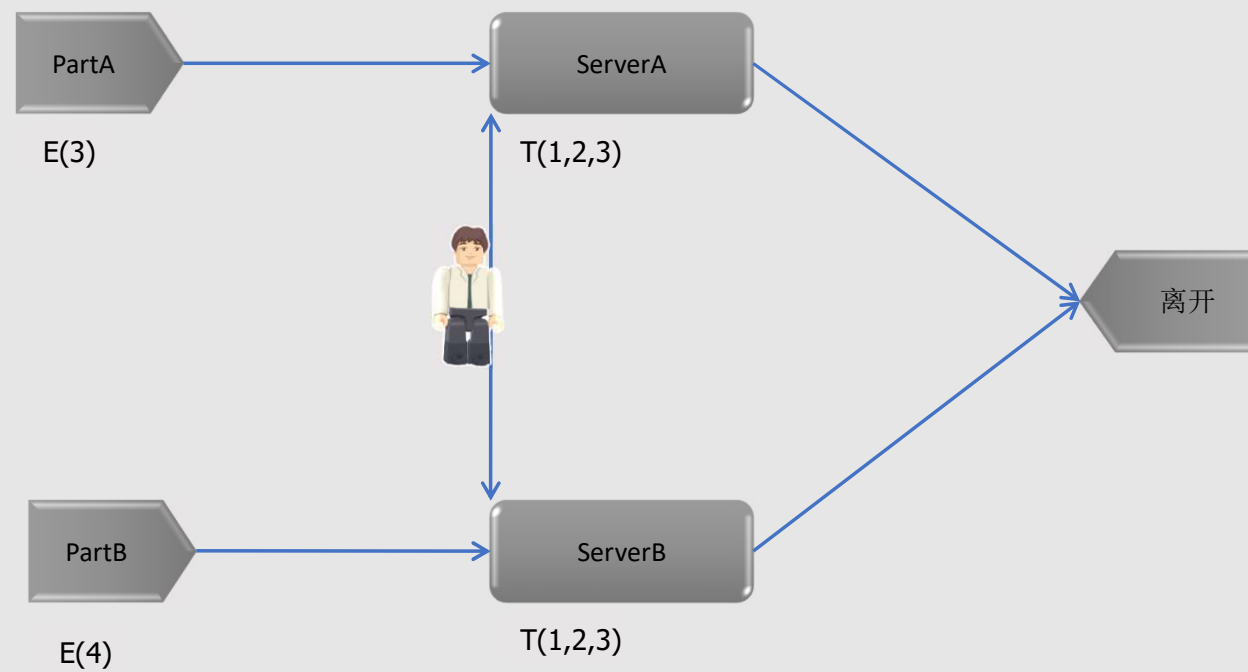
等待的地方不同

- 分配队列（Allocation Queue）
- 运输请求队列（VisitRequestQueue）

抓取优先于运输请求。

注意不要同时进行获取和搬运请求的工作以免造成冲突。

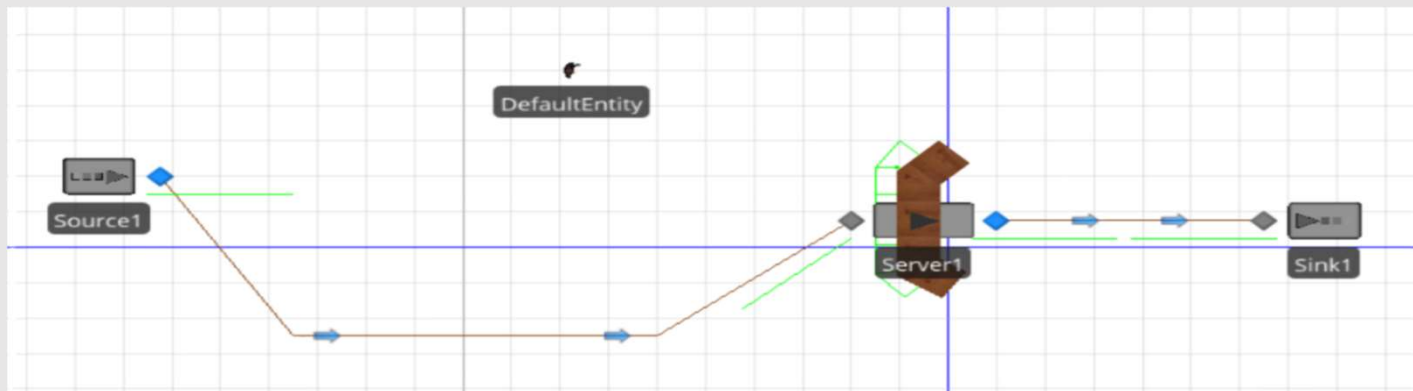
例子：工人



练习-Agony机场电车

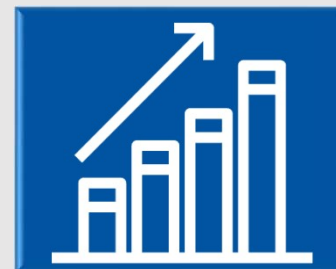
▶和之前相同:

- ▶ Agony机场的售票大厅在评估使用一个专用有轨电车，需要知道电车的容量（最大载客人数）。
- ▶ 一天来800个乘客，平均每1.2分钟(指数分布)的到达间隔，需要花费1-3分钟(均匀分布)走到售票处。
- ▶ 在售票处有5个工作人员，服务时间大概花1-10分钟(均匀分布)。
- ▶ 乘客步行200米至电车站，然后等待电车到达门口。电车在它550米的轨道上以每小时8公里的速度来回移动。
- ▶ 电车一次最多载客10个人，直接送到门口。10个人是最佳运载量吗？
- ▶ 假设乘客到达门口后，立刻离开。
- ▶ 挑战：增加有趣的三维图形，路径和动画。



Agony机场例子提示

- ▶ 为决策者提供适合的绩效指标。
- ▶ 用加工时间为0的Server代表电车站。
- ▶ 为运载能力创建一个新的引用属性。
- ▶ 使用仿真实验响应变量来比较所选的绩效指标。



第六章



模型数据



(0510) 82132584



无锡迅合信息科技有限公司©



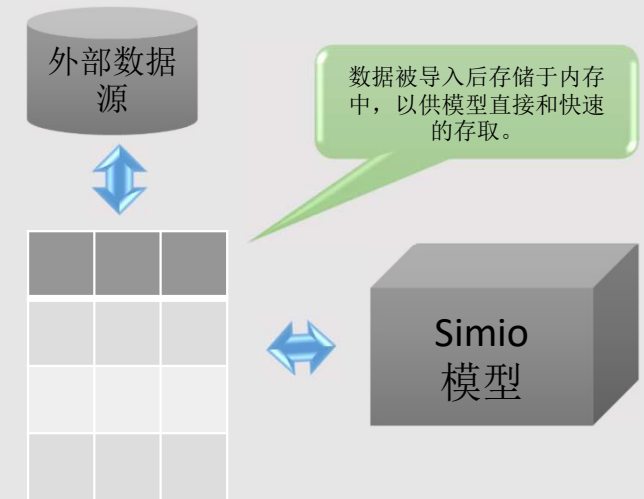
<http://www.simio-china.com/>

模型数据的主题

- ▶ 创建和编辑模型中的数据
- ▶ Simio数据表
 - ▶ Data Tables(数据表)
 - ▶ Sequence Tables(序列表)
 - ▶ Arrival Tables(到达表)
- ▶ Tabular Data（列表数据）
 - ▶ Rate tables(速率表)
 - ▶ Lookup tables(查询表)
 - ▶ WorkSchedules(工作计划)

模型数据

- ▶ 仿真模型通常具有用于描述系统中流动的不同对象和实体信息的大量数据。
- ▶ 当模型运行时，重复地存取外部数据是非常缓慢和低效的，Simio将外部存储于内存以便快速存取。
- ▶ Simio 可以使用简单的数据表存取数据，或者使用多重关系的完整数据集合存取数据。
- ▶ 数据表的数据格式和内容由用户自定义。



数据表

▶ **Tables**（数据表）具有列和行。

- 表格的列是标准属性或者是对象和元素的引用。

▶ 表格单元格的值可以使用以下语法对行和列进行引用：

- `TableName[RowNumber, ColumnNumber]`
- `TableName[RowNumber].ColumnName`

▶ 一个对象可以对表格行进行引用，这样的情况下，表格的值可以用下面的方法引用：

- `TableName.ColumnName`

▶ 其他和表格相关的函数：

- `TableName.AvailableRowCount`
- `TableName.PropertyName.RandomRow`

表格列是用强类型属性定义的。

表格行可以被对象或实体引用

数据表的功能块

The screenshot displays a software interface for managing data tables. At the top, a ribbon contains various tool groups:

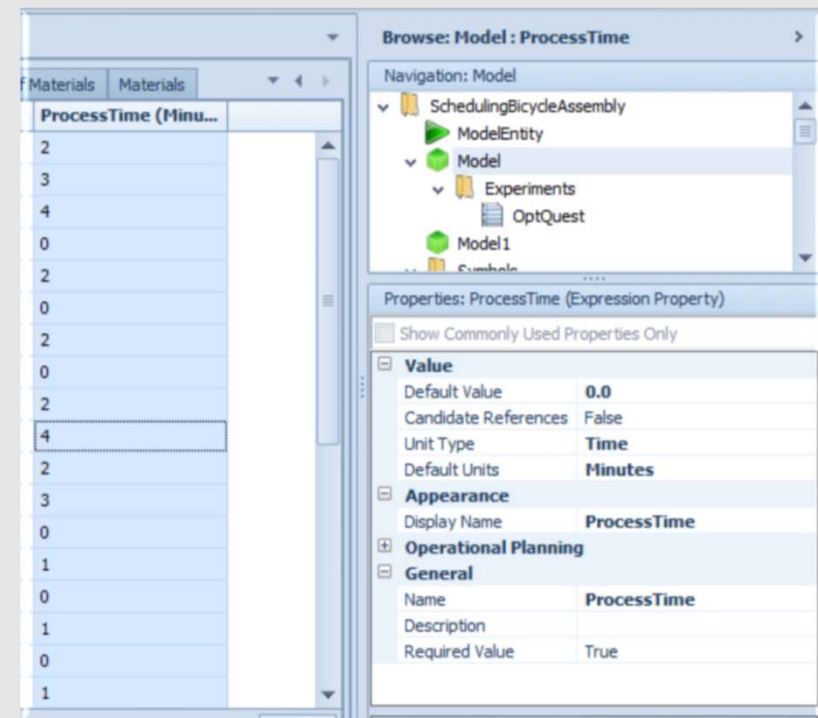
- Tables:** Add Data Table, Add Sequence Table, Add Output Table, Remove Table, Convert to Repeat Group.
- Add Column:** Standard Property, Element Reference, Object Reference, Foreign Key, Set Column As Key.
- Edit Column:** Move Left, Move Right, Change Type, Remove Column, Insert Row, Remove Row.
- Data:** Bind To, Remove Binding, Binding Options, Import, Export.

The main workspace is divided into several panes:

- Views:** A sidebar on the left with icons for Tables, Lookup Tables, Rate Tables, Work Schedules, Changeover Matrices, and Input Parameters.
- Materials:** A list of materials with columns for Order ID, Order Name, and Order Type.
- Definitions:** A central pane with a scrollable list of data types and categories:
 - Real, Integer, Boolean, Expression, DateTime, String, Sequence Number, List, Enumeration, Event, Rate Table, Table, Sequence Table, State, Schedule, Day Pattern, Selection Rule, Steering Behavior, Changeover Matrix, Task Dependency, Sequence Destination, Color.
 - Activity, Batch Logic, Changeover Logic, Container, Cost Center, Failure, Material, Monitor, Network, Operation, Output Statistic, Process, Regulator, Routing Group, State Statistic, Station, Storage, Tally Statistic, Task Sequence, Timer.
- Data Table:** A grid on the right showing data for a selected table. The columns are Priority, Quantity, Number Of Batches, and Ship Date. The data includes various entries with numerical values and dates.

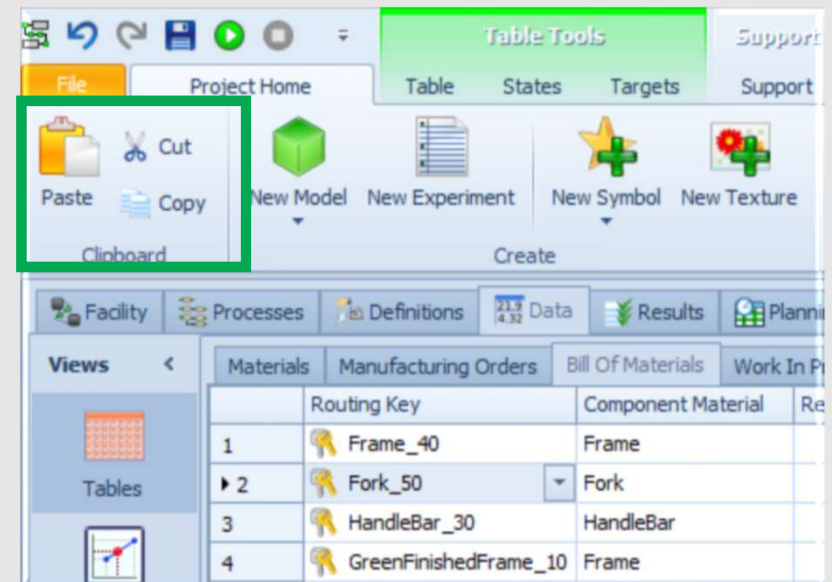
表格列的属性

- ▶ 选中表格的整列，观察它的属性。
- ▶ 改变列的名字。
- ▶ Required Value 属性允许单元格是空白的。
- ▶ 数值型和表达式类型的列具有 Unit Type（单位类型）和 Default Units（默认单位）属性。



复制表格

- ▶ 激活想要复制的表格
- ▶ 鼠标点击Project Home 菜单
- ▶ 使用Copy（复制）和Paste（粘贴）按钮 (不是 ctrl+c/ctrl+v)。



选择实体类型

▶ 你可以给表格增加一个数值型的列，指定每一行的权重（或实体类型的权重）。

▶ 根据权重的那一列，指定随机的行号，使用函数

TableName.ColumnName.RandomRow

这样就会根据权重来选择实体类型。

▶ SimBit模型: **SelectEntityTypeFromTable**

Product Table			
	Part Type	Product Mix	ProcessTime (Minutes)
▶ 1	PartA	50	.1
2	PartB	30	.3
3	PartC	20	Random.Uniform(.3,.5)

Properties: Source1 (Source)

Show Commonly Used Properties Only

Entity Arrival Logic

Entity Type: ProductTable.PartType

Arrival Mode: Interarrival Time

Time Offset: 0.0

Interarrival Time: Random.Exponential(.05)

Entities Per Arrival: 1

Stopping Conditions

Table Reference Assignments

Before Creating Entities

Table Name: ProductTable

Row Number: ProductTable.ProductMix.RandomRow

On Created Entity

练习-创建实体

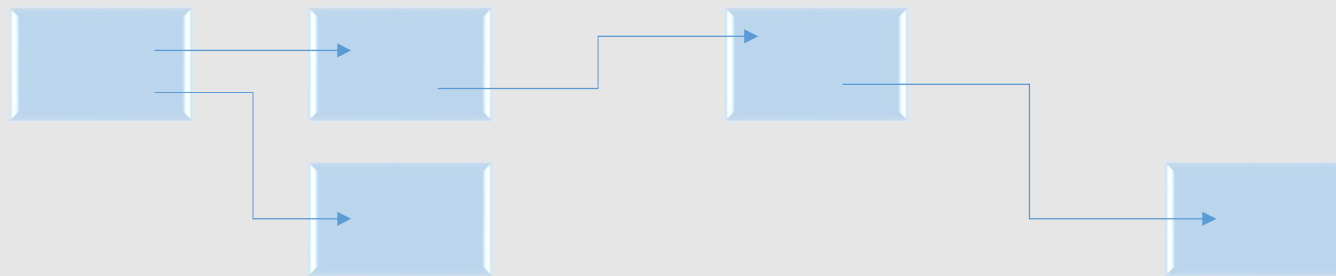
加载SimBit: *SelectEntityTypeFromTable*

- 1) 修改模型，使它永远创建第三行的实体类型（使用行号属性）。
- 2) 修改模型，使它永远创建第二行的实体类型（使用Entity Type属性），有问题么？
- 3) 修改模型，使用*Random.Discrete* 函数选择某个行，而不是使用*RandomRow* 函数 (采用同样的概率)。

注意: 概率是累积的。

关系型表

- ▶ **关系型表**是相互之间有指定关系的数据表。
- ▶ 关系是通过使用*Table Key*（主键）和*Foreign Key*（外键）构成的。
- ▶ 关系型数据表包含一个主从数据视图，使得表格之间的复杂关系可视化。
- ▶ 数据表的关系可以任意的嵌套。



关系型数据表的例子

Project Home Table

Tables: Add Data Table, Add Sequence Table, Remove Table

Properties: Standard Property, Object Reference, Foreign Key, Remove Property, Set Column As Key

Type Of Service	Service Time	Car Color
Brakes	Random.Triangular(1.5, 2, 2.5)	Green
Oil	1	Blue
Tires	Random.Triangular(.5, 1, 1.2)	Red

Arrivals

Arrival Time	Type Of Service
0	Brakes
0.25	Tires
0.5	Oil
0.75	Oil
1	Tires

ServiceTimes

Type Of Service	Service Time	Car Color
Brakes	Random.Triangular(1.5, 2, 2.5)	Green
Oil	1	Blue
Tires	Random.Triangular(.5, 1, 1.2)	Red

Arrivals

Arrival Time	Type Of Service
0.5	Oil
0.75	Oil

Properties: TypeOfService (Foreign Key Property)

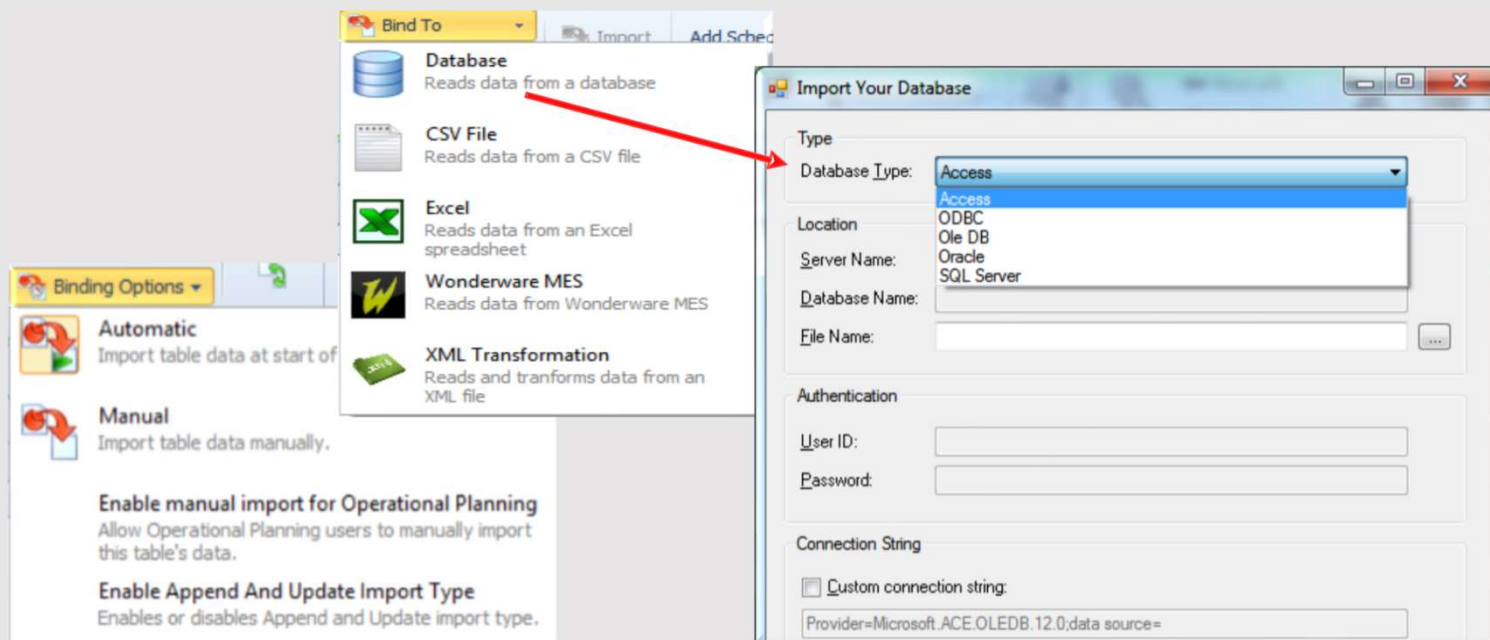
Show Commonly Used Properties Only

Value	
Default Value	
Table Key	ServiceTimes.TypeOfService
Appearance	
Display Name	TypeOfService
General	
Name	TypeOfService
Description	
Required Value	True

- ▶ 我们将在讨论序列表的时候展示更多。
- ▶ 参考SimBits模型和Example模型:
 - ▶ UsingRelationalTables
 - ▶ ServersUsingTaskSequenceWithDataTables_JobShop
 - ▶ SchedulingBicycleAssembly例子

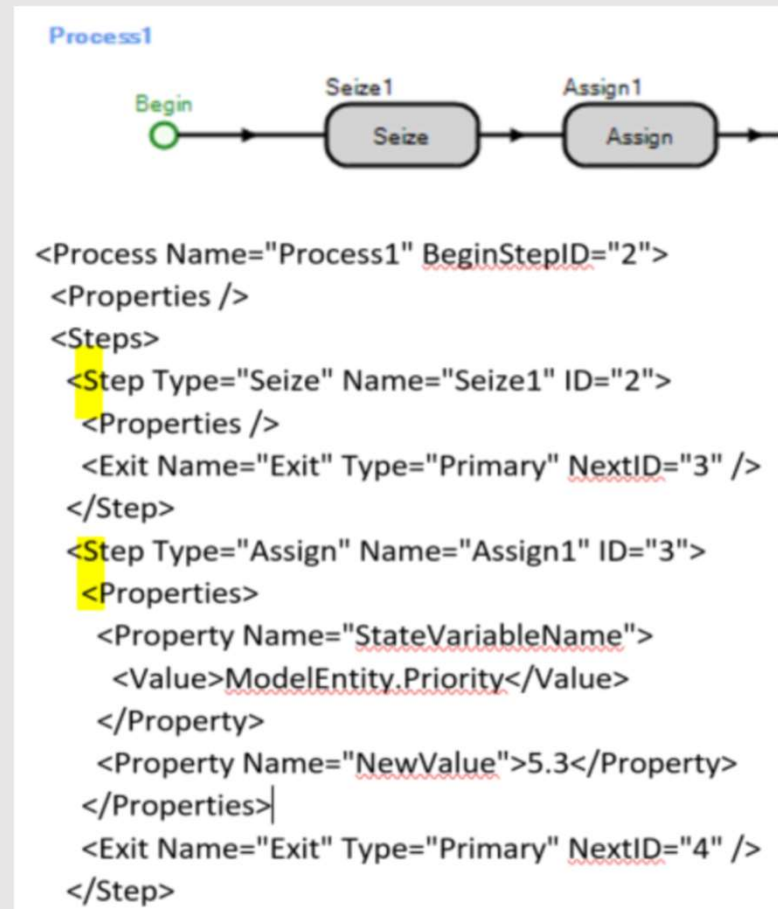
数据表输入/输出

▶ Simio提供灵活的数据导入和导出功能，支持许多常规的数据形式，数据库和灵活的数据映射。



可扩展标记语言(XML)

- ▶ 可扩展标记语言
- ▶ 机器可读和人工可读的数据交互标准
- ▶ 符合Microsoft标准:
 - DocX, XlsX
- ▶ Simio标准:
 - SpfX



练习-写入数据文件

加载SimBit: *SelectEntityTypeFromTable*

将数据导出成CSV文件

打开文件，加入一行

```
PartType: 'PartD'  
Product Mix: 50  
Process Time: 0.2
```

保持并关闭数据文件

绑定该数据文件，
导入数据至Simio

表格高级功能

▶ 按时间索引的表格将每一行和一个时间段联系起来。

Properties: Table 1 (Table)

Show Commonly Used Properties Only

Advanced Options

<input checked="" type="checkbox"/> Time-Indexed	True
Starting Time	6/20/2016
<input checked="" type="checkbox"/> Interval Size	1
Units	Hours
On Interval Process	

General

Name	Table1
Description	

Time-Indexed

If a table is time-indexed, then simulation time ranges will be mapped to the rows in the table using the specified Starting Time and Interval Size values.

During a simulation run, the `TableName.TimeIndexedRow` and `TableName.PropertyName.TimeIndexedValue` functions may be used to return information for the row in the table that corresponds to the current simulation time.

表格高级功能

▶ 从数据表中自动创建元素

Materials			
	Material Name	Initial Inventory	UnitCost (Default Curren...
1	Widget	150	12.42
▶ 2	6InchBolt	2500	4.58

Properties: MaterialName (Material Element Property)

Show Commonly Used Properties Only

Value

Default Value: null

Appearance

Display Name: MaterialName

Advanced Options

Reference Type: Create

Initial Property Values: 2 Rows

Auto-set Table Row P...: False

Operational Planning

Reference Type

If set to 'Create', then an element will be auto-created for the property instance, and changes to the property value will rename the element. If set the 'Reference', then changes to the property value will change the reference to the element with that name.

Initial Property Values - Repeating Property Editor

Items:

- Name: 'InitialQuantity', Value: 'Materials.InitialInven...
- Name: 'CostPerUnit', Value: 'Materials.UnitCost'

Properties:

Show Commonly Used Properties Only

General

Property Name: InitialQuantity

Value: Materials.InitialInventory

表格高级功能

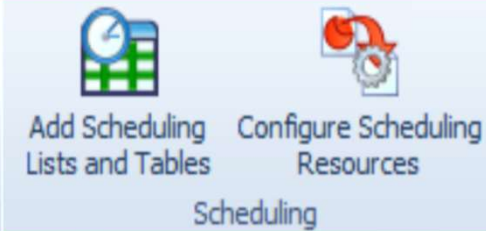
▶ Table Schema (表模式) 和重复组

The screenshot displays a software interface with a 'Table' tab selected. The 'Available Schemas' list includes 'Server.AssignmentsBeforeExiting' which is highlighted. A context menu is open over a table, showing 'AssignmentsTable' as an option. Another menu is open over the 'Before Exiting' state assignment, showing 'AssignmentsTable' and 'Create New Referenced Property...'. A table view titled 'Assignments Table' is shown below, containing the following data:

	State Variable Name	New Value ()
1	ModelEntity.DesiredSpeed	4
2	ModelEntity.Cost	ModelEntity.Cost+10
3	CurrentCapacity	2

表格高级功能

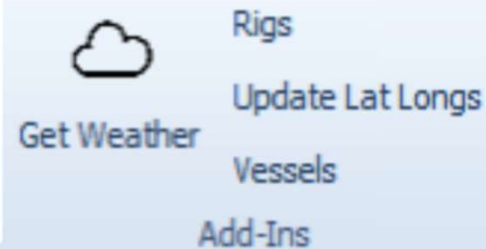
B2MML 数据结构（业务到制造标记语言）



Wonderware MES数据结构



定制的特定项目插件
如“天气”



状态变量和目标值（企业版）

▶ 表状态（Table States）

- 增加状态值类型的列
- 将输出数据写入该列 (Assign)
- 在运行中创建新的行(AddRow)

▶ 表格内目标（Table Targets）

- 识别目标值(表达式)
- 评估绩效

The screenshot displays the software interface with the 'Targets' ribbon selected. A red circle highlights the 'Targets' ribbon, which includes options like 'Integer', 'Boolean', 'DateTime', 'List', 'String', 'Element Reference', 'Object Reference', 'List Reference', 'Foreign Key State', 'Level', 'Level with Acceleration', 'Move Left', 'Move Right', and 'Remove State'. Below the ribbon, there are tabs for 'Facility', 'Processes', 'Definitions', 'Data', 'Results', and 'Planning'. The 'Views' section shows 'Table1'. A 'Performance Classification' panel is visible, featuring 'Add Target' and 'Remove Target' buttons, a 'Performance Classification' slider, and 'Risky', 'Caution', and 'Safe' labels. The 'Properties: TargetCost (Target)' panel is open, showing the following details:

Properties: TargetCost (Target)	
<input type="checkbox"/> Show Commonly Used Properties Only	
Value	
Expression	ManufacturingOrders.ProductionCost
Units	USD
Data Format	Real
Unit Type	Currency
<input type="checkbox"/> Lower Bound	
Units	USD
<input type="checkbox"/> Upper Bound	100000
Units	USD
<input type="checkbox"/> Value Classification	
Within Bounds	OnBudget
Below Lower Bound	
Above Upper Bound	Overrun
No Value	Incomplete
<input type="checkbox"/> Appearance	
<input type="checkbox"/> Operational Planning	
<input type="checkbox"/> General	
Name	TargetCost

输出表（企业版）

- ▶ 输出表（**Output Table**）的列都是状态变量类型。
 - ▶ 在模型运行过程中，使用**Add Row**步骤或在固定对象上使用**Table Row Referencing**功能，来增加行数。
 - ▶ 用作一个临时表格，或者是记录结果。
 - ▶ 输出表的外键的状态列允许访问上一级表格。
 - ▶ 使用**RemoveRow** 步骤从输出表中删除一行。
-
- **SimBit模型: UsingAddRowAndOutputTable_Enterprise**
 - **Example模型: SchedulingDiscretePartProduction**
 - **Help帮助文档: Enterprise Data Window**

输出表

Resources	Materials	Material Lots	Manufacturing Orders	Routings	Bill Of Materials	Work In Process	Manufacturing Orders Output
	Order Id	Routing Key	Scheduled Resource	Scheduled Start Time	Scheduled End Time		Scheduled Quantity
1	Order05	FinishedGoodC_10	Cut2	12/01/16 8:00:05 AM	12/01/16 4:18:05 PM		10.0000
2	Order04	FinishedGoodB_10	Cut1	12/01/16 8:00:06 AM	12/01/16 11:00:06 AM		10.0000
3	Order03	FinishedGoodB_20	Weld2	12/01/16 8:00:15 AM	12/01/16 10:18:15 AM		10.0000
4	Order01	FinishedGoodA_20	Weld1	12/01/16 8:00:17 AM	12/01/16 11:42:17 AM		10.0000
5	Order03	FinishedGoodB_30	Finish1	12/01/16 10:18:22 AM	12/01/16 4:18:22 PM		10.0000
6	Order10	FinishedGoodB_10	Cut1	12/01/16 11:00:08 AM	12/01/16 3:54:08 PM		10.0000
7	Order04	FinishedGoodB_20	Weld2	12/01/16 11:00:16 AM	12/01/16 3:24:16 PM		10.0000
8	Order01	FinishedGoodA_30	Shape1	12/01/16 11:42:30 AM	12/02/16 8:30:30 AM		10.0000
9	Order04	FinishedGoodB_30	Finish2	12/01/16 3:24:23 PM	12/02/16 11:24:23 AM		10.0000
10	Order11	FinishedGoodB_10	Cut1	12/01/16 3:54:11 PM	12/02/16 10:48:11 AM		10.0000
11	Order10	FinishedGoodB_20	Weld1	12/01/16 3:54:21 PM	12/02/16 10:18:21 AM		10.0000
12	Order06	FinishedGoodC_10	Cut2	12/01/16 4:18:07 PM	12/02/16 2:18:07 PM		10.0000
13	Order05	FinishedGoodC_20	Shape2	12/01/16 4:18:15 PM	12/02/16 1:06:15 PM		10.0000
14	Order01	FinishedGoodA_40	Finish1	12/02/16 8:30:41 AM	12/02/16 1:54:41 PM		10.0000
15	Order16	FinishedGoodB_10	Cut1	12/02/16 10:48:14 AM	12/02/16 3:42:14 PM		10.0000
16	Order11	FinishedGoodB_20	Weld1	12/02/16 10:48:24 AM	12/02/16 3:12:24 PM		10.0000
17	Order10	FinishedGoodB_30	Finish2	12/02/16 11:24:25 AM	12/05/16 8:24:25 AM		10.0000
18	Order05	FinishedGoodC_30	Finish1	12/02/16 1:54:44 PM	12/05/16 9:18:44 AM		10.0000
19	Order07	FinishedGoodC_10	Cut2	12/02/16 2:18:10 PM	12/05/16 11:18:10 AM		10.0000

Properties: Cut1 (Server)	
<input type="checkbox"/> Show Commonly Used Properties Only	
<input checked="" type="checkbox"/> Table Row Referencing	
<input checked="" type="checkbox"/> Before Processing	
Action Type	Add New Row
Table Name	ManufacturingOrdersOutput
<input checked="" type="checkbox"/> State Assignments	
On Entering	0 Rows
Before Processing	4 Rows
After Processing	2 Rows

List States (企业版)

- ▶ 企业版本也支持List中的成员作为状态列。
- ▶ 当结合Output Table和RemoveRows步骤使用时，支持动态运行时建立列表。

- Node Lists (节点列表)

Entity Destination Type	Select From List
Node List Name	NodeListMembers.NodeName
Selection Goal	Preferred Order
Selection Condition	NodeListMembers.Exclude = ...

- 标准库对象的附属资源，和Seize, Move, Release步骤

The screenshot shows a software interface with a table and a configuration panel. The table, titled 'Table1', has two columns: 'Which Worker' and 'Which Resource'. It contains three rows of data:

	Which Worker	Which Resource
1	Worker 1	Resource 1
2	Worker 2	Resource 2
3	Worker 3	Resource 3

The configuration panel on the right shows the following settings:

- Reliability Logic
- Table Row Referencing
- State Assignments
- Secondary Resources
 - Resource for Processing
 - Object Type: Select From List
 - Object List Name: Table1.WhichWorker
 - Selection Goal: Preferred Order
 - Request Move: To Node
 - Destination ...: Input@Server1
 - Keep Reserved If: (empty)

An arrow points from the 'Which Worker' column of the table to the 'Object List Name' field in the configuration panel.

序列表

- ▶ 实体和车辆经常需要按照固定线路移动。
 - 挂号 - 分类 - 就诊 - 治疗
 - 下城 - 中城 - 上城 (公交线路)
- ▶ 根据实体和车辆的不同, 采用的路径也许不同。
 - 病人的严重程度
 - 不同的车辆区, 本地配送/快递
- ▶ 序列表具有一列是目的地信息, 指定路径的目的地。
- ▶ 在转移节点的路由规则是“*By Sequence*”时使用。

Simbits: 序列表

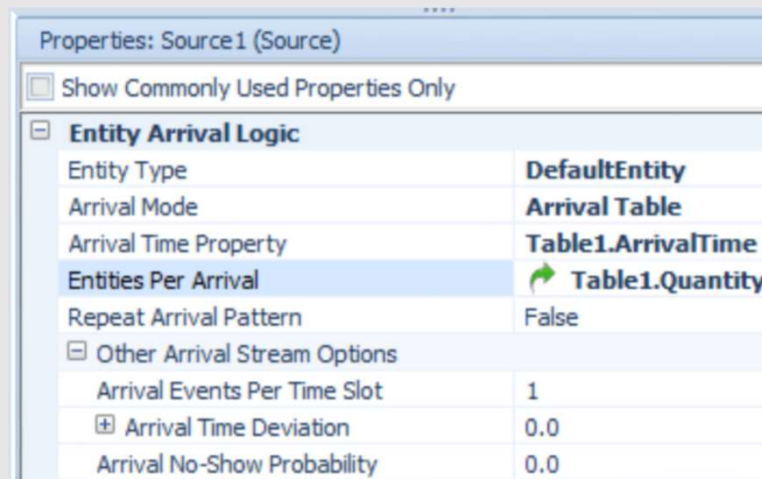
- ▶ EntityFollowsSequence
 - 单一实体类型
- ▶ EntityFollowsSequenceMultiple
 - 多个实体类型
- ▶ EntityFollowsSequenceWithTable
 - 实体的工作表引用它的序列数据
- ▶ EntityFollowsSequenceWithRelationalTables
 - 使用关系型数据表访问序列
- ▶ VehicleFixedRoute
 - 车辆在不同的站点装载和卸载

到达表 (Arrival Table)

- ▶ 包含到达信息的数据表。
- ▶ 用于在已知的到达时间产生实体。
- ▶ 能够产生单个实体或者成批实体。
- ▶ 确定性模式对于模型校验和调度应用是有帮助的。
- ▶ 随机模式对于计划的到达是有用的（医生预约）。

▶ SimBits模型:

- AppointmentArrivals
- UsingRelationalTables



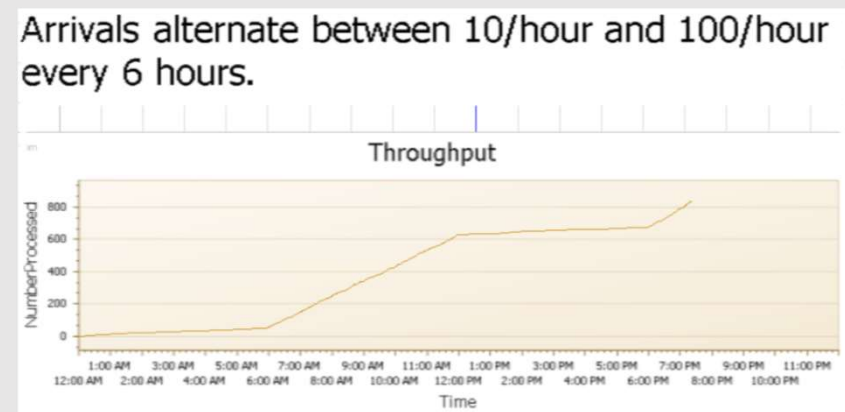
The screenshot shows the Properties window for 'Source1 (Source)'. The 'Entity Arrival Logic' section is expanded, showing the following properties:

Entity Arrival Logic	
Entity Type	DefaultEntity
Arrival Mode	Arrival Table
Arrival Time Property	Table1.ArrivalTime
Entities Per Arrival	Table1.Quantity
Repeat Arrival Pattern	False
Other Arrival Stream Options	
Arrival Events Per Time Slot	1
Arrival Time Deviation	0.0
Arrival No-Show Probability	0.0

速率表

- ▶ 速率表：在固定的时间间隔内用户自定义的到达速率（每小时）。
- ▶ 区间的数量和区间的长度由用户设定。
- ▶ 一般用于产生根据时间段变化的实体/事件到达 (不同的繁忙阶段)。
- ▶ 非静态指数分布
- ▶ *PlotRateTableArrivals*例子

Name		
▼ Rate Tables		
RateTable1		
Starting Offset	Ending Offset	Rate (events per hour)
Day 1, 00:00:00	Day 1, 06:00:00	10
Day 1, 06:00:00	Day 1, 12:00:00	100
Day 1, 12:00:00	Day 1, 18:00:00	10
Day 1, 18:00:00	Day 2, 00:00:00	100



查询表 (Lookup Table)

▶ 已知 x , 返回 $f(x)$ 的值。可以用于任何的 $f(x)$, 而不仅仅是基于时间的。

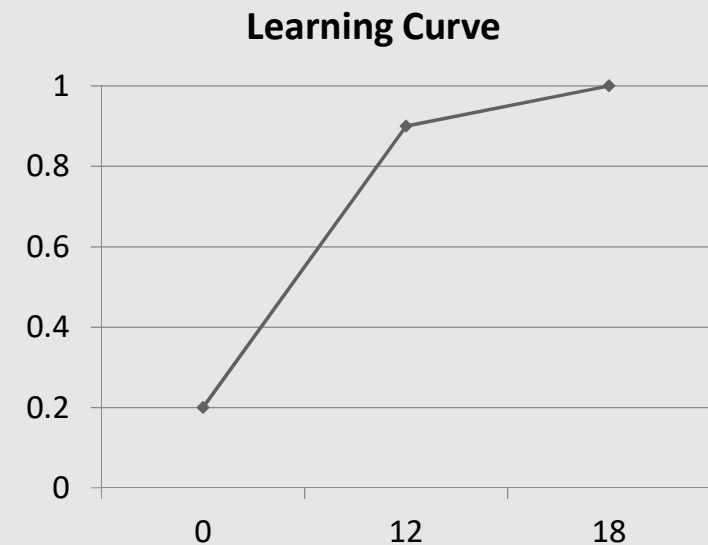
▶ 语法: LookupTableName[X表达式]

▶ 解释

▶ 查找

▶ 插值运算

▶ 端点的水平延伸

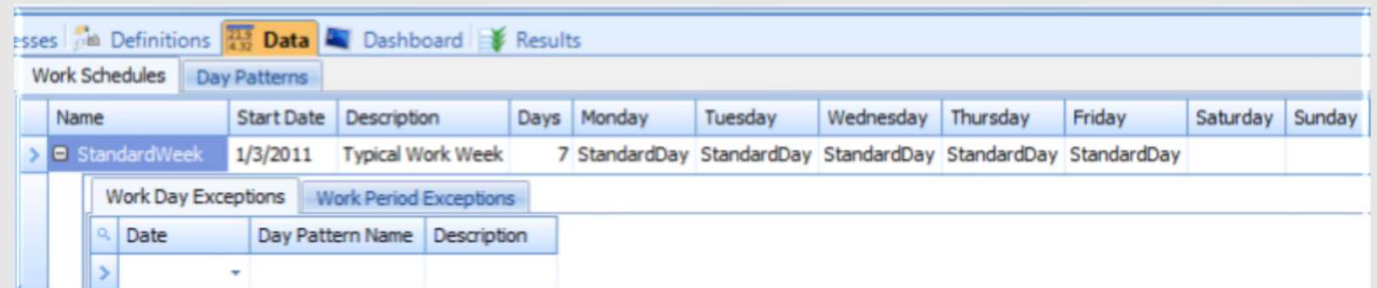


资源能力计划

- ▶ 任何对象都可以成为一个资源，随着时间的变化，其资源的能力可以变化。
- ▶ 许多对象都支持使用工作计划（ Work Schedule ），允许资源能力随着时间自动地变化。
- ▶ 数值型的能力也用来表示资源是否处于On-Shift状态（能力>0）,或Off-Shift 状态(能力= 0).
 - 对于某些对象而言 (Worker, Workstation)能力只能是 0 (Off-Shift) 或者是1 (On-Shift)。
 - 对于大多数对象来说，工作计划表示一个可变的能力 (比如, 8小时内是5个，后面2小时是4个，后面14个小时是0个)。

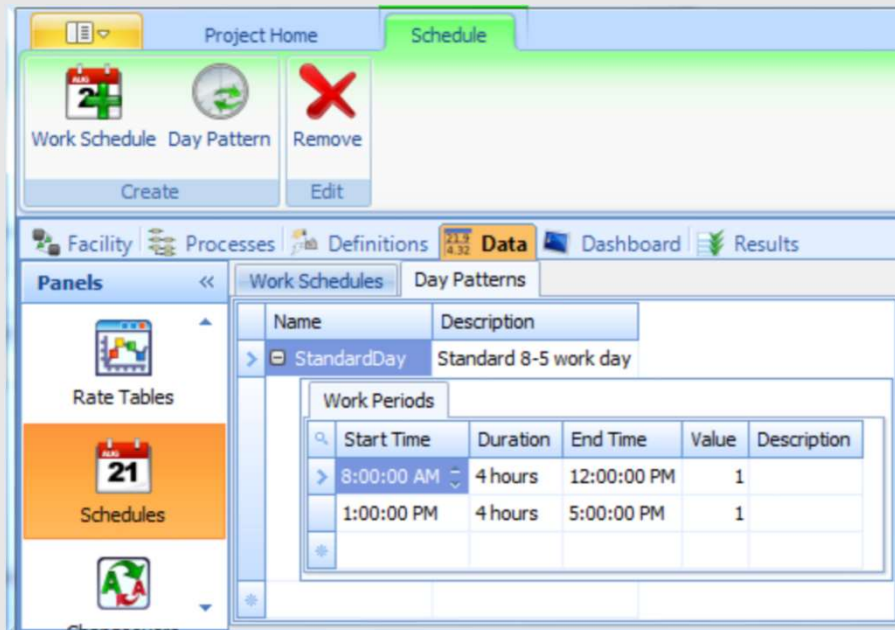
基于模式的工作计划

- ▶ **Work Schedule**（工作计划）是日模式的重复。



The screenshot shows the 'Work Schedules' table in Primavera P6. The table has columns for Name, Start Date, Description, Days, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. A row is highlighted for 'StandardWeek' with a start date of 1/3/2011 and a description of 'Typical Work Week'. Below the table, there are tabs for 'Work Day Exceptions' and 'Work Period Exceptions'.

Name	Start Date	Description	Days	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
StandardWeek	1/3/2011	Typical Work Week	7	StandardDay	StandardDay	StandardDay	StandardDay	StandardDay		



The screenshot shows the 'Day Patterns' table in Primavera P6. The table has columns for Name and Description. A row is highlighted for 'StandardDay' with a description of 'Standard 8-5 work day'. Below the table, there is a 'Work Periods' table with columns for Start Time, Duration, End Time, Value, and Description.

Name	Description
StandardDay	Standard 8-5 work day

Start Time	Duration	End Time	Value	Description
8:00:00 AM	4 hours	12:00:00 PM	1	
1:00:00 PM	4 hours	5:00:00 PM	1	

- ▶ **Day Patterns** 定义班次或者工作时间段，以及资源能力（数值）
- ▶ **Exceptions** 定义例外情况，如加班，节日，保养 ...

数据表的工作计划

- ▶ 数据表包含重复的工作阶段。
- ▶ 可以使用任何数据模式。

The screenshot displays the Primavera P6 software interface. The top ribbon is set to 'Schedule'. In the 'Schedule' group, the 'Table Based Work Schedule' icon is highlighted with a red circle. Below the ribbon, the 'Views' pane shows 'Table Based' selected. The main workspace displays a table of work schedules. A secondary window titled 'Table 1' is open, showing a detailed view of the schedule data. A white arrow points from the 'Table 1' window back to the main workspace.

Name	Summary
WorkSchedule1	On 'Table1' from 'Table1.StartingTime' to 'Table1.EndingTime' with value 'Table1.Capacity'

Starting Time	Ending Time	Capacity
1 7/20/2016 8:00:00 AM	7/20/2016 12:00:00 PM	1
2 7/20/2016 12:30:00 PM	7/20/2016 4:00:00 PM	1

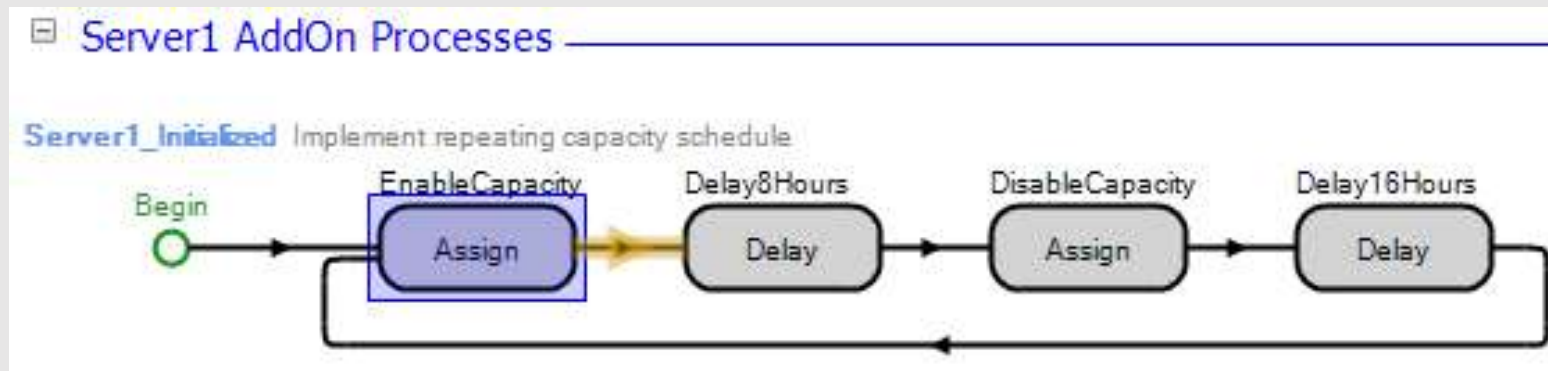
Properties: WorkSchedule1 (Table Work Schedule)	
Show Commonly Used Properties Only	
General	
Name	WorkSchedule1
Table Name	Table1
Start Date Time	Table1.StartingTime
End Date Time	Table1.EndingTime
Value	Table1.Capacity
Cost Multiplier	1

高级的计划概念

- ▶ 每个工作计划都指定一个名为Value的值，它可以表示任意内容，而不仅仅是能力的数值。
- ▶ 用于计划表中的函数
 - ▶ **Value(dateTime)** – 返回指定日期时间的Value值。
 - ▶ **NextValue(dateTime)** – 在指定日期时间上返回下一个时间上的Value值。
 - ▶ **TimeOfNextValue(dateTime)** – 如果在值定的时间日期上，返回下一个Value值变化的那个时间。
 - ▶ **TimeUntilNextValue(dateTime)** – 如果在指定的时间日期上，返回到下一个Value值变化前的那段持续时间。**AverageValue(fromDateTime, toDateTime)** – 返回在指定的时间段内，Value值的平均值。
 - ▶ **MinimumValue(fromDateTime, toDateTime)** – 返回在指定的时间段内最小的Value值。
 - ▶ **MaximumValue(fromDateTime, toDateTime)** – 返回在指定的时间段内最大的值。

“手工创建”计划

- ▶ (可能是)一个定期改变能力的简单过程。
- ▶ 提供了逻辑上的简易性，或者所需的控制和灵活性（“智能”的转换期）
- ▶ 参见SimBit **ResourceStatesWhenOffShift**，了解如何控制转换期内的细节。





第七章



使用过程扩展逻辑



(0510) 82132584



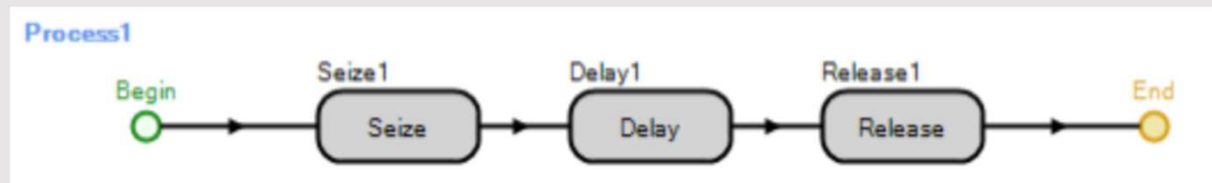
无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

过程(Processes)

- ▶ 过程是随着时间发生的一系列活动，它们能够横跨时间，并且改变模型的状态。
- ▶ 过程是由包含过程步骤的流程图组成，令牌执行步骤(行为)，步骤改变一个或多个“元素”的状态。



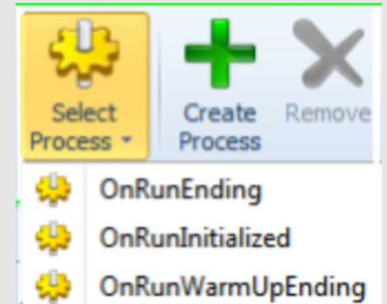
- ▶ 过程步骤执行的活动，例如：
 - 延迟指定的时间
 - 抓取或释放一个对象
 - 等待事件的发生
 - 根据概率和条件进行判断
 - 将实体转入站点
 - 搜索一个内容或者对象 ...

过程的应用

- ▶ 过程用于建立模型逻辑。
- ▶ 模型可以完全用过程或者对象来建立，或者用两者的组合。
- ▶ 对象提供了 *快速建模* 的能力；过程则提供了 *灵活性*。
- ▶ 标准库对象都是用过程建立的。
- ▶ 过程能够用于对标准库对象插入特殊的逻辑，这是在实例层面增加的逻辑。

建立Process的方法

- ▶ 标准过程是Simio软件定义的过程，被Simio仿真引擎自动执行。例如，*OnRunInitialized*过程在模型的每个对象初始化时候被Simio自动执行。
 - ▶ 决策过程是仿真引擎使用的标准过程，是0耗时的判断过程,要求对象返回True/False决策。(例如：你会装载我吗?)。决策过程内不能有延迟类型的过程步骤。
- ▶ 附加过程（Add-on Process）是用户定义的过程，被加入到模型对象之中，允许用户在那个对象中插入特殊逻辑。
- ▶ 事件触发的过程是用户自定义的过程，这个过程被模型中某个发起的事件所触发。(例如,进入Server的输出节点时会触发*Output@Server1.Entered*)。



附加过程

- ▶ 给对象增加新功能，而无须改变对象逻辑。
- ▶ 根据每个附加过程的描述，决定最佳的触发点。
- ▶ 过程在上级模型对象（Parent Object）
 环境内被执行，即在父对象内部“存活”
 （具有相应的数据可视性）。

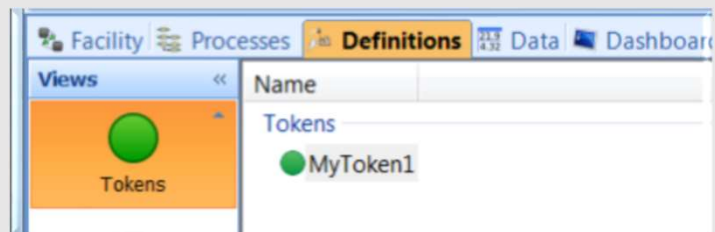
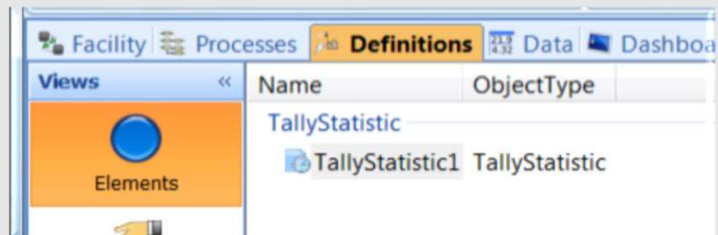
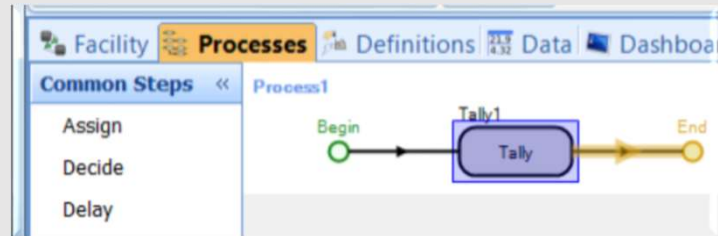
Properties: Server1 (Server)

Show Commonly Used Properties Only

- ⊕ Process Logic
- ⊕ Buffer Capacities
- ⊕ Reliability Logic
- ⊕ State Assignments
- ⊕ Secondary Resources
- ⊕ Financials
- ⊖ Add-On Process Triggers
 - Run Initialized
 - Run Ending
 - Entered
 - Before Processing
 - Processing
 - After Processing
 - Exited
 - Failed
 - Repaired
 - Evaluating Seize Request**
 - On Shift
 - Off Shift
- ⊕ Advanced Options
- ⊕ General
- ⊕ Animation

Evaluating Seize Request
Occurs when this Server object is evaluating whether to accept or reject a request to seize capacity of the server. In the executed decision process, assigning a value less than or equal to 0.0 to the executing token's ReturnValue state (Token.ReturnValue) indicates that the seize request is rejected.

过程总结



▶ 步骤 (被令牌执行)

▶ 属性

▶ 元素 (被步骤所引用)

▶ 属性

▶ 状态

▶ 事件

▶ 令牌

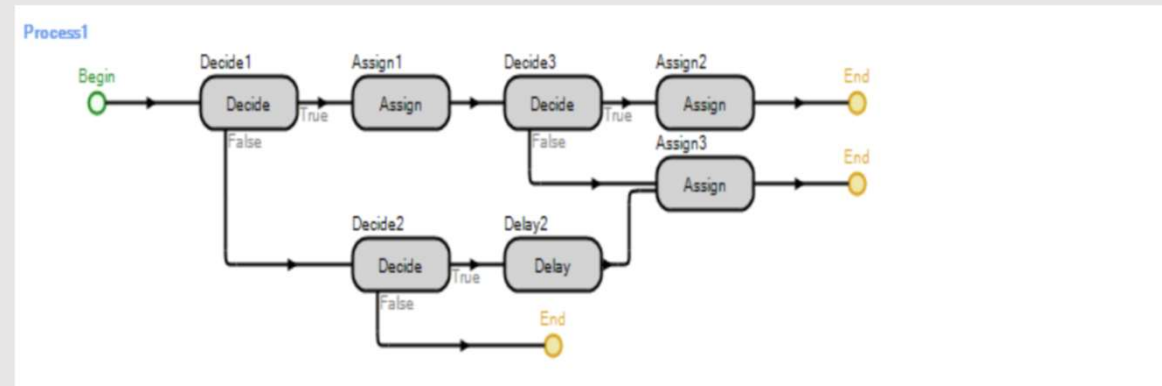
▶ 属性

▶ 状态



过程的编辑

- ▶ 过程有一个起点 (*Begin*) 和一个或多个终点 (*End*)。
- ▶ 拖放步骤到图中，软件将自动排列这些步骤。
- ▶ 尝试拖动一个终点到某个步骤的起点。
- ▶ 尝试将步骤移动到新的位置。
- ▶ 点击步骤的外部区域，可选中整个过程。
- ▶ 使用+/- 键对选中的过程缩放。
- ▶ 在Simio 中拷贝/粘贴一个过程，或者拷贝至外部应用程序（如微软Office中特殊粘贴）。
- ▶ 给过程设定*Category*名，以便可以对过程进行归类，展开/隐藏。
- ▶ 双击Add-on Process Triggers 创建相应附加过程。
- ▶ 为了可读性，用户可以给过程中的某些过程步骤标记颜色（*General* 属性项-*Color*属性）。



Properties: Process1 (Process Element)	
<input type="checkbox"/> Show Commonly Used Properties Only	
Basic Logic	
Triggering Ev...	
Triggering Ev...	
Advanced Options	
General	
Category	Inventory
Name	Process1
Description	
Public	False
Report Statistics	True

练习-基本过程

- ▶ 在模型层面添加一个Integer类型的状态变量，取名为 Counter（计数器）。
- ▶ 创建一个初始化流程，当仿真开始时被执行(OnRunInitialized)。
- ▶ 增加过程逻辑：等待10分钟 (Delay),给状态变量递增1 (Assign), 然后通过一个判断分支 (Decide)再次回到Delay步骤（条件是计数器到达50）。
- ▶ 在Facility视图增加状态动画，显示当前仿真时间和刚才那个状态变量的值。
- ▶ 高级：当上面的流程完成时，再执行（Execute）其他过程，其他过程的逻辑是每一分钟对状态变量增加1，直到计数器到达250。

令牌 (Token)

令牌是实体的代理。

允许一个实体一次做多件事情。

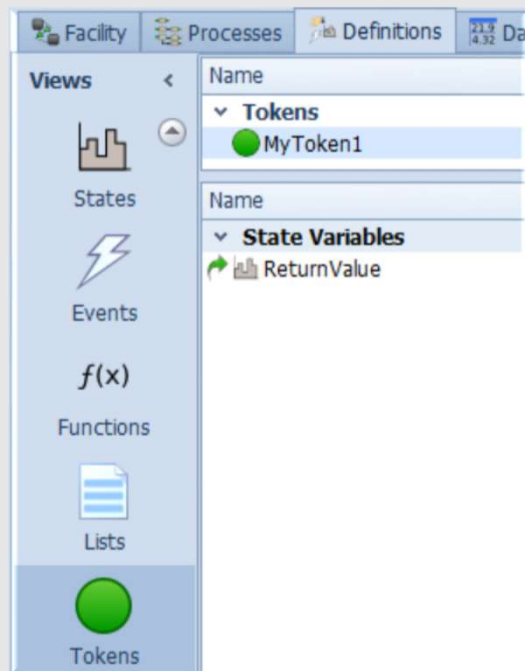
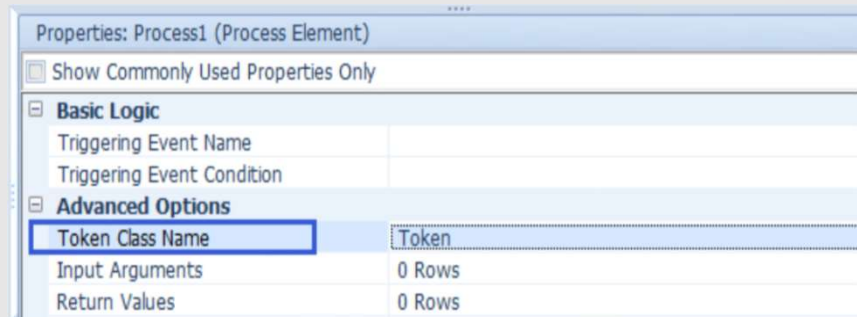
美食广场聚餐

一个家庭需要不同的食品，但是他们想在一起用餐...



- ▶ 老爸买单
 - (实体)
- ▶ 代理
 - (令牌)

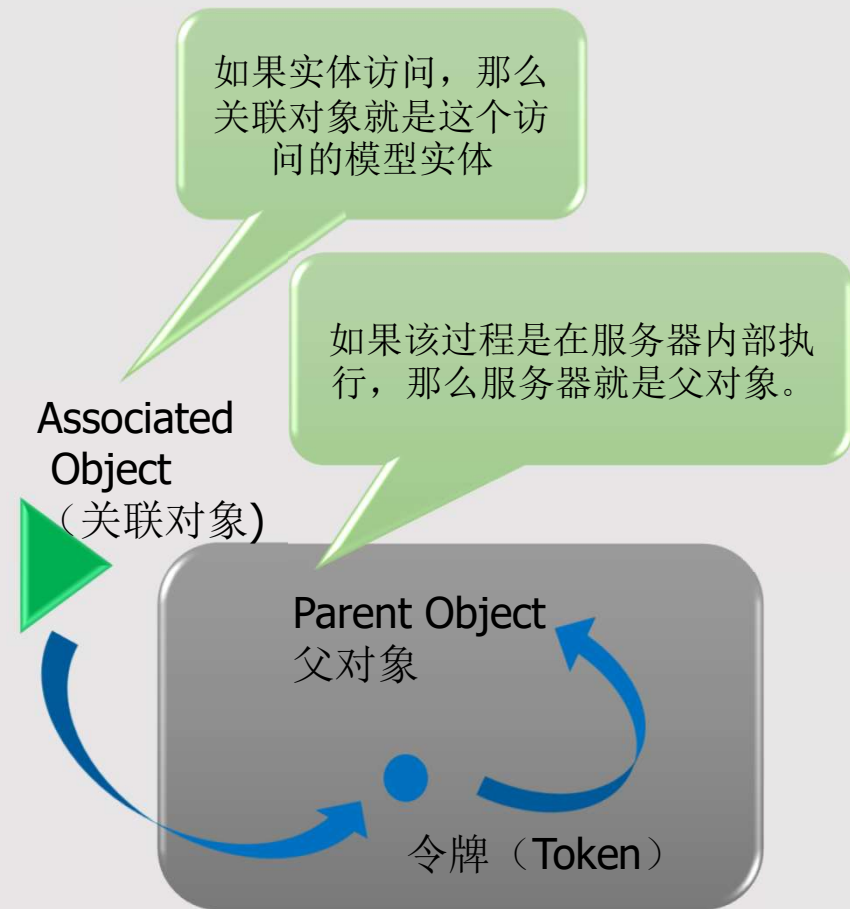
令牌 (Token)



- ▶ 令牌是实体的代理。
- ▶ 允许一个实体一次做多件事情。
- ▶ 每个过程指定它所需要的令牌类型。
 - ▶ Process > Advanced > Token Class Name
- ▶ 令牌可以有定制的属性 and 状态值。
 - ▶ Definitions > Tokens

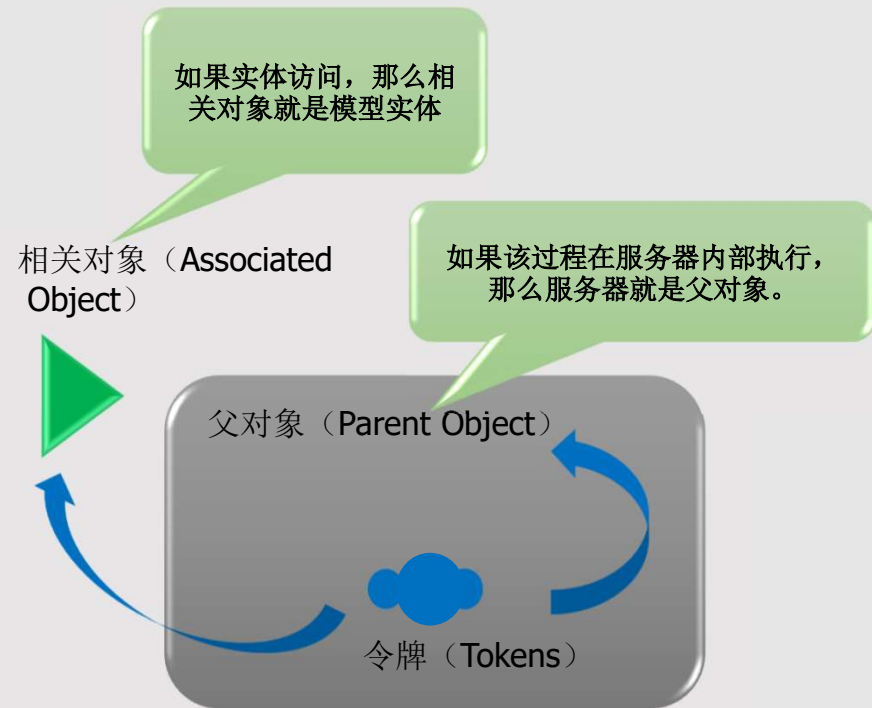
创建令牌

- ▶ 一个“关联”对象引起一个令牌的创建，令牌执行另外一个对象（父对象）中的过程。
- ▶ 典型例子：
实体执行了服务器内部的一个过程。

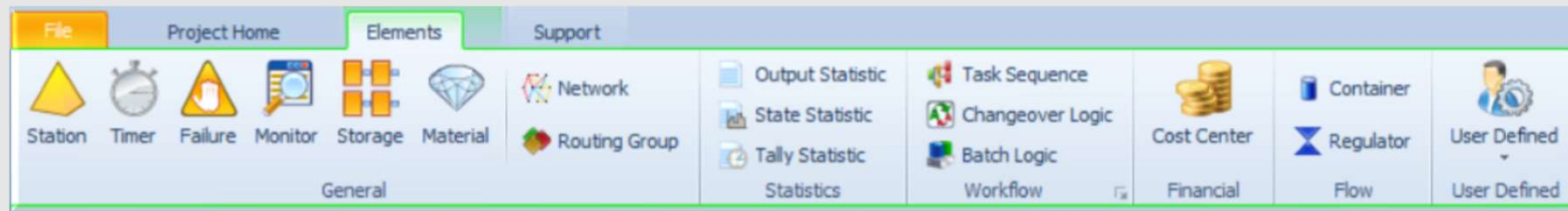


令牌的数据引用

- ▶ 令牌装载了指向父对象和关联对象的索引。
- ▶ 相关对象的属性（状态）可以直接用类名来引用，例如：
ModelEntity.TimeCreated
父对象的属性（状态）直接用名字引用，例如：
ProcessTime
- ▶ 过程在父对象内部被定义和执行。
- ▶ 关联对象就是导致过程被执行的外部对象或在“Search”过程中匹配的对象。
- ▶ 令牌还有一个*ReturnValue*状态值，可以在运行过程中被赋值，该返回值通常用在特殊的决策过程中。



元素



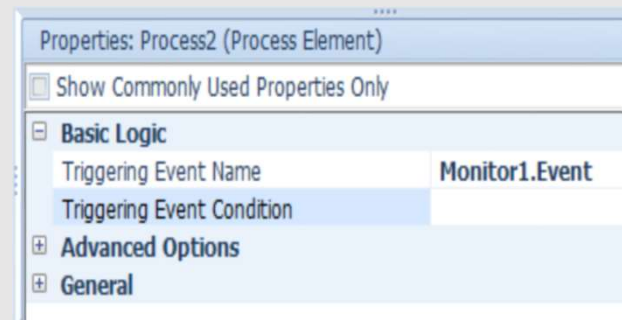
- ▶ 元素是由Simio定义的属性，状态，事件和行为的集合。
- ▶ 元素具有Simio定义的行为，由过程步骤所控制。
- ▶ 元素有指定输入选项的属性值。
- ▶ 元素可以在选定的时间点上触发事件。
- ▶ 元素是在模型的Definitions窗口被添加的。

元素的简介

元素名称	中文名	描述
Station	站点	定义在对象内部一个具有容量约束的地方。
Timer	计时器	根据IntervalType触发事件流。
Failure	失效	为父对象定义失效（故障）状态。
BatchLogic	批量逻辑	把实体匹配在一起，并绑定到父实体上，合成一个批次。
Storage	存储	定义一个队列，以指定的顺序临时存储一个或多个实体（逻辑上队列）。
Monitor	监控器	当状态变量变化或者超越某个指定的阈值时，发起一个事件。
Network	网络	定义多个连接形成的网络。
RoutingGroup	路由组	定义路由的逻辑，从一个候选的节点列表里选择一个目的地节点。
OutputStatistic	输出统计	在每次运行结束的时候对某个表达式的值进行记录和输出。
StateStatistic	状态统计	对某个状态变量记录时间持久型的统计数值（时间加权型）。
TallyStatistic	计数统计	保存观察值的摘要统计量，使用Tally步骤记录。
Changeover Logic	切换逻辑	定义一系列随着时间进行的活动（基于排序的切换）。
Material	物料	定义那些可以生产或消耗的物料，或者具有物料清单的物料。

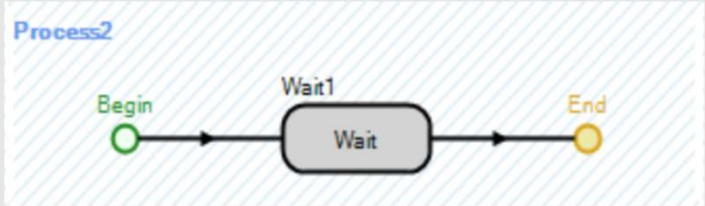
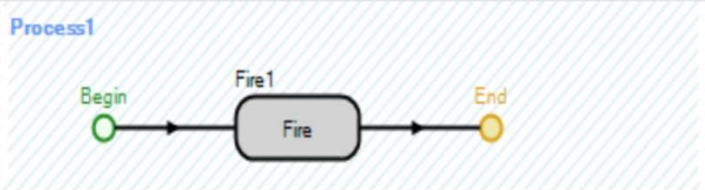
基于事件的决策逻辑

- ▶ “事件”是被触发用于在对象间沟通的信息。
- ▶ “事件”用于通知其他对象，有重要的事情发生了，这时可以加入定制化逻辑。
- ▶ 事件可以由模型逻辑所触发或者自动产生。
- ▶ 系统自动事件：
 - ✓ 实体进入/离开节点，开始移动/结束移动，被销毁时
 - ✓ 资源被分配或者释放，容量变化，失效时，或者修复时
 - ✓ 服务器的能力变化，宕机，或者修复
 - ✓ 工人/车辆进入或离开节点，开始移动/结束移动，分配和释放，改变容量，失效或者修复。
- ▶ 常见的事件触发器
 - ✓ **Station元素**：Entered, Exited, CapacityChanged
 - ✓ **Timer元素**：Timer1.Event
 - ✓ **Monitor元素**：Monitor1.Event
 - ✓ **Fire过程步骤**：用于触发一个用户定义的事件。



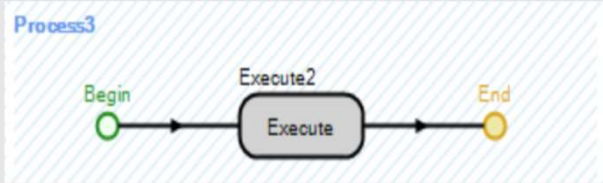
用户自定义事件

▶ 用户在过程逻辑中触发 (Fire)自定义事件。



Properties: Wait1 (Wait Step Instance)	
<input type="checkbox"/> Show Commonly Used Properties Only	
Basic Logic	
Event Name	
Event Condition	
Events (More)	6 Rows
Advanced Options	
Token Wait Action	WaitUntilAnyEvent
Exclusion Expression	WaitUntilAnyEvent
General	
Name	Wait1

Properties: Process3 (Process Element)	
<input type="checkbox"/> Show Commonly Used Properties Only	
Basic Logic	
Triggering Event Name	Event1
Triggering Event Condition	
Advanced Options	
General	



步骤 (Steps)



- ▶ 步骤执行行动，例如**seize** (抓取资源)，**delay** (延迟)，**decide** (判断)，**wait** (等待)等。
- ▶ 步骤是没有状态的。
- ▶ 步骤可以改变元素/令牌/实体/对象的状态。
- ▶ 步骤按组归类 (常用, 全部, 用户定义)。
- ▶ 步骤可分为通用步骤，和专用步骤 (例如运输器专用的PickUp步骤)。

练习-工艺炉

- ▶ 创建一个Source-Server-Sink模型，只产生一个实体，服务器的加工时间为60分钟。
- ▶ 在模型层面添加一个Level类型的状态变量，命名为OvenTemp。
- ▶ 创建服务器的“Processing”附加过程，实现如下内容：
 - ▶ 设定 $OvenTemp.Rate = 100$ (Assign步骤) ,
 - ▶ 等待 20分钟(Delay步骤) ,
 - ▶ 设定 $OvenTemp.Rate = -50$,
 - ▶ 等待 40 分钟,
 - ▶ 设定 $OvenTemp.Rate = 0$ 。
- ▶ 添加状态标签显示OvenTemp 变化率。
- ▶ 添加折线图显示OvenTemp 的值。

- ▶ 高级: 在服务器加工时间里实施上面的过程?

常用步骤

步骤名	行为
Assign	给状态变量赋值或赋表达式
Decide	根据判断表达式将令牌发往两个离开点之一
Delay	将令牌延迟指定的时间
Execute	执行一个过程，等待或者继续
Find	搜索一个数组找到某个索引值
Tally	对指定的Tally统计元素进行计数
Create	创建一个新实体对象
Destroy	销毁父对象或关联实体对象
Fire	触发指定的对象事件
Move	将被抓取的一个或多个移动资源发起移动请求
Wait	等待某个指定的事件的触发
Search	搜索命令用于从对象集合中，或从表格中搜索符合条件的内容
Seize	抓取一个或多个资源对象
SetNode	设置实体对象的目的地节点属性
Release	释放一个或多个资源对象
Transfer	将关联实体在对象之间，或者自由空间或者站点之间进行转移
EndTransfer	完成将相关实体转移到对象或者站点内部的工作

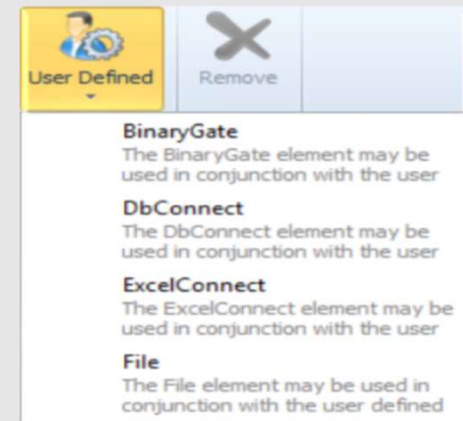
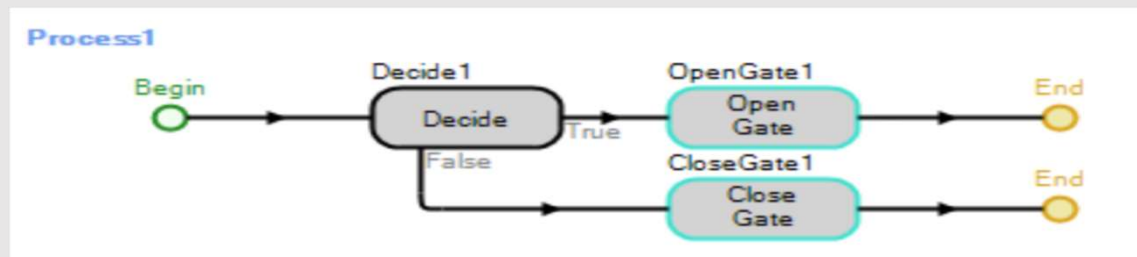
高级步骤

- ▶ 除了我们列出的常用步骤之外，Simio提供了模型运行时对模型对象进行控制和操作的更多高级步骤，我们列出在项目中用的相对较多的步骤。
- ▶ 过程步骤的学习是Simio系统建模中的最核心内容，也是构建复杂逻辑的基础。

步骤名	描述
Allocate	手动触发对父对象的资源分配
Batch	使用BatchLogic (批处理逻辑) 元素将关联对象合并成一个批次
Consume	消耗指定数量的物料元素
Fail	指定某个失效元素进入失效状态 (故障模式)
Insert	将关联对象或父对象插入到队列
Interrupt	用于打断某个过程中的延迟步骤
Notify	输出用户可自定义的跟踪记录，警告和错误信息。
Produce	产生指定数量的物料元素
Remove	从队列中移去关联对象或者父对象
Repair	指定失效元素，将其状态从失效模式恢复成修复状态
Resume	恢复被暂停的过程，或者恢复父对象/关联对象的继续移动
SetNetwork	为父对象或关联对象设定其当前网络
SetRow	为令牌或者父对象/关联对象设定表格引用和行索引
StartTasks	开始某个对象所关联的任务序列操作
Subscribe	为某个过程设定一个触发事件
Suspend	暂停一个过程，或者暂停父对象/关联对象的移动
Travel	在自由空间设定实体的移动轨迹
UnBatch	从关联对象实体的成员队列移除合并成员
UnSubscribe	删除过程的某个触发事件

用户自定义的步骤和元素

- ▶ Simio有开放的.NET Framework 框架，支持用户开发自定义步骤和元素。
- ▶ 过程步骤/元素可以使用任何.NET语言来编写代码 (比如 Visual Basic, C#,C++)。
- ▶ 用户自定义的步骤有他们自己的属性，但是使用起来和内置的步骤一样。自定义步骤是浅蓝色显示的。



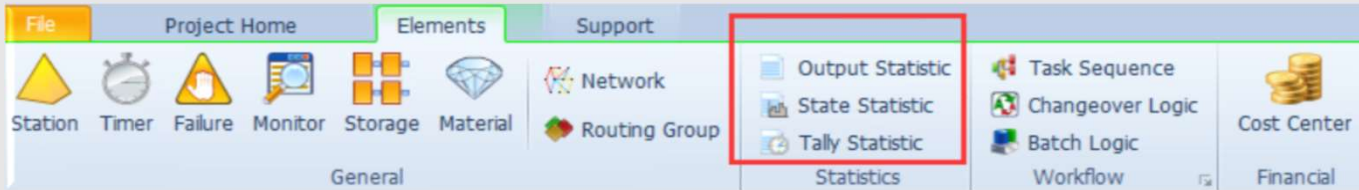
过程的使用

统计

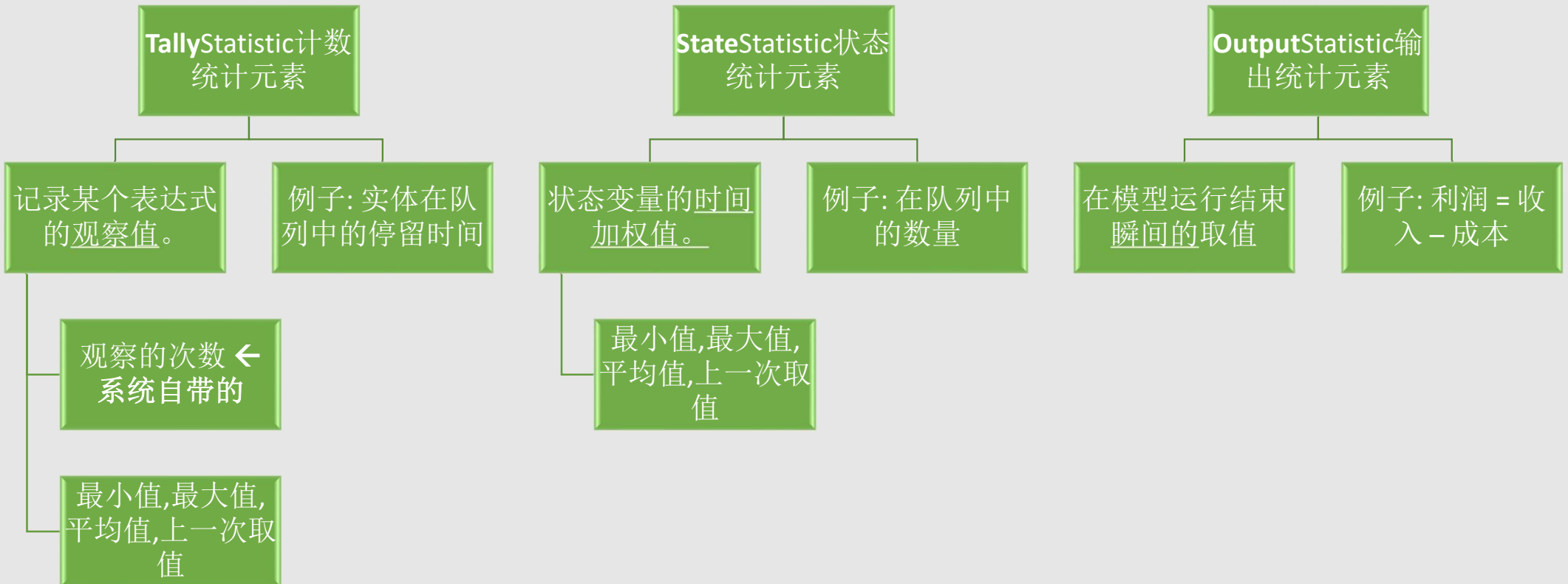
- Tally statistic 元素 + Tally 步骤
- State statistics 元素 - 定义状态变量, 更新状态

逻辑

- 判断
- 延迟
- 设定目的地
- 搜索...

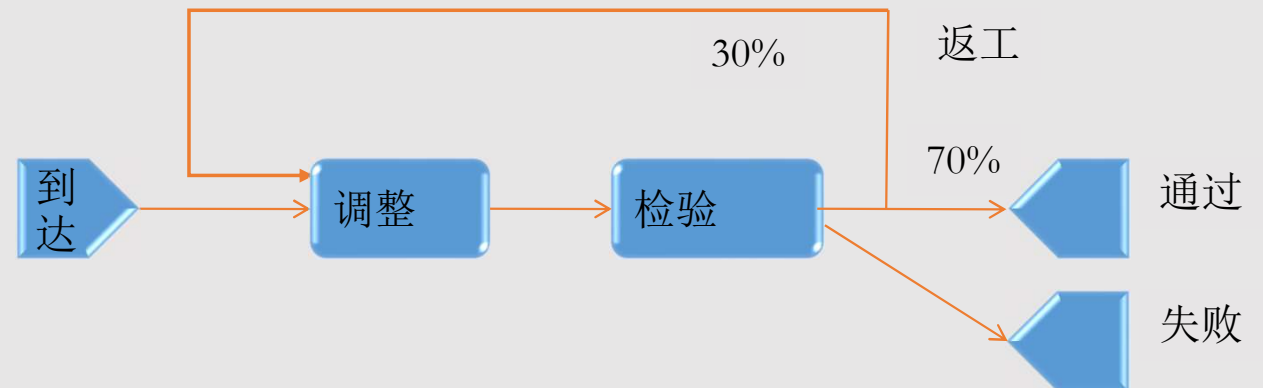


统计元素



- ▶ 到达 – 调整 – 检验 – 返工 (30%).
- ▶ 废品的颜色标记为“红色”
- ▶ 允许在报废前有三次调整的机会。
- ▶ 记录通过测试的产品的调整次数。
- ▶ 确认模型的表现，现在开始...

赋值&决策过程



继续Agony机场

完善Agony机场模型，对晚到的旅客记录。

旅客在航班起飞前0.4-1.2小时 (均匀分布) 到达 (比如，他们的航班在他们进入门口后那么长时间内要起飞)。

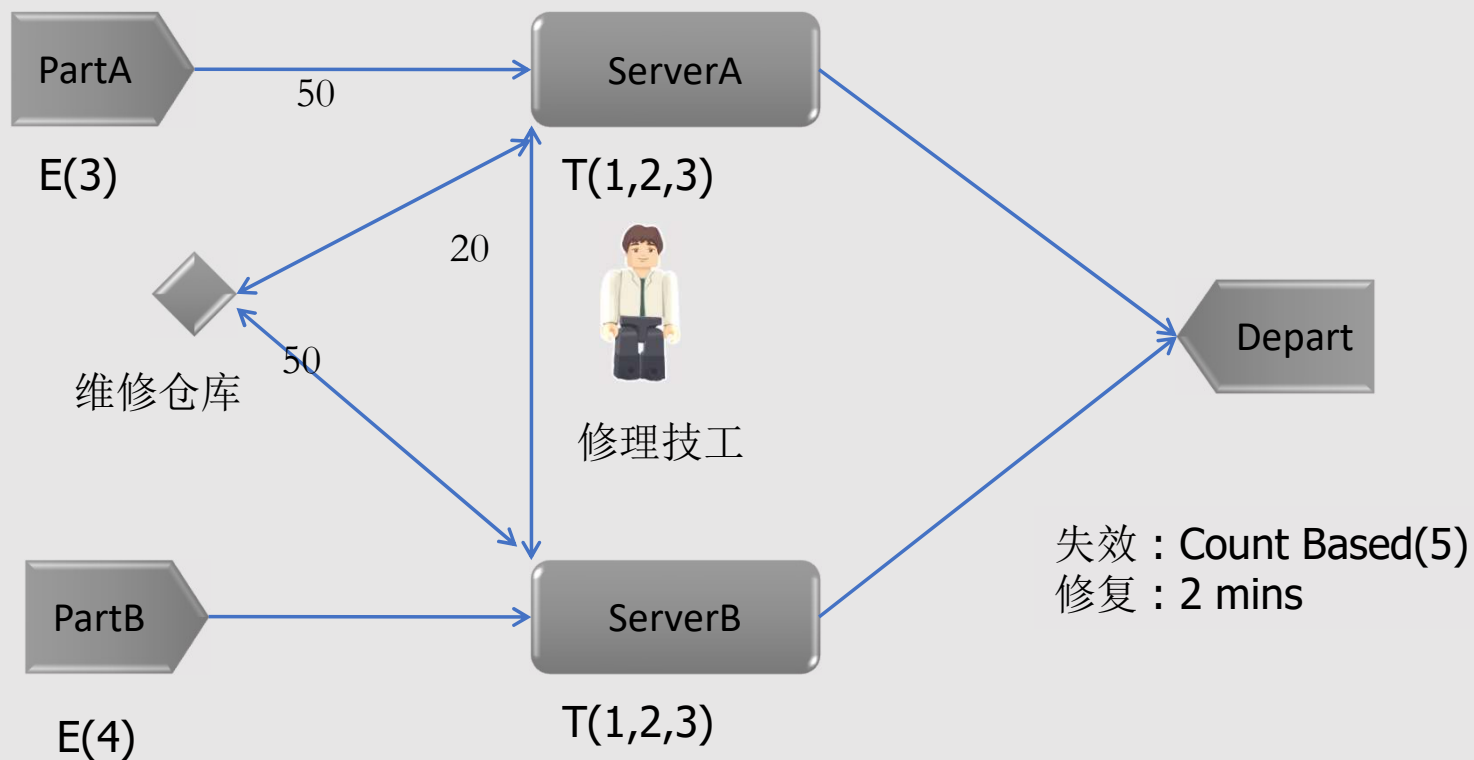
为模型增加逻辑，记录晚到的旅客的数量和晚到的时间。

Agony机场提示

- ▶ 给ModelEntity 增加一个状态变量**FlightTime**，用于存储每个客户的航班起飞时间。在Source的附加过程中对航班起飞设置为TimeNow+一个分布函数。
- ▶ 添加一个Tally Statistic元素，取名**LateDepartures** 并用Tally步骤来存储这个输出指标，在登机口(Sink)的附加过程中赋值。
- ▶ 根据**FlightTime < TimeNow**来判断。
- ▶ Tally 表达式为 **TimeNow - FlightTime** 。

例子：修理技工

使用附加过程来实施设备故障和修理工的需求。



练习-MRI和技工

- ▶ 使用标准服务器代表一个核磁共振设备。
- ▶ 使用附加过程，在正常工作时间的基础上，核磁共振设备需要技术员，5分钟用于准备，5分钟用于清洁。
- ▶ 病人每一个小时到达。
- ▶ 服务时间是45分钟。

MRI和技工的提示

- ▶ 使用Resource，名字叫Technician。
- ▶ 使用服务器的Processing附加过程来抓取技工资源, 延迟5分钟,然后释放。
- ▶ 使用服务器的AfterProcessing附加过程，获取技工，延迟5分钟，然后释放技工。
- ▶ 其他: 使用任务序列来完成同样的模型，不用Processes。

Simbits: 使用Processes

SeizingVehicle

- 特定的部件需要抓取小车，其他情况小车做正常的运输。

TransferLine

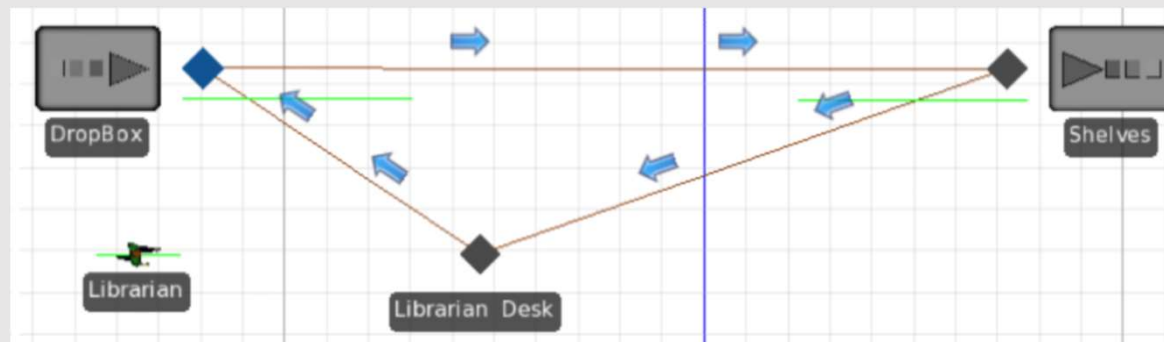
- 同步的移动，定制的节点。

MoveableOperator

- 展示智能的实体行为。

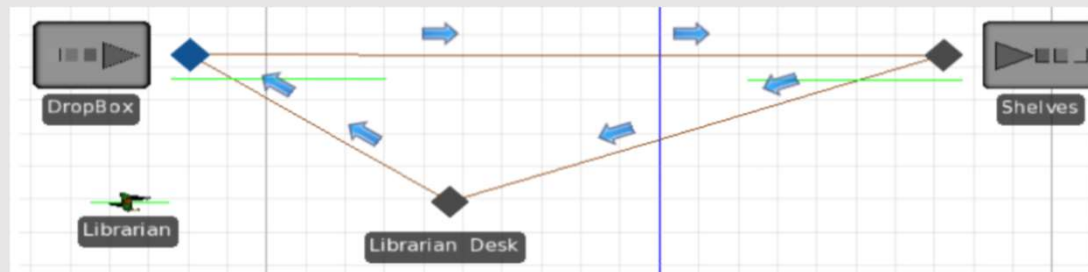
讨论：图书馆还书

- ▶ 每0.25分钟(指数分布)书被归还到图书馆的一个箱子内。
- ▶ 图书管理员等至少有10本书,才开始处理这些书,并且一下子移动到书架上。
- ▶ 用折线图显示等待的书籍数量。



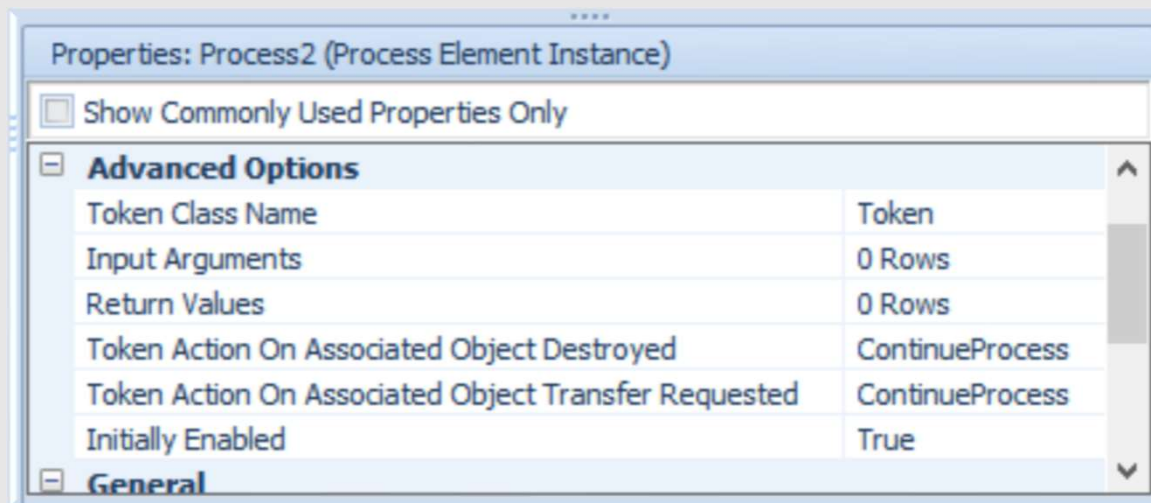
讨论提示 - 图书馆还书

- ▶ 使用发生器和接收器分别代表进来的书和归还到书架的书。
- ▶ 使用Worker来表示图书管理员。
- ▶ 增加过程来评估是否要搬运(Evaluating Transport Request附加过程)。
 - ▶ **Decide** `DropBox.OutputBuffer.Contents < 10 && Worker.NumberRiders == 0`
 - ▶ **Assign** `Token.ReturnValue = -1` (拒绝)。



高级的过程选项

- ▶ 使用定制化令牌传递信息。
- ▶ 输入参数（Input Arguments）和返回值（Return Values）功能在多个“调用”位置共享过程。
- ▶ Token Actions（令牌行动）和关联对象的变化有关。



过程参数的传递

- ▶ 输入参数(Input Arguments) 和返回值 (ReturnValues) 使得用户可以在多个调用的地方共享过程。
- ▶ 和定制化的令牌和状态结合使用。

The screenshot illustrates the configuration of process parameters in a software application. It shows four main windows:

- Return Values - Repeating Property Editor:** This window is used to define return values. It has an 'Items' list with the entry 'Count, How many passes it requires, MyToken1.Count' and a 'Properties' section with 'Basic Logic' containing 'Name: TimeStarted', 'Description: the time process starts', and 'Expression: MyToken1.TimeStamp'. A 'Description' field is also present for the returned value.
- Input Arguments - Repeating Property Editor:** This window is used to define input arguments. It has an 'Items' list with the entry 'WhichTally, The name of the Tally' and a 'Properties' section with 'Basic Logic' containing 'Name: WhichTally', 'Description: The name of the Tally where you want', and 'State Variable Name: MyToken1.MyTally'. A 'Name' field is also present for the input argument.
- Advanced Options:** This window shows configuration for the token class 'MyToken1'. It includes 'Token Class Name: MyToken1', 'Input Arguments: 1 Row', 'Return Values: 2 Rows', and 'Token Action O...: ContinueProcess'.
- Properties: Server1 (Server):** This window shows the configuration for the server. It includes 'Run Initialized', 'Run Ending', 'Entered: Process1', 'Input Arguments', 'Returned Values', 'Count', and 'Time Started'. The 'Which Tally' property is highlighted in orange.

Red arrows indicate the flow of data from the 'Input Arguments' window to the 'Return Values' window, and then to the 'Advanced Options' window. A blue arrow points from the 'Advanced Options' window to the 'Which Tally' property in the 'Properties: Server1 (Server)' window.

过程的总结

- ▶ 过程是随着时间而发生的一系列活动，它可以改变系统的状态。
- ▶ 过程为精确模拟您的系统提供了极大的*灵活性*。
- ▶ 过程可以自动触发，由事件触发或者由附加过程触发器触发。
- ▶ 可以执行更为复杂的逻辑，例如搜索，事件管理，决策，赋值，和转移等等。



第八章



解释仿真结果



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

主要内容

纠错技巧

响应图 (SMORE图)

OptQuest优化

纠错 (Debugging)



为什么我建的
模型的行为是
这样子的?!

Simio纠错方法

不同的模型错误种类

动画技巧

断点（**Breakpoints**）

跟踪（**Trace**）

监控（**Watch**）

单步执行方式

Notify步骤

给步骤标记颜色

观察日志 / 甘特图表 (企业版)

运行时性能评测（**Profile**）

特定的仿真运行样本

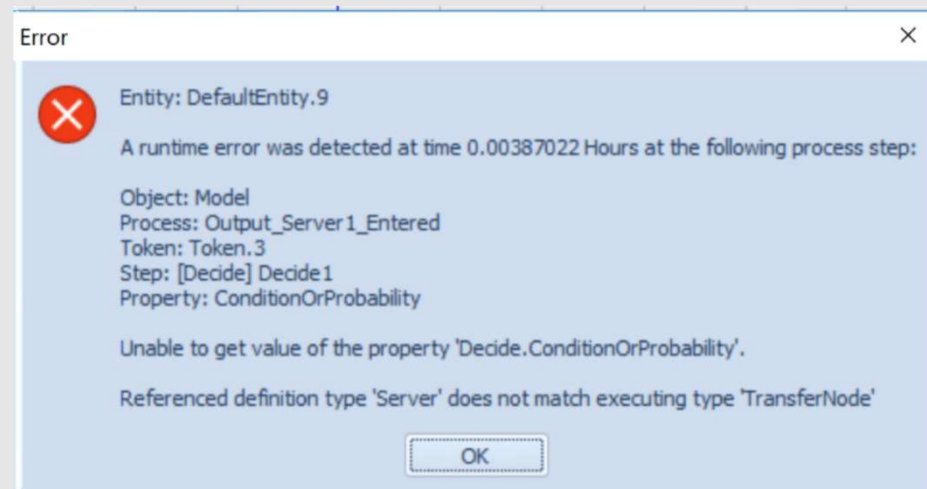
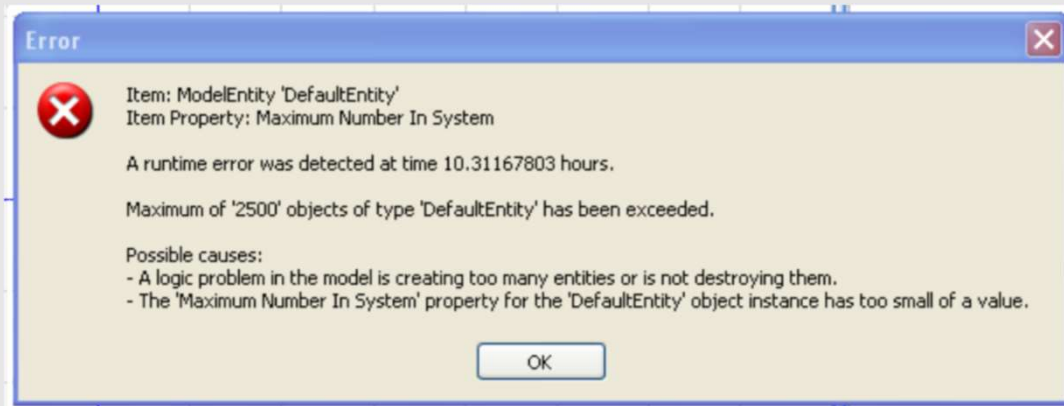
检测 and 解决模型错误

- ▶ 模型纠错是仿真研究重要的部分。过程可能非常耗时，或者令人沮丧。
- ▶ 通常来说，在仿真模型的开发和实施过程中会遇到三类仿真错误：
 - *语法错误* - 对于语言结构/模块的错误使用
 - *运行时错误* - 在模型运行时才发生的程序错误（例如：内存溢出）
 - *逻辑错误* - 模型逻辑中存在着隐含错误，通常导致模型产生令人意外的行为。

语法错误

- ▶ 绝大多数商业化的系统仿真软件（包括**Simio**）在模型开发和初始化（编译）阶段会自动搜索语法错误，检测到的常见错误有：
 - 关键字或其他系统预留变量名的错误拼写
 - 对象名称和其他“字符串”的重复使用
 - 未定义的状态变量，资源，统计变量等等
 - 未连接的对象
- ▶ 当**Simio**发现错误时，将错误反馈给用户，**Simio**提供了查找错误的功能 并列出的可能的错误原因。
- ▶ 语法错误通常容易被发现和修复（用户在理解**Simio**的语法规则后）。

运行时错误



逻辑错误

- 虽然Simio在查找和修复语法错误和运行时错误时非常有帮助，但是通常来说，并不会自动的侦测到模型逻辑错误。
- 当逻辑错误发生时，模型依然会编译并执行，但是错误的逻辑会使模型的行为和结果出现偏离/错误。错误比较隐晦。
- 系统仿真建模的大量时间是花在解决这类错误上。
- 您纠错的效率和有效性等同于您作为仿真建模者的有效性。

“纠错不够快，是因为错的不够多！” -Simio建模者语录

动画技巧

2D动画（顶部视图）

3D动画

“H” 显示提示语

“R” 旋转, “Esc”
停止

“W” 在地面上
行走

增加视图名称,在不同角度轻松切换

命名的视图
(Named Views)

镜头的放置
(Camera
Placement)

动画太慢?

调整速度因子

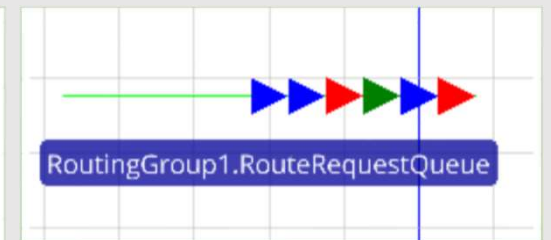
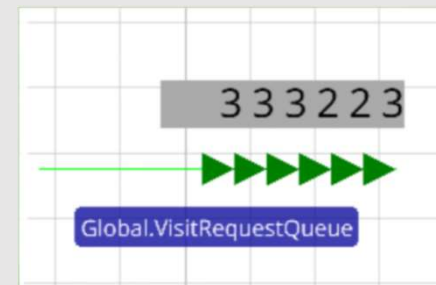
确保Trace功能被
关闭

动画技巧

Status Labels
(状态标签)

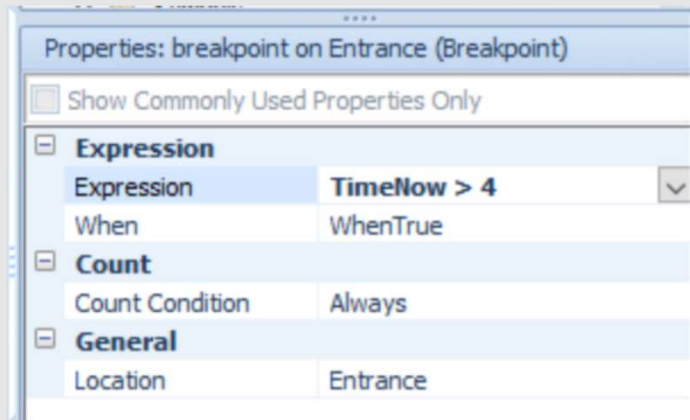


Queues
(队列)



断点的设置

- ▶ Facility窗口- 对象
- ▶ Processes窗口 - 步骤
- ▶ 通过属性来控制断点行为



打开/关闭Breakpoints

When (发生条件) : When True (当为真时)
When Changed (当变化时)

Count Condition (计数条件) : Always (永远)
Equal To (等于)
At Least (至少)

Breakpoints	
<input type="checkbox"/> Location	Expression
<input checked="" type="checkbox"/> Entrance	When 'TimeNow > 4' is true
<input type="checkbox"/> Server1	(none)
<input checked="" type="checkbox"/> BagXRay1	(none)

跟踪(Trace)

- ▶ 跟踪功能被打开后，将提供系统运行过程发生的所有行为的完整列表。
- ▶ 列宽可以被调整，尽量利用屏幕可用空间。
- ▶ 过滤功能使你聚焦于某些内容的子集。
- ▶ 跟踪的信息自动被复制到一个外部**CSV**文件。
 - ▶ 和当前模型（**SPFX**）在同一个目录下
 - ▶ 可以导入**EXCEL**表格，更好地排序和筛选。
- ▶ 跟踪功能可以和设定断点，单步执行组合使用。

监控窗口 (Watch)

- ▶ 鼠标右键点击打任何对象, 将其添加到监控窗口。
 - ▶ 静态对象
 - ▶ 动态对象
- ▶ 在每个对象的树列表里展开, 查看当前值。
 - ▶ 状态
 - ▶ 函数
 - ▶ 元素
 - ▶ 过程
 - ▶ 令牌
- ▶ 也可以和设置断点一起使用。

监控窗口 (Watch)

Watch	
Name	Value
Four31	
Size	287.49136850065219 Cubic Meters
AllocationQueue	0
CurrentCapacity	1
Cost	0 USD
ResourceState	Processing
TotalMolten	0
LocationID	23
LocationAreaID	8
CurrentCharge	
LocationName	Four31
Si	0
Fe	0
ReserveEmptySpot	False
TruckArrived	False
Functions	
InputBuffer	
TransferInState	0
Contents	1
52 (Crucible 1.3358)	
Size	13.614568851340414 Cubic Meters
AllocationQueue	0
CurrentCapacity	0
Cost	0 USD
Movement	0 Meters
DesiredSpeed	3599996400 Meters per Hour

▶ Facility窗口 Vs Processes窗口

The screenshot displays the Simio software interface. The main workspace shows a facility layout with a Source1, Server1, and Sink1. A DefaultEntity is shown moving through the system. The interface includes a top menu bar, a toolbar with various symbols and tools, and a right-hand panel for model navigation and properties. At the bottom, a trace log window is open, showing a detailed record of the simulation's execution.

Time (Hours)	Entity	Object	Process	Token	Step	Action
0	Source1	Source1	OnEntityArrival	0	[Create] NewObj...	Creating '1' object(s) of type 'DefaultEntity' using creation metho... Object 'DefaultEntity.11' created.
0	DefaultEntity.11	Source1	OnEntityArrival	1	[Transfer] ToOutp...	Entity 'DefaultEntity.11' initiating transfer into station 'Source1.0...
0	DefaultEntity.11	Source1	OnEntityArrival	1	[End]	Process 'OnEntityArrival' ended.
0	Source1	Source1	OnEntityArrival	0	[End]	Process 'OnEntityArrival' ended.
0	DefaultEntity.11	Source1	OnEnteredOutputB...	2	[Begin]	Process 'OnEnteredOutputBuffer' initiated.
0	DefaultEntity.11	Source1	OnEnteredOutputB...	2	[EndTransfer] Int...	Entity 'DefaultEntity.11' ending transfer into station 'OutputBuffer'.
0	DefaultEntity.11	Source1	OnEnteredOutputB...	2	[Transfer] ToOutp...	Entity 'DefaultEntity.11' initiating transfer to parent external node...
0	DefaultEntity.11	Output@Source1	OnEnteredFromAss...	0	[Begin]	Process 'OnEnteredFromAssociatedObject' initiated.
0	DefaultEntity.11	Output@Source1	OnEnteredFromAss...	0	[Fire] EnteredEve...	Firing event 'Output@Source1.Entered'.
0	DefaultEntity.11	Output@Source1	OnEnteredFromAss...	0	[Transfer] ToOutp...	Entity 'DefaultEntity.11' initiating transfer into node 'Input@Serv...
0	DefaultEntity.11	Input@Server1	OnEnteredToAssoc...	1	[Begin]	Process 'OnEnteredToAssociatedObject' initiated.
0	DefaultEntity.11	Input@Server1	OnEnteredToAssoc...	1	[Fire] EnteredEve...	Firing event 'Input@Server1.Entered'.
0	DefaultEntity.11	Input@Server1	OnEnteredToAssoc...	1	[Transfer] ToAsso...	Entity 'DefaultEntity.11' initiating transfer from node 'Input@Serv...
0	DefaultEntity.11	Input@Server1	OnEnteredToAssoc...	1	[End]	Process 'OnEnteredToAssociatedObject' ended.
0	DefaultEntity.11	Output@Source1	OnEnteredFromAss...	0	[End]	Process 'OnEnteredFromAssociatedObject' ended.
0	DefaultEntity.11	Source1	OnEnteredOutputB...	2	[End]	Process 'OnEnteredOutputBuffer' ended.
0	DefaultEntity.11	Server1	OnEnteredInputBuff...	3	[Begin]	Process 'OnEnteredInputBuffer' initiated.

单步模式

Notify步骤

Properties: Notify1 (Notify Step Instance)

Show Commonly Used Properties Only

Basic Logic

Notification Type	Error
Message Heading	"Hello Simio User Group!"
Message Content	"More Than 3 Entities Waiting"

Advanced Options


Notify Condition

Exclusion Expression

General

错误, 警告, 跟踪

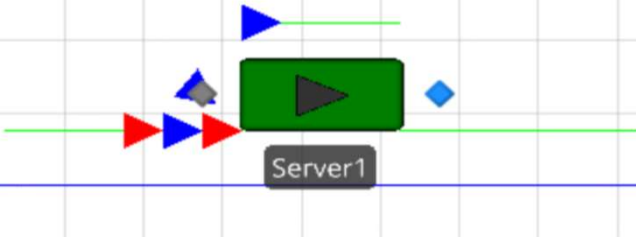
Simio Runtime

 Hello Simio User Group!
More Than 3 Entities Waiting


→ Continue running

→ Pause the simulation

Show details
 Don't show this warning again



Error

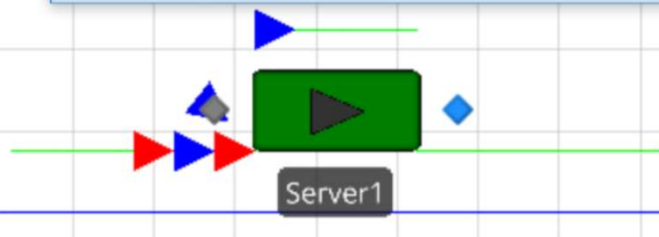
 Object: Model

A runtime error was detected at time 0.01489326 Hours at the following process step:

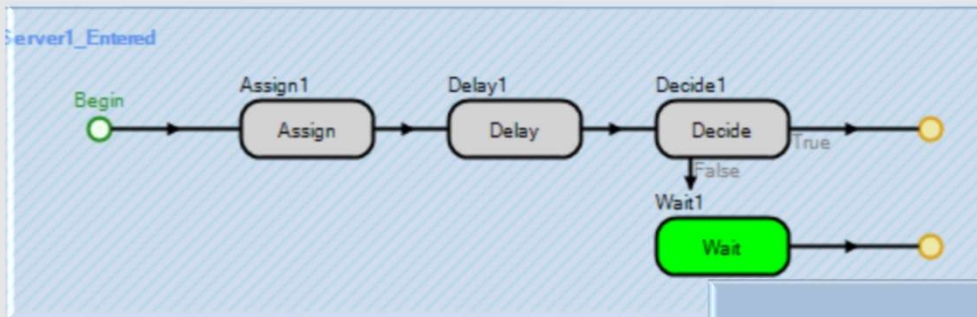
Object: Model
Process: Process1
Token: Token.5
Step: [Notify] Notify1

Hello Simio User Group!
More Than 3 Entities Waiting

OK

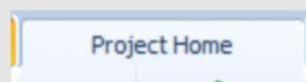


给步骤标记颜色



Step	Action
[Begin]	Process 'Server1.OnEnteredInputBuffer' started.
[Execute] Entered_A...	Executing process 'Server1_Entered'.
[Begin]	Process 'Server1_Entered' started.
[Delay] Delay1	Delaying token for '0.3333333333333333' Hours until time '0.334420634920635' Hours.
[Decide] Decide1	Token branching on condition 'timenow > 1'. Token sent to 'False' exit.
[Wait] Wait1	Token waiting for event 'Timer1.Event'.
	Token waiting at Wait step for '1' event(s).
-	Firing event 'Source1.EntityArrivals.Event'.
[Begin]	Process 'Source1.OnEntityArrival' started.
[Create] Entities	Creating '1' object(s) of entity type 'DefaultEntity' using creation method 'NewObject'.
-	Object 'DefaultEntity.12' created.

搜索窗口

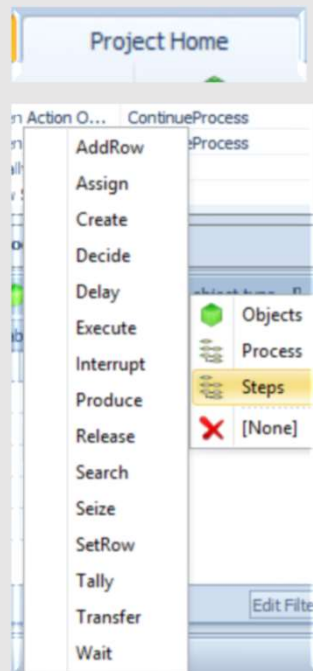


Search - Matches: 107

Value	Property	Item Name	Item Type	Project Item
BulkMinInv	Name	BulkMinInv	Integer Property	Model
DisposeBulkOrders	Name	DisposeBulkOrders	Sink	Model
GreenBulk	Name	GreenBulk	Material Element	Model
GreenBulk	MaterialName	ManufacturingOrders	[Property Group]	Model
GreenBulk	MaterialName	ManufacturingOrders	[Property Group]	Model

双击即可以转到搜索到的对象

属性电子表格 (Property Spreadsheet)



- ▶ 双击可转到搜索到的引用
- ▶ 多选
- ▶ 拷贝和粘贴

Instance Name	Capacity Type	Work Schedule	Initial Capacity	Ranking Rule	Ranking Expression	Dynamic Selection Rule	Transfer-In Time	Units	Process Type	Processing Time	Unit
HoldingTankStorageServer	Fixed		Infinity	First In First Out	Entity.Priority	None	0.0	Minutes	Specific Time	1000	Days
RedPalletStorage	Fixed		Infinity	First In First Out	Entity.Priority	None	0.0	Minutes	Specific Time	1000	Days
GreenPalletStorage	Fixed		Infinity	First In First Out	Entity.Priority	None	0.0	Minutes	Specific Time	1000	Days
BluePalletStorage	Fixed		Infinity	First In First Out	Entity.Priority	None	0.0	Minutes	Specific Time	1000	Days
Receiving	Fixed		Infinity	First In First Out	Entity.Priority	None	0.0	Minutes	Specific Time	1000	Days

观察日志（企业版）

Advanced Options

Speed Factor: 1.000

Adjust Speed: [Slider]

Units Settings

Display

Warning Level
Set the warning reporting level for the model.

Randomness
Advanced options related to random sampling in the model.

Financials
Set financial-related options for the model.

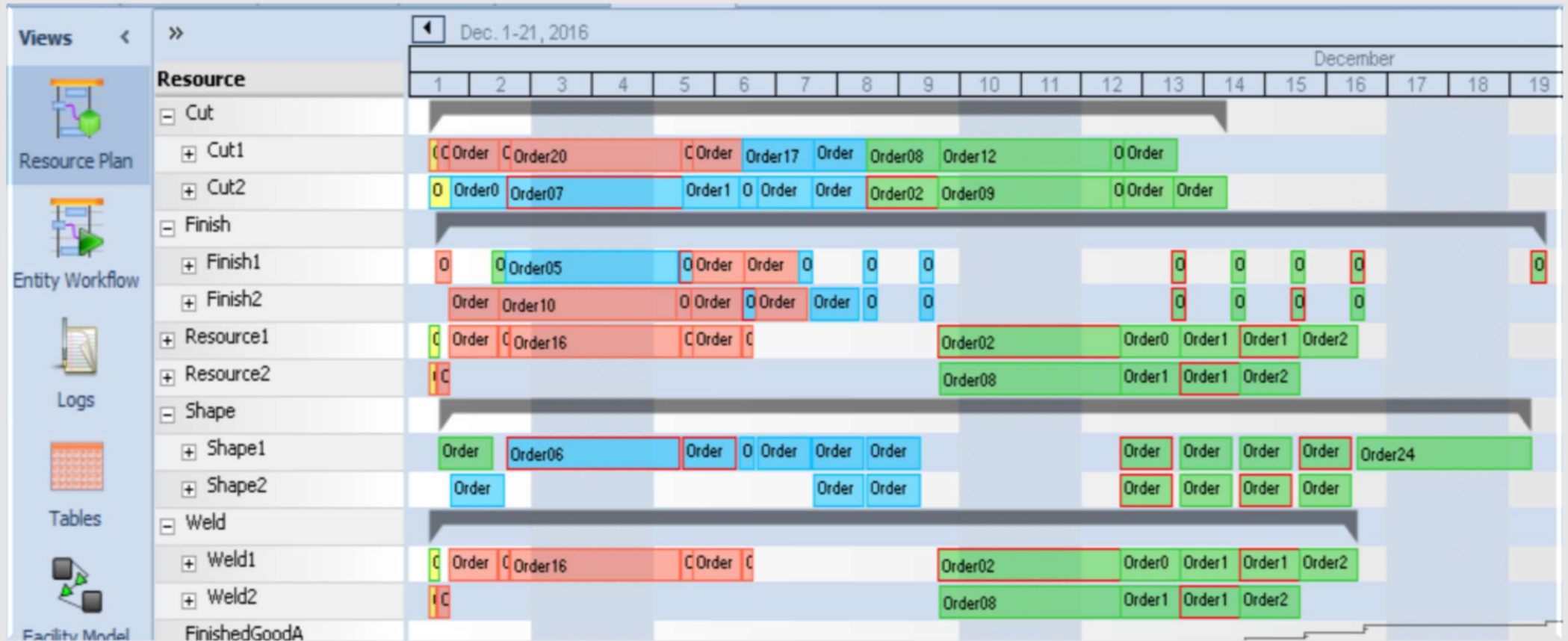
Enable Interactive Logging
Enables logging to object-based logs during Run or Fast-Forward

Enable Run-time Profiling
Records model run-time performance information.

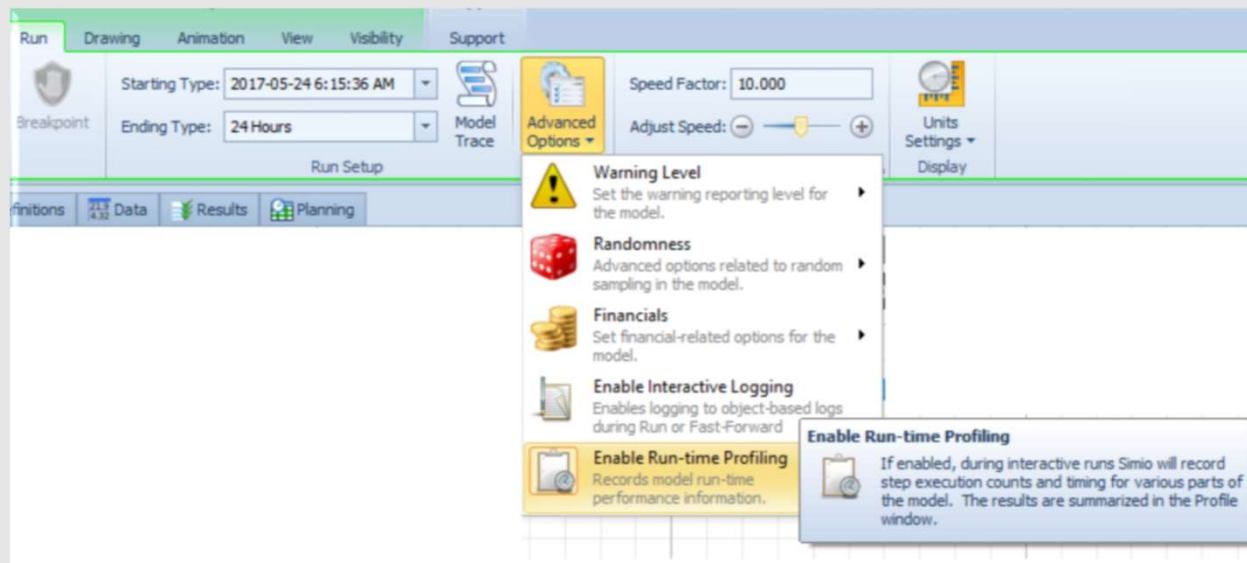
Enable Interactive Logging
In Enterprise licensed software, enables logging to the following logs during a standard interactive run:
 - Resource Usage Log
 - Resource State Log
 - Resource Capacity Log
 - Constraint Log
 - Transporter Usage Log
 - Material Usage Log
 - Task Log

Resource Id	Resource	Resource List	Node List	Owner Id	Owner	Start Time	End Time	Duration (Hours)
Cut2	Cut2		CutList	DefaultEntity.40	Order02	2016-12-08 11:18:21 AM	2016-12-09 3:42:21 PM	28.4
Weld1	Weld1		WeldList	DefaultEntity.40	Order02	2016-12-09 3:42:31 PM	2016-12-12 3:06:31 PM	71.4
Resource 1	Resource 1			DefaultEntity.40	Order02	2016-12-09 3:42:31 PM	2016-12-12 3:06:31 PM	71.4
Shape 1	Shape1		ShapeList	DefaultEntity.40	Order02	2016-12-12 3:06:45 PM	2016-12-13 10:54:45 AM	19.8
Finish1	Finish 1		FinishList	DefaultEntity.40	Order02	2016-12-13 10:54:56 AM	2016-12-13 4:18:56 PM	5.4

甘特图（企业版）



运行时性能评测



Item	% Time in Parent	% Time Overall	Count	Plot
> InitializeCrucibles.GetDropoffLocation [Execute]	16.70	6.8093	1	
▼ SearchEmptyControl_D.Execute1 [Execute]	16.52	6.7362	12	
▼ SearchForEmptyCrucible_D.FindEmptyCrucible [Search]	99.09	6.6751	21	
▼ SearchForEmptyCrucible_D.GetNextEmptyDropoffLocation [Execute]	99.87	6.6663	5	
> GetNextEmptyDropoffLocation.PotGroupDirection [Search]	99.97	6.6643	5	
GetNextEmptyDropoffLocation.Find1 [Find]	0.01	0.0008	5	

双击某一行，转到逻辑所在位置

特定的仿真运行样本

► Run菜单

The screenshot shows the 'Run' menu in a simulation software. The menu is open, displaying several options. The 'Replication Number' option is highlighted with a red circle, indicating it is the current focus. The 'Replication Number' is set to 1. The 'Advanced Options' button is visible in the top left corner of the menu. The 'Speed Factor' is set to 50.000. The 'Adjust Speed' slider is also visible. The 'Units Settings' and 'Display' buttons are in the top right corner. The 'Warning Level' option is described as 'Set the warning reporting level for the model.' The 'Randomness' option is described as 'Advanced options related to random sampling in the model.' The 'Financials' option is described as 'Set financial-related options for the model.' The 'Enable Interactive Logging' option is described as 'Enables logging to object-based logs during Run or Fast-Forward.' The 'Enable Run-time Profiling' option is described as 'Records model run-time performance information.'

Advanced Options

Speed Factor: 50.000

Adjust Speed: - +

Units Settings

Display

Warning Level
Set the warning reporting level for the model.

Randomness
Advanced options related to random sampling in the model.

Financials
Set financial-related options for the model.

Enable Interactive Logging
Enables logging to object-based logs during Run or Fast-Forward

Enable Run-time Profiling
Records model run-time performance information.

Disable Randomness
Indicates whether random sampling will be disabled at runtime.

Replication Number
Use the same random sampling as this experiment

1

发现微妙的问题

- ▶ 在运行模型之前，计算出你的期望值。
- ▶ 仔细地观察仿真动画。
- ▶ 增强动画，使它能够提供更多信息。
 - ◆ 绑定的标签
- ▶ 仔细地检查输出统计。
- ▶ 充分利用软件已有的纠错工具。

SMORE图（风险误差指标）

最大值

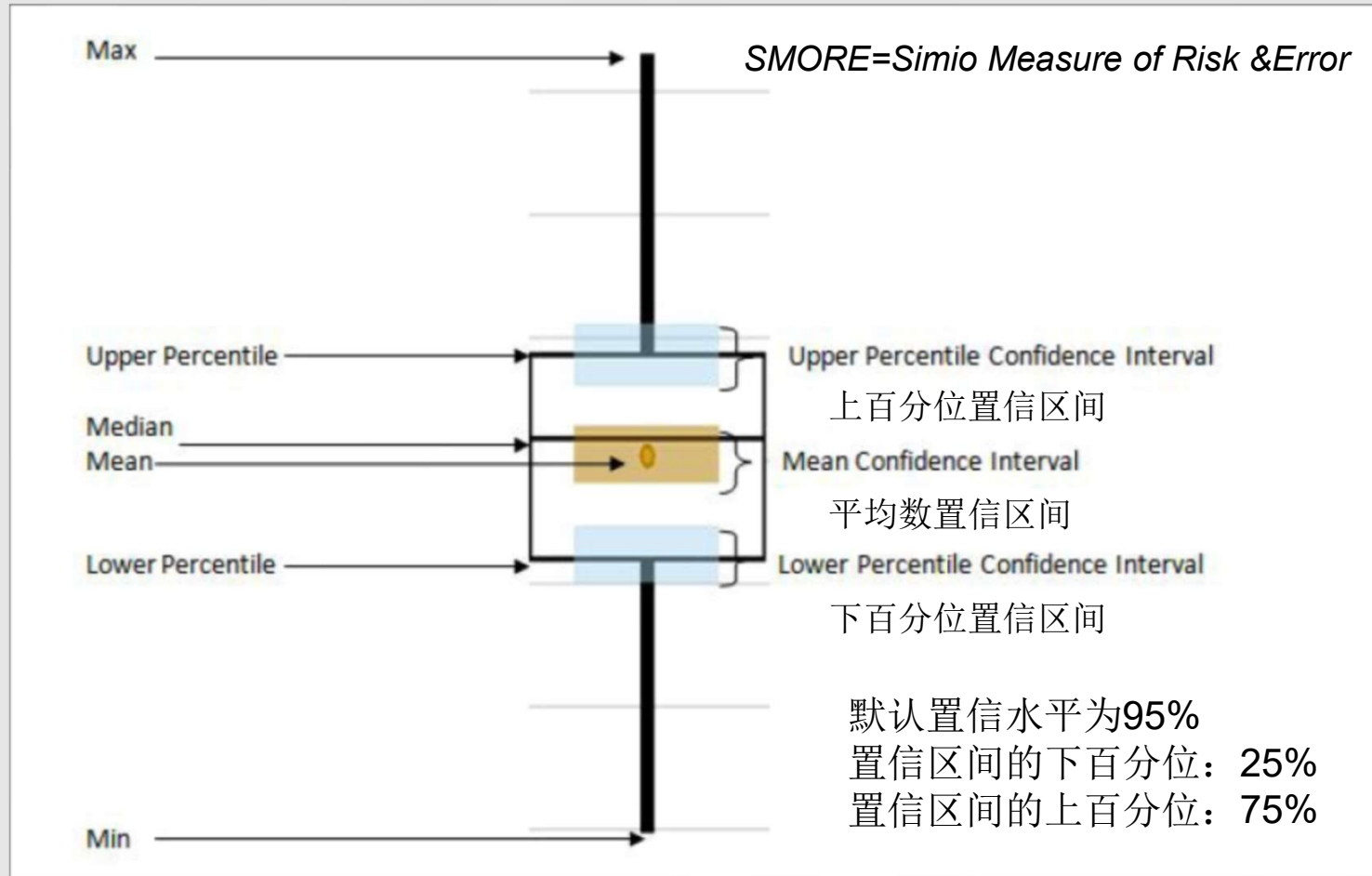
上百分位

中位数

平均数

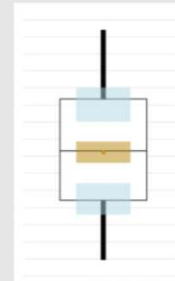
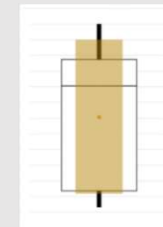
下百分位

最小值



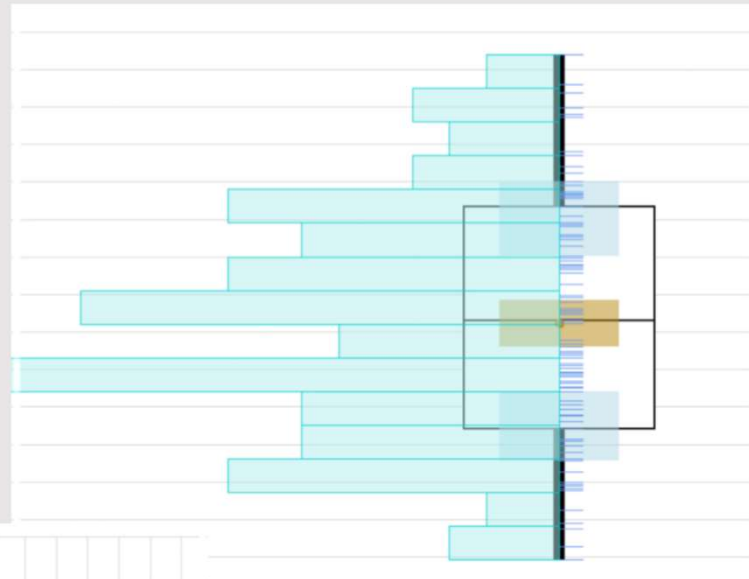
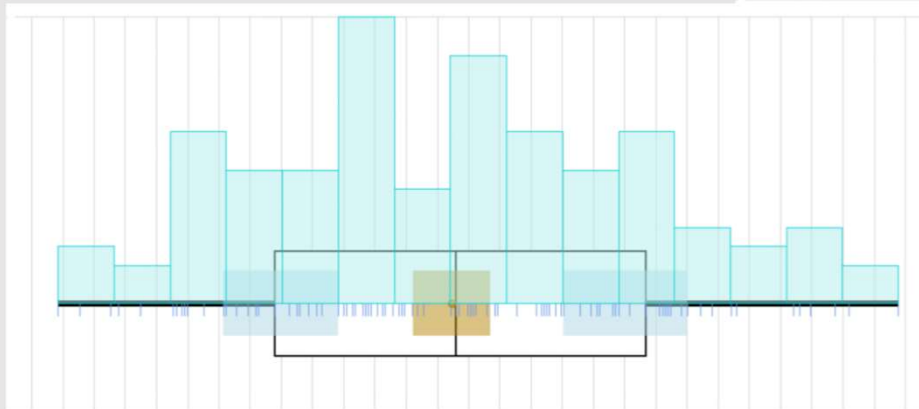
不同等级的数据量

- ▶ 1 次运行
 - 单个响应值
- ▶ 2-5 次运行
 - 最大值, 最小值, 平均值
- ▶ 6+ 次运行
 - 上 + 下百分位数, 中位数, 均值置信区间
- ▶ 大量的统计学有效的结果
 - 上 + 下百分位数的置信区间

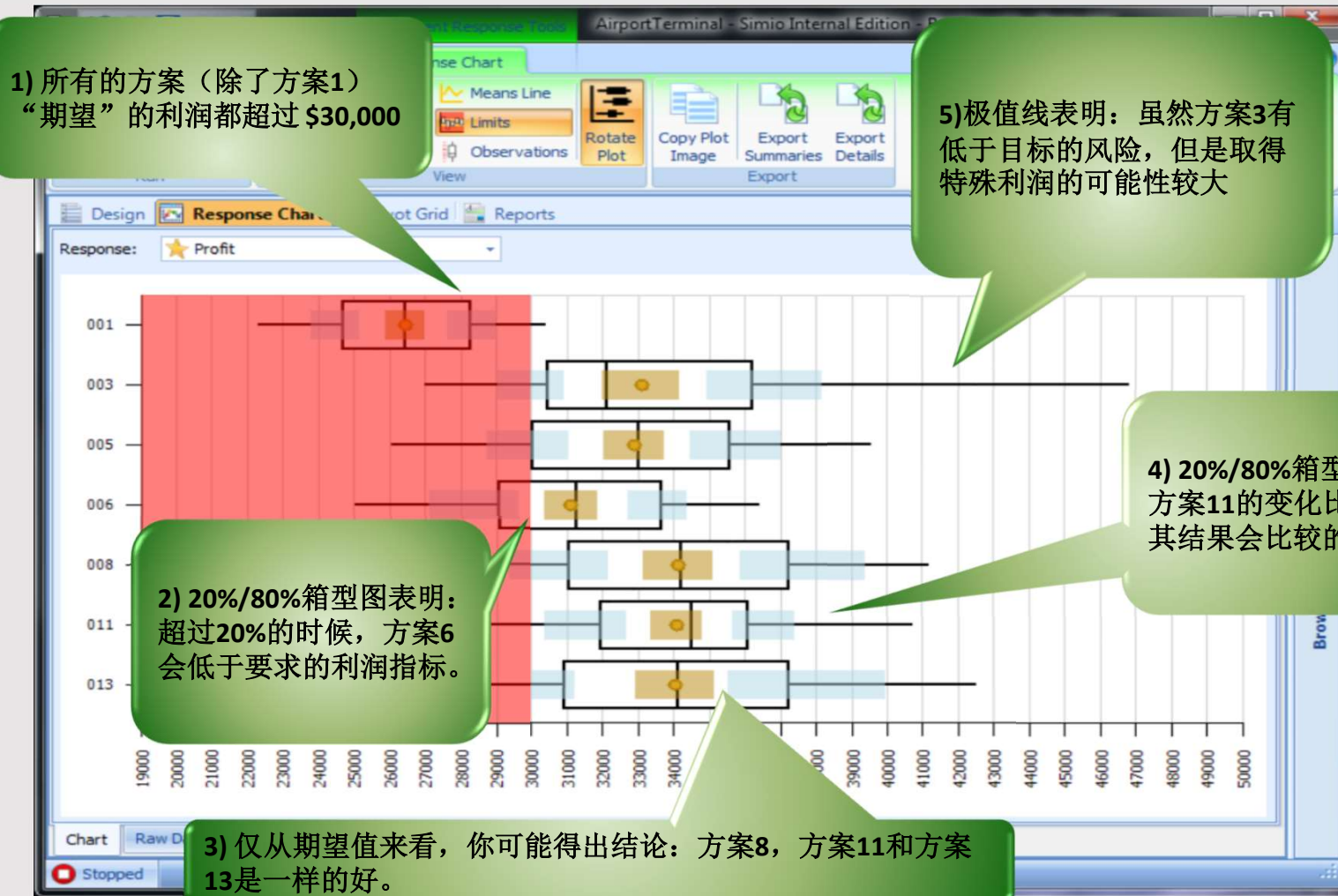


观察原始数据

- ▶ 可以以垂直方向和水平方向展示原始数据的分布



SMORE图解



找到最佳方案 (OptQuest)



定义“最优”

► 这并不容易...



“Someone calling themselves a customer says they want something called service.”

定义“最优”

▶ 湖泊最深处在哪里？



确定“最优方案”

- ▶ 视觉检查
 - ▶ 观察响应图，找到“最佳”的方案
- ▶ 使用KN插件选择最优方案
- ▶ 使用优化插件OptQuest for Simio

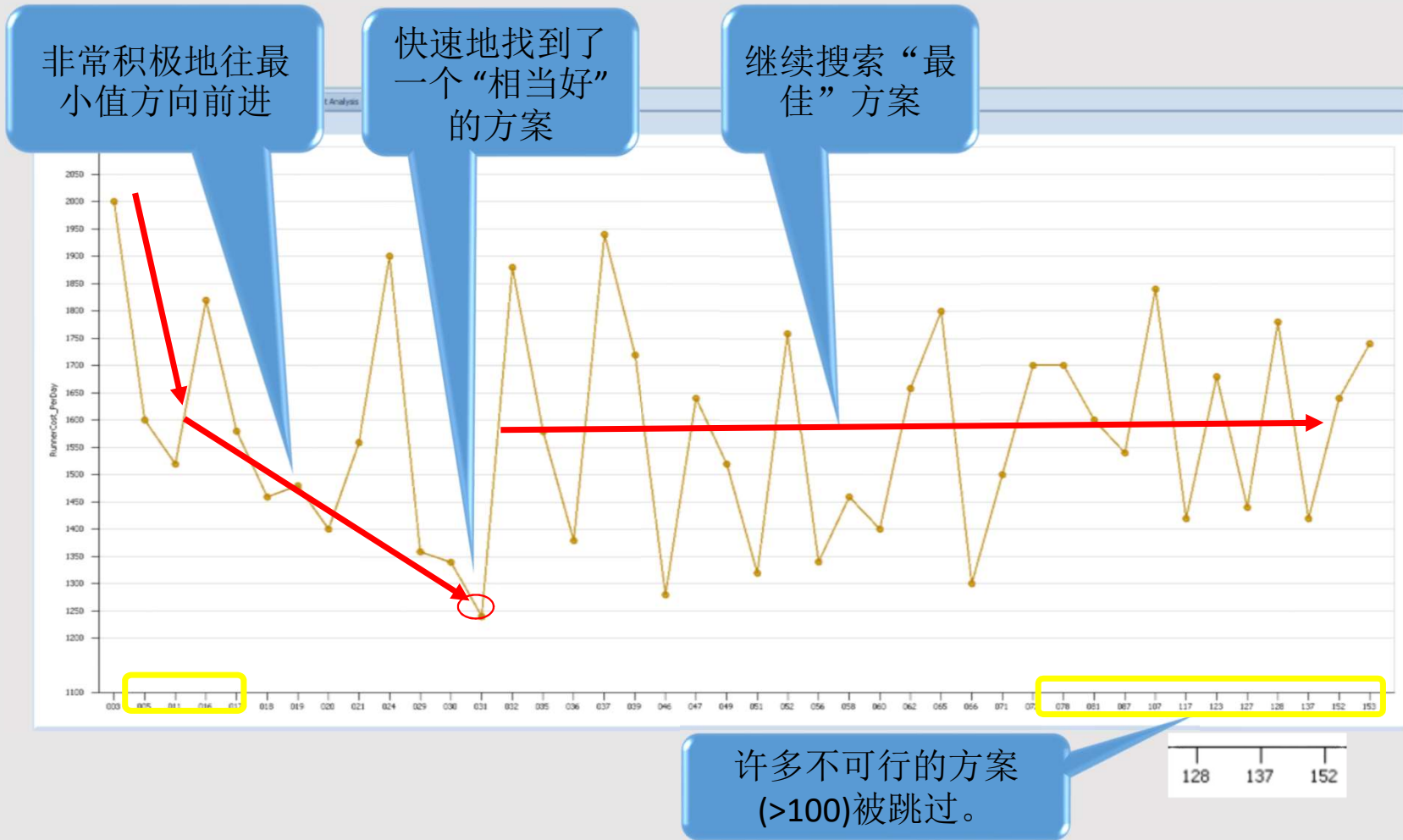
OptQuest for Simio 插件

- ▶ 替代了传统的“试错”方法的不精确性。
- ▶ 快速地确定哪些控制变量能够最好满足目标。
- ▶ 元启发式搜索算法获得更佳方案
 - ▶ 记忆哪些方案是较优的，然后将他们重新整合到新的更佳方案中。
 - ▶ 不会陷入局部最优的陷阱，或者被模型的噪音数据误导优化方向。
- ▶ 结合了禁忌搜索法，分散搜索法，整数规划，神经网络，整合为一个复合的搜索算法（知识产权）。
- ▶ 寻优速度比其他方法快几个数量级，根据优化理论的专家评价，OPTQUEST是当今优化器市场上最为可靠的优化软件。

OptQuest for Simio插件

- ▶ OptQuest设定控制变量的值（引用属性），开始仿真运行，获取结果。
- ▶ 插件为所有的控制变量添加了参数，允许用户定义OptQuest如何在输入值中推进。
- ▶ 约束条件(**Constraints**)可以进一步限制输入值，并引导算法逼近最佳方案。
 - ▶ 例如, $\text{Input1} + \text{Input2} + \text{Input3} < 10$
- ▶ 仿真实验可以有多个响应变量的列，定义目标函数：
 - ▶ 单目标(最小化/最大化主要的响应变量)。
 - ▶ 多目标权重
 - ▶ **Pattern Frontier**（模式边界）
- ▶ 一旦控制变量, 约束条件, 和目标函数被确定, OptQuest将非常智能地创建不同的方案使结果向最优方案快速地逼近。
 - ▶ OptQuest负责所有的活!无需人为干预。

OptQuest搜索最小值



OptQuest仿真实验

The screenshot shows the OptQuest software interface for an airport terminal simulation. The main window displays a table of 24 scenarios, each with a name, status, required and completed replications, and various control parameters (Checkin Kiosks, Checkin Clerks, ID Check, Scanner Stations) and response metrics (Revenue, Profit, Cost). The right-hand pane shows the experiment's properties, including analysis parameters like confidence level and primary response.

Scenario		Replications		Controls				Responses		
Name	Status	Required	Completed	Checkin Kiosks	Checkin Clerks	ID Check	Scanner Stations	Revenue	Profit	Cost
001	Compl...	6	6 of 6	5	3	3	3	32149.3	25749.3	6400
002	Compl...	6	6 of 6	1	1	1	1	5808	3698	2110
003	Compl...	9	9 of 9	8	4	5	5	36231.6	25621.6	10610
004	Compl...	8	8 of 8	3	2	2	2	22170.5	17915.5	4255
005	Compl...	9	9 of 9	6	3	4	4	37898.7	29433.7	8465
006	Compl...	6	6 of 6	8	1	5	2	34804	30329	4475
007	Compl...	9	9 of 9	1	1	1	5	11806.7	1696.67	10110
008	Compl...	6	6 of 6	1	4	4	1	7553.33	5218.33	2335
009	Compl...	10	10 of 10	4	4	1	5	21978	11628	10350
010	Compl...	6	6 of 6	8	1	2	3	34730.7	28345.7	6385
011	Compl...	10	10 of 10	1	3	4	5	14854.4	4564.4	10290
012	Compl...	9	9 of 9	6	3	5	1	18343.1	15848.1	2495
013	Compl...	10	10 of 10	3	1	4	3	19294	13024	6270
014	Compl...	6	6 of 6	6	3	1	1	18729.3	16354.3	2375
015	Compl...	10	10 of 10	3	4	5	3	21410.4	14975.4	6435
016	Compl...	6	6 of 6	7	1	5	2	236036	31596	4440
017	Compl...	6	6 of 6	8	1	5	1	16896	14421	2475
018	Compl...	10	10 of 10	6	2	4	4	38306.4	29886.4	8420
019	Compl...	6	6 of 6	6	1	5	2	35911.3	31506.3	4405
020	Compl...	10	10 of 10	8	3	4	4	37703.6	29168.6	8535
021	Compl...	7	7 of 7	8	1	4	2	36174.3	31729.3	4445
022	Compl...	7	7 of 7	7	2	4	3	39354.9	34899.9	4455
023	Compl...	6	6 of 6	8	2	5	3	38932.7	32412.7	6520
024	Compl...	9	9 of 9	6	3	3	4	38187.1	29752.1	8435

Properties: Experiment1 (Experiment)

- Analysis**
 - Warm-up Period: 0
 - Default Replications: 10
 - Confidence Level: 95%
 - Upper Percentile: 80%
 - Lower Percentile: 20%
 - Primary Response: Profit
- OptQuest for Simio - Parameters**
 - Min Replications: 6
 - Max Replications: 10
 - Max Scenarios: 100
 - Confidence Level: 98%
 - Relative Error: 0.1
- General**
 - Name: Experiment1

仿真实验的操作步骤

- ▶ 定义控制变量，响应变量和约束条件。
- ▶ 运行OptQuest优化器/或者手工定义控制变量的取值，创建一组感兴趣的方案。
- ▶ 运行“最小数量”的仿真副本。
- ▶ 使用KN 插件选择最佳方案。

使用KN 算法选择最佳方案

- ▶ 缩小备选方案的范围，找到最佳的方案。
- ▶ 对不同方案使用相同的仿真运行次数，以及Indifference Zone（无差别区域）进行比较
- ▶ 确定是否某个方案从统计学上绝对比其他方案差，进而排除该方案（去掉打勾）。
- ▶ 剩余的方案进行更加多的仿真运行（具体次数由算法自己决定）。
- ▶ 排除绝对差的方案。
- ▶ 此过程周而复始，直到只剩一个方案/或者已经运行了最大允许的仿真次数。

使用KN算法选择最佳方案

The screenshot displays the Simio software interface for an experiment titled "AirportTerminal". The main window shows a table of simulation scenarios with columns for Scenario Name, Status, Replications (Required and Completed), Controls (Checkin Kiosks, Checkin Clerks, ID Check, Scanner Stations), and Responses (Revenue, Profit, Cost). A red box highlights the first 22 rows of the table, and a green box highlights the "Completed" column, showing that all 22 scenarios have completed 10 out of 10 replications. The right sidebar shows the "Analysis" section for "Experiment1", with parameters for "Select Best Scenario using KN - Para..." including Confidence Level (0.95), Indifference Z-score (20), and Replication Limit (100).

Scenario Name	Status	Required	Completed	Checkin Kiosks	Checkin Clerks	ID Check	Scanner Stations	Revenue	Profit	Cost
022	Compl...	10	10 of 10	7	2	4	2	39921.2	35466.2	4455
023	Compl...	10	10 of 10	8	2	5	3	38874	32354	6520
021	Compl...	10	10 of 10	8	1	4	2	36427.6	31982.6	4445
016	Compl...	10	10 of 10	7	1	5	2	35970	31530	4440
006	Compl...	10	10 of 10	8	1	5	2	35371.6	30896.6	4475
019	Compl...	10	10 of 10	6	1	5	2	35248.4	30843.4	4405
024	Compl...	10	10 of 10	6	3	3	4	38368	29933	8435
018	Compl...	10	10 of 10	6	2	4	4	38306.4	29886.4	8420
005	Compl...	10	10 of 10	6	3	4	4	37862	29397	8465
026	Compl...	10	10 of 10	8	2	2	4	37681.6	29251.6	8430
020	Compl...	10	10 of 10	8	3	4	4	37738.8	29203.8	8535
025	Compl...	10	10 of 10	7	1	2	3	35428.8	29078.8	6350
010	Compl...	6	6 of 6	8	1	2	3	34730.7	28345.7	6385
001	Compl...	6	6 of 6	5	3	3	3	32149.3	25749.3	6400
003	Compl...	9	9 of 9	8	4	5	5	36231.6	25621.6	10610
004	Compl...	8	8 of 8	3	2	2	2	22170.5	17915.5	4255
014	Compl...	6	6 of 6	6	3	1	1	18729.3	16354.3	2375
012	Compl...	9	9 of 9	6	3	5	1	18343.1	15848.1	2495
015	Compl...	10	10 of 10	3	4	5	3	21410.4	14975.4	6435
017	Compl...	6	6 of 6	8	1	5	1	16896	14421	2475
013	Compl...	10	10 of 10	3	1	4	3	19294	13024	6270
009	Compl...	10	10 of 10	4	4	1	5	21978	11628	10350
008	Compl...	6	6 of 6	1	4	4	1	7553.33	5218.33	2335
011	Compl...	10	10 of 10	1	3	4	5	14854.4	4564.4	10290
002	Compl...	6	6 of 6	1	1	1	1	5808	3698	2110

第八章总结

- ▶ 比较全面的模型纠错方法包含**Trace**（跟踪），**Watch**(监视)，**Step**（单步执行），**Console**（数据面板）。
- ▶ 仿真结果可以在两个模式下观察：
 - 交互的模式，用于获得系统的洞察力，和校验/验证。
 - 仿真试验，用于预测和比较多个方案。
- ▶ **Simio**通过独有的**SMORE**图和输入分析工具，帮您理解和消除：估计错误，样本量错误和风险。
- ▶ **Simio**提供了选择最佳方案的内置辅助功能。



第九章



建立对象定义



(0510) 82132584



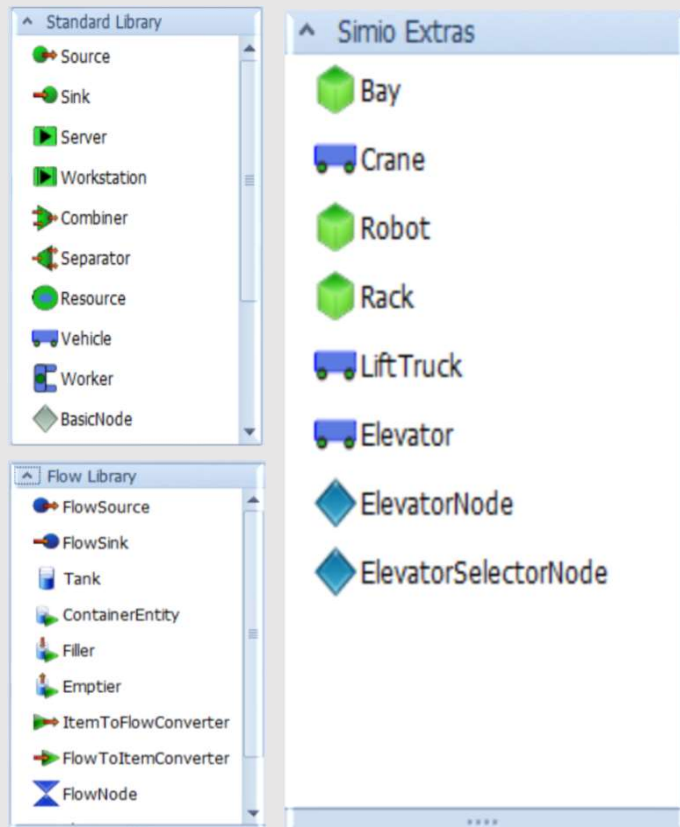
无锡迅合信息科技有限公司©



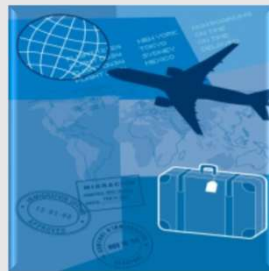
<http://www.simio-china.com/>

对象库

对象模板



- ▶ 模板库是对象定义(或模型)的集合。
- ▶ 模板库为某个特定模型而建,或者在一个应用领域内使用。
- ▶ 模板可以在企业内共享。



对象构建一览

▶ 建立对象和模板库很困难吗？

我们一起来看看...



为什么建立定制化对象

实用对象

- 简单，可重复利用。

复杂对象

- 包含定制化数据和行为。
- 灵活，经过纠错证实。

在更为广泛的用户群分享技能

- 降低对象库的进入门槛。

共享/管理更新

- 当对象定义变化时，所有模型都会得到这个变化。

对象定义构成

对象定义包含三个主要部分:

逻辑和接口可以继承而来。

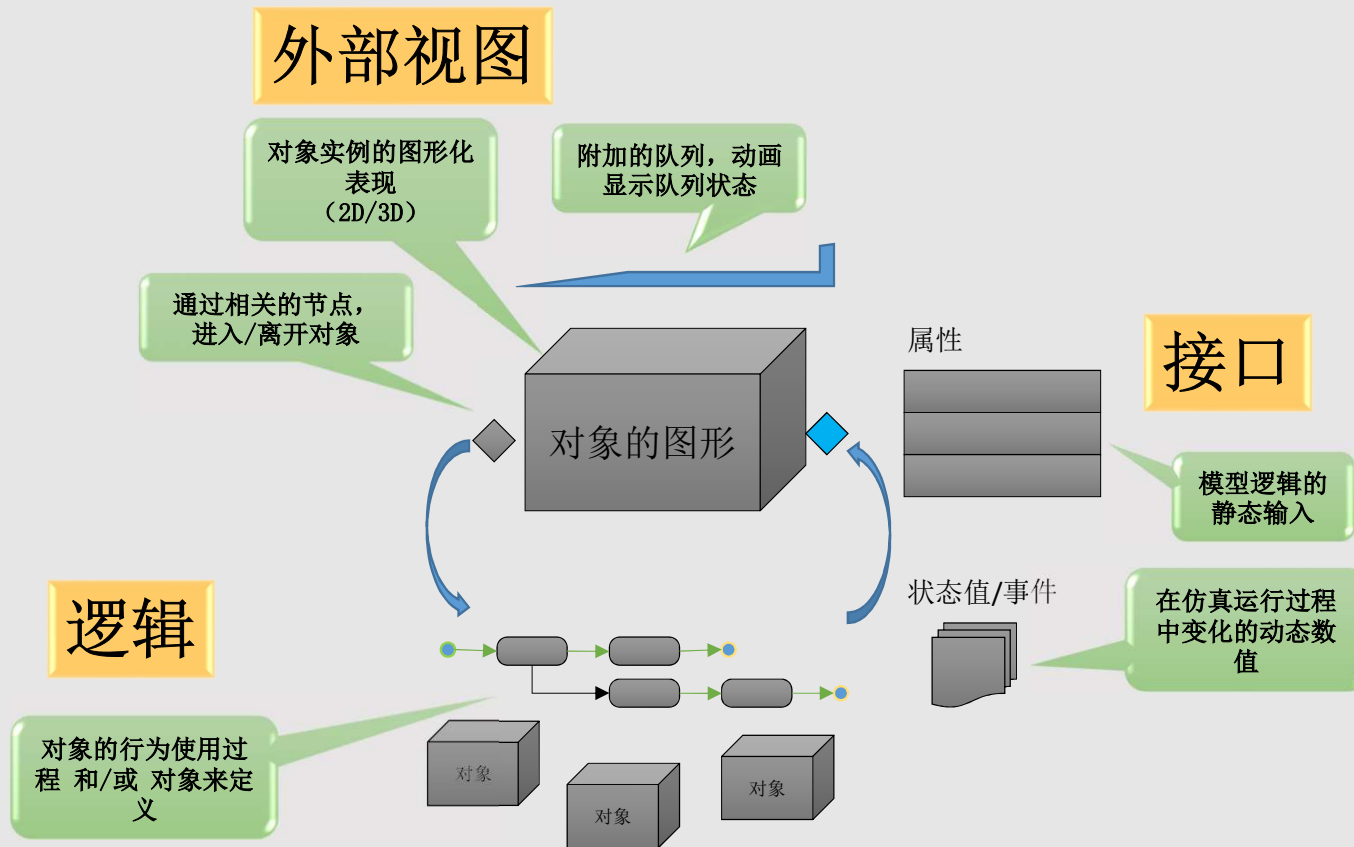
对象定义部分也可以有可选的数据面板，以显示动态变化的信息。

逻辑（对象和/或流程）

接口(属性, 状态, 事件)

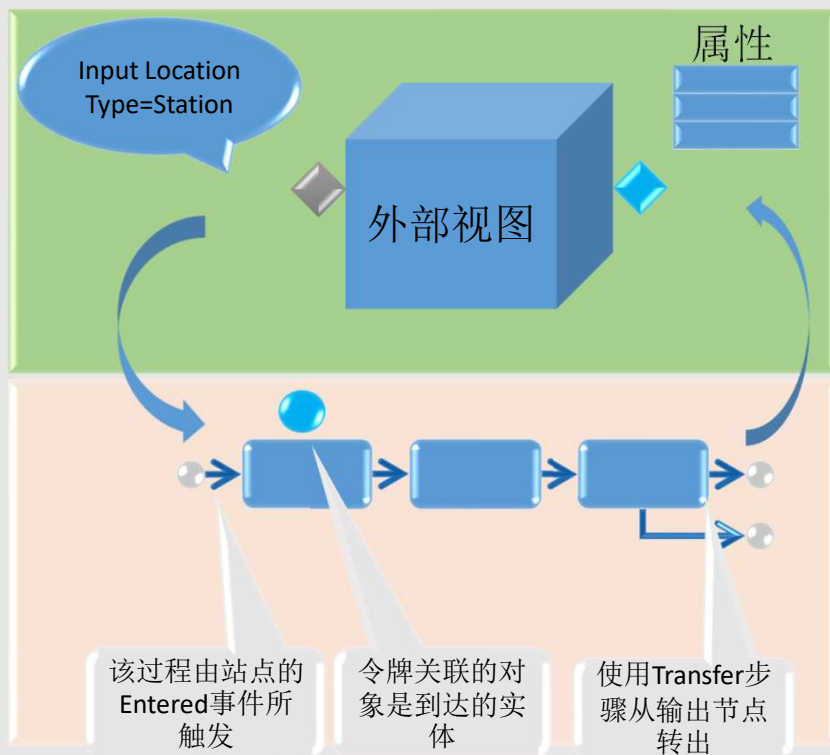
外部视图(图形, 入口/出口节点)

对象的剖析（回顾）

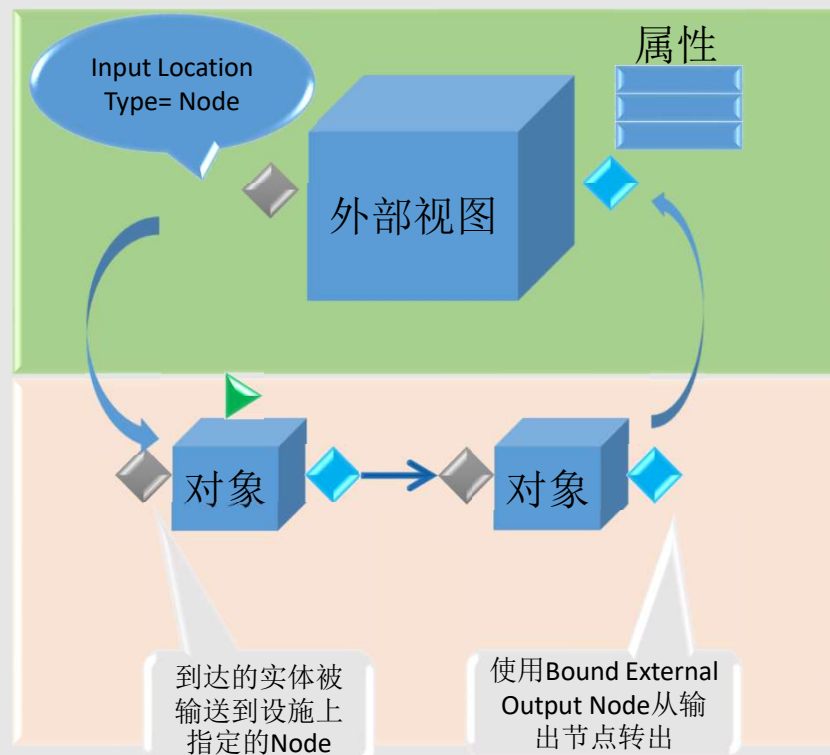


进入/输出节点

过程逻辑

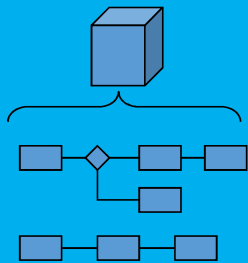


对象逻辑

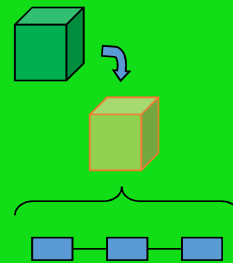


建立对象定义

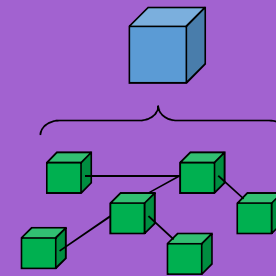
基础类
(使用过程步骤建立)



派生
(从其他对象子类化)



层级
(组合了对象/其他模型)



对象属性

- ▶ 属性是对象的输入。
- ▶ 对象无法知道任何外部的信息，可通过属性来输入外部信息。例如：
 - ◆ 一个表达式类型的属性：用户可以输入这个对象的加工时间。
 - ◆ 一个事件类型的属性：可以传递事件名称给对象，该事件发生时对象可以响应。
 - ◆ 一个对象类型的属性：可以传递外部的对象名字给此对象，使其知道模型中的其他对象。
- ▶ 在建立对象时，要考虑这个对象需要获得哪些外部信息。

Properties: Output@Server1 (TransferNode)

Crossing Logic	
Initial Capacity	Infinity
Ranking Rule	First In First Out
Routing Logic	
Outbound Link Rule	Shortest Path
Outbound Link Prefere...	Any
Entity Destination Type	Continue
Transport Logic	
Ride On Transporter	True
Transporter Type	Specific
Transporter Name	
Reservation Method	Reserve Closest
Selection Goal	Preferred Order
Selection Condition	
Add-On Process Triggers	
General	
Name	Output@Server1
Description	
Public	False
Report Statistics	False
Report Gantt	False
Auto Transfer On Entry	None

Ride On Transporter
Indicates whether entities entering this node object will need to select and ride on a transporter object for further travel.

属性被分配给某个属性组，属性组可以缩小或展开

属性可以使用一个开关来启用或关闭，依据于某个条件

显示名称 Display Name 可以有空格和特殊符

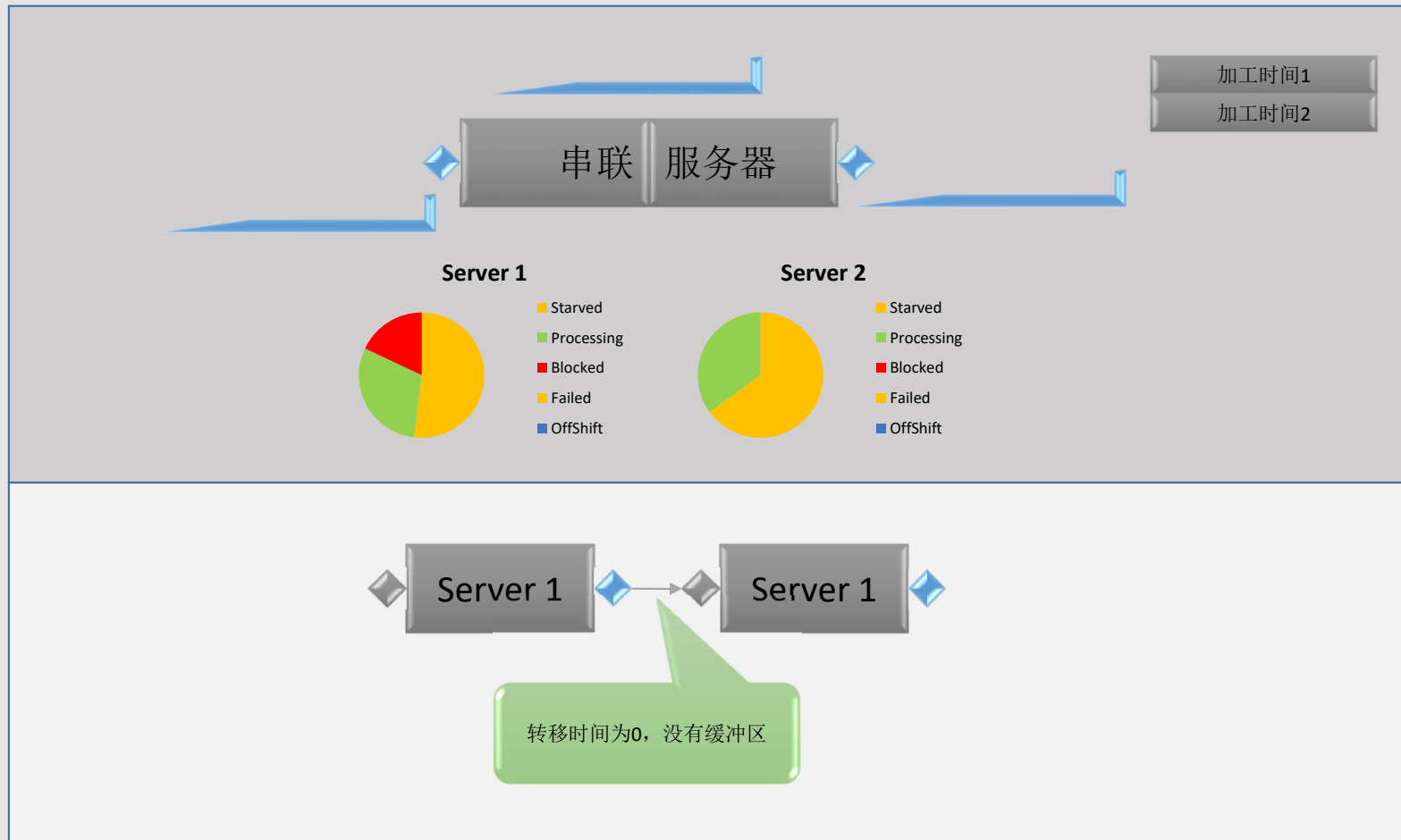
属性有个默认值 Default Value.

属性可能需要一个 Required Value.

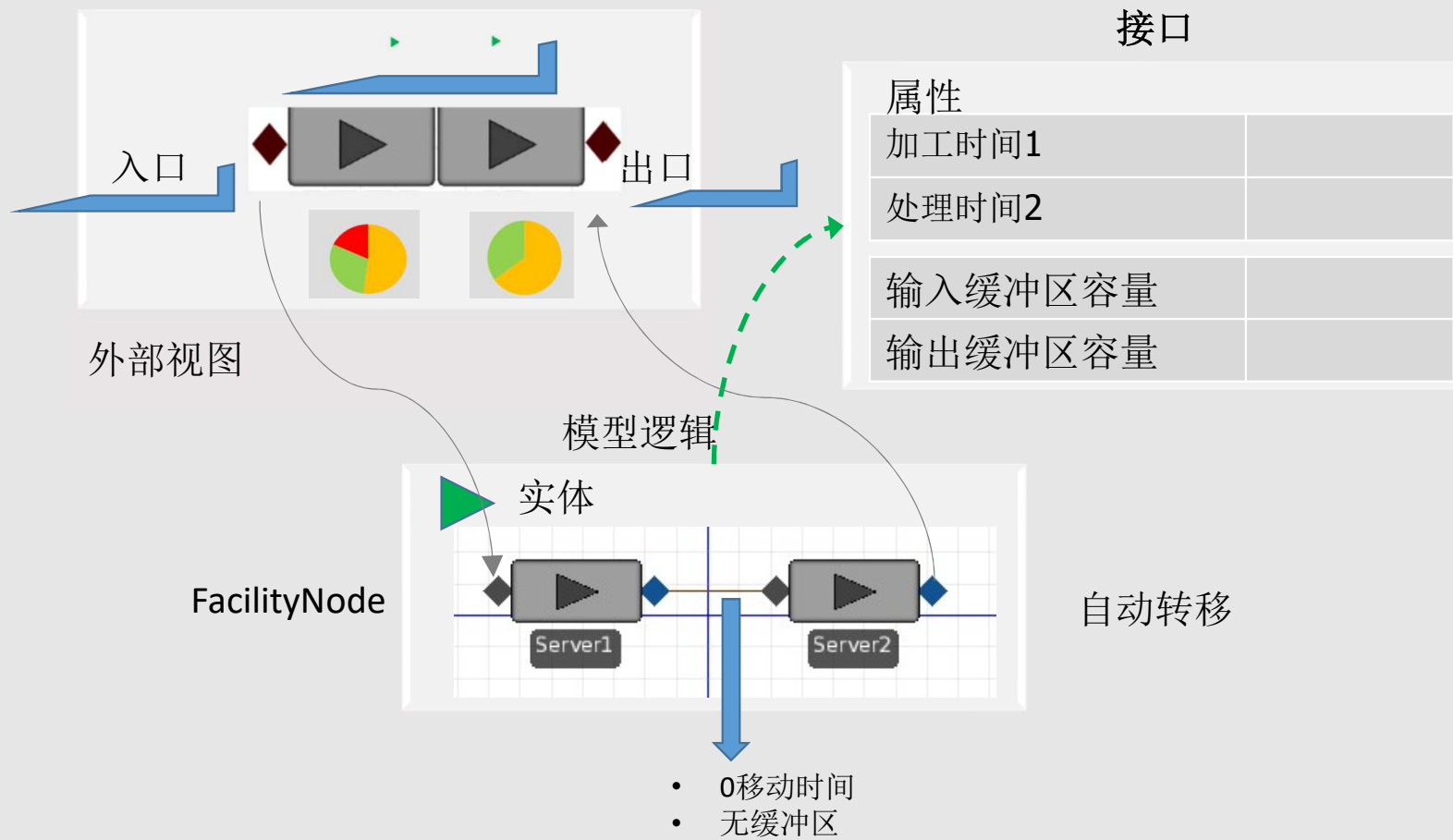
有些属性是不能编辑或者不可见的

当用户选中属性时，会显示属性的描述。

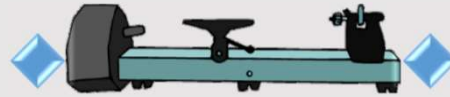
串联服务器对象



串联服务器对象定义



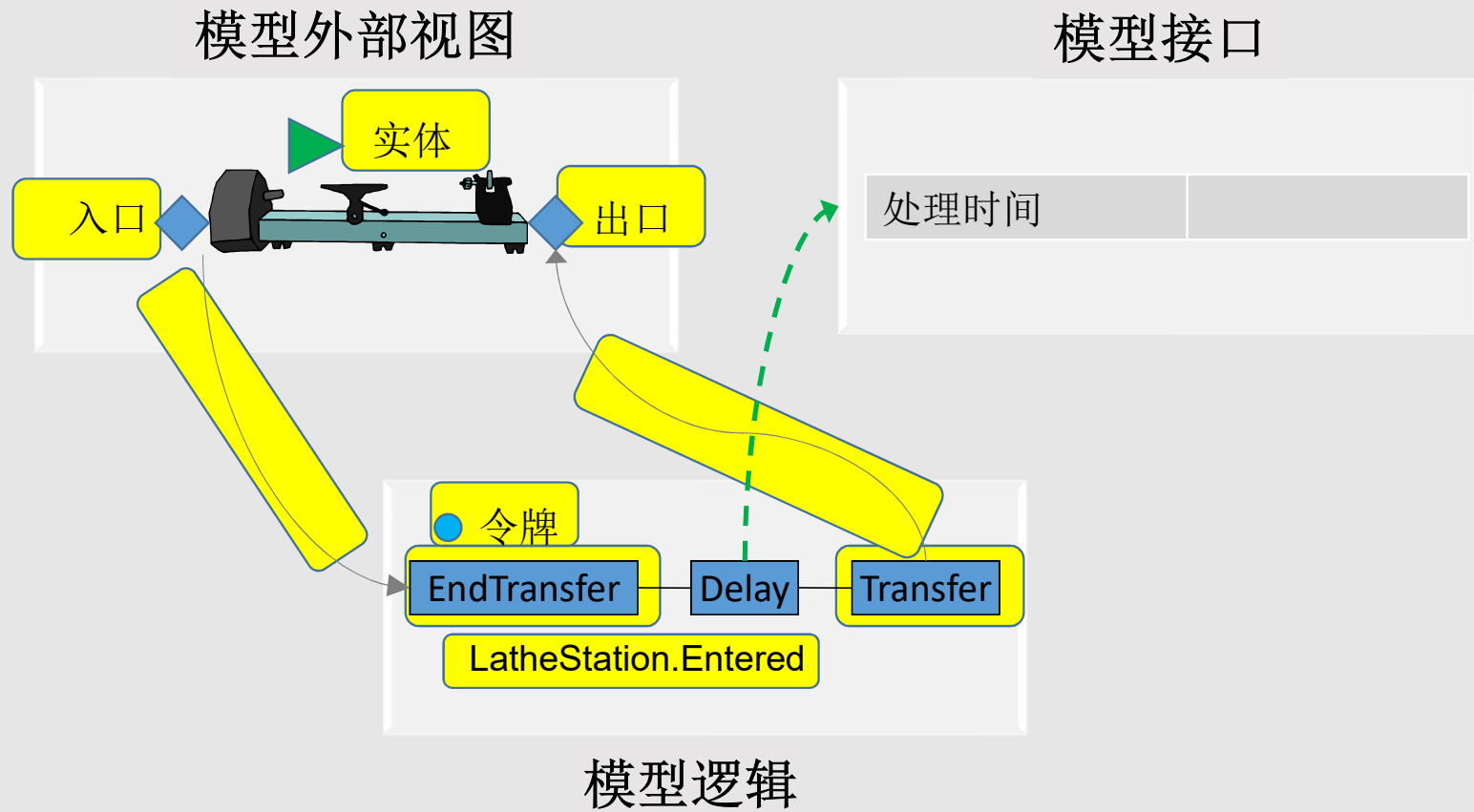
简单车床例子



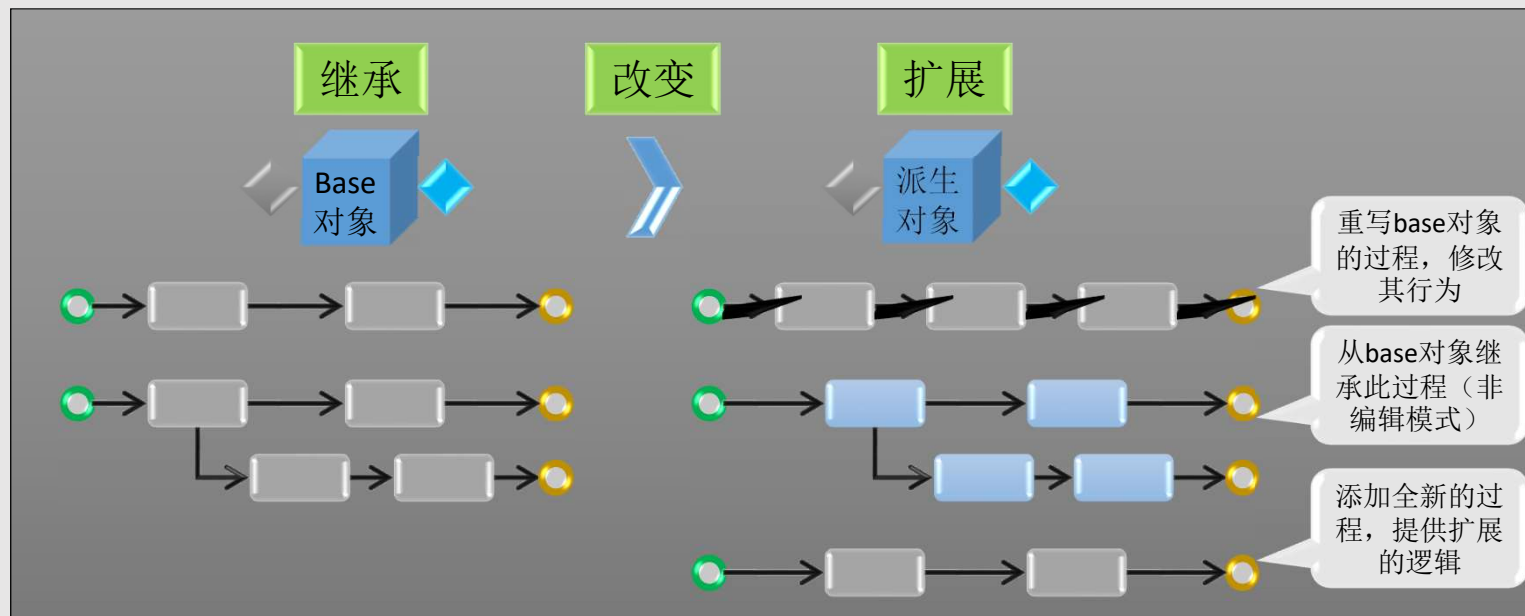
- 转移到站点内，停止
- 根据服务时间延迟一段时间
- 转移到外部输出节点

注意: 没有进入/输出缓冲区的空间

车床定义



子类化 (Subclassing)



子类化库对象

- ▶ 标准库对象是完全开放的。
- ▶ 拖动库对象到项目区,创建该对象的子类对象。
- ▶ 编辑子对象
 - ◆ 重写(Override)过程
 - ◆ 添加新过程
 - ◆ 给外部视图增加图形修饰
 - ◆ 隐藏/修改继承的属性
 - ◆ 增加新的属性

带有修理工的MRI设备



重载这个继承的过程，
增加一个Seize和
Release步骤

修改对象定义后会发生什么

- ▶ 绑定对象（Attached Objects）和属性的默认值仅仅当对象被实例化时才被使用，所以这些不会被更新。
- ▶ 实例化对象的其他内容均被更新。
- ▶ 绑定对象包括带有用户可编辑的表达式（折线图，标签 ...）的任何东西和任何可以被编辑和删除的东西（例如队列...）。
- ▶ 任何被放入Facility视图的对象是一个静态（不可编辑）的对象，而不是一个绑定对象，所以被自动更新。

第九章总结

- ▶ 任何模型都可以提供对象定义的逻辑-只需增加接口和外部视图。
- ▶ 对象定义可以通过三种途径实现：
 - ▶ 基础类对象使用流程。
 - ▶ 利用继承,派生出新的对象。
 - ▶ 利用组合功能建立层级对象。
- ▶ 包含对象定义的项目可以作为库文件打开。
- ▶ 库文件可以在企业内部共享。

研讨会-建立对象的三种方法

使用组合的建模方法创建三个串联服务器的对象，和课堂的例子一样。



使用基础对象类重新建立和课堂上讲的同样的模型。



从标准库中子类化服务器对象，增加一个修理资源作为属性，重载相应过程，使用该修理资源进行修理。



第十章



计划和调度简介



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

内容

- ▶ 介绍
 - ▶ Simio 产品家族
 - ▶ 什么是Simio RPS?
 - ▶ 为什么系统变化至关重要
 - ▶ 确定型的计划调度vs基于风险的计划调度
- ▶ 调度项目的方法
- ▶ 总结



Simio 产品家族

▶ Simio 设计版

- 使用过程来定制化对象和建立对象库。

▶ Simio 团队版

- 分享模型（免费运行时）&在网络内分布式运行。

▶ Simio 企业版

- 建立和执行计划/调度模型，在企业内部部署。
- 增加输出状态 (比如，船期) 到数据表中。
- 增加目标 (比如，松弛时间=客户要求交货日-发运日) 到数据表中。
- 定制化计划数据表的查看/编辑选项。
- 计划/调度的功能。

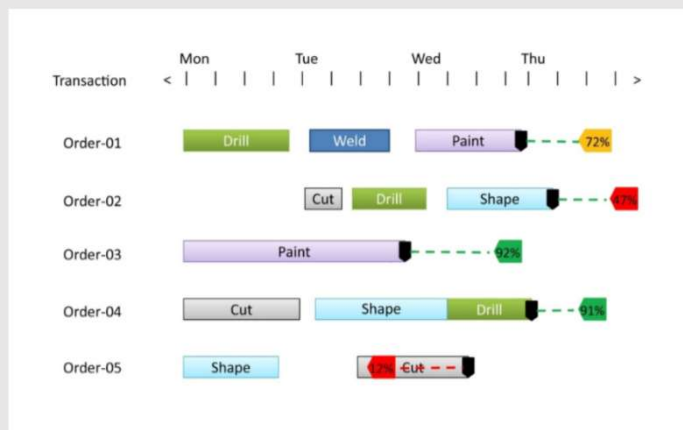
▶ Simio 门户版(SaaS)

- 将仿真结果分配给任何人，任何地方，任何终端。
- 上传模型到云端，在线执行和优化。

Simio基于风险的计划与调度 (RPS)

► 设计 & 分析

- 可视化和理解系统
- 分析系统的绩效
- 评估可行方案
- 实施风险的最小化

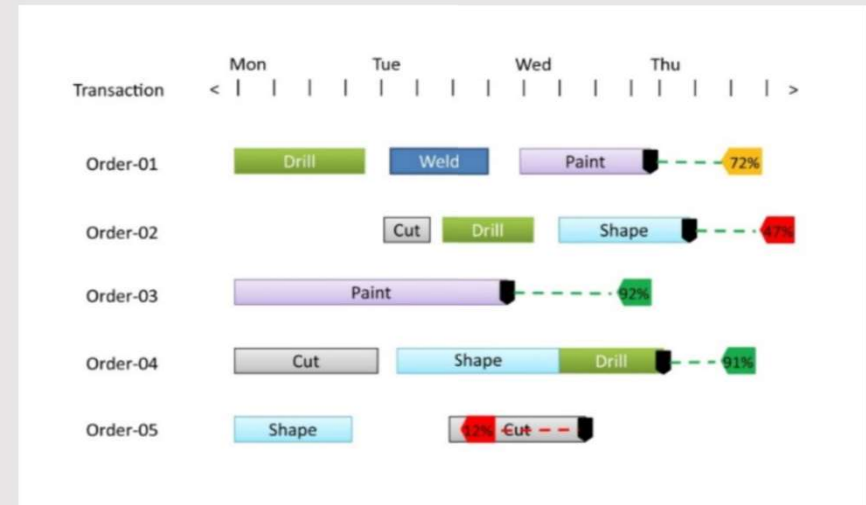


► 计划和调度

- 正确地分配资源，改进日常运营
- 在不确定的环境下准时和按预算交付产品/服务
- 增加自动化的风险分析，评价计划可行性

为什么使用Simio计划调度技术

- ▶ 允许Simio仿真模型能够提供用于计划和调度的逻辑上的设施模型。
- ▶ 添加日志，甘特图，报告，交互数据面板，观察和追踪订单在系统中的流动。
- ▶ 调度中使用用户自定义的“目标”(比如，发运日)。
- ▶ Simio计划模型加入了加工时间的波动和未计划事件的影响，用于评估确定型计划的实施风险。
 - ▶ Simio自动地复制模型，产生风险指标。
 - ▶ 达成目标的概率
 - ▶ 预期的完成时间
- ▶ Simio自动跟踪给计划调度带来非增值时间的约束条件。



确定型计划调度已经过时

- ▶ 采用固定架构的模型，只适用于单一用途且不灵活。
- ▶ 难以模拟动态对象。
 - ▶ 吊车,机器人,液体罐, 传送带
 - ▶ 更为高级的设备
- ▶ 难以表现具体的流程细节和决策规则。
- ▶ 确定型计划无法精确地模拟工厂约束条件和变动。

为何确定型计划失效？

▶ 无法精确地捕捉系统的复杂性

- ▶ 物料处理
- ▶ 专业化的作业 / 资源分配
- ▶ 基于经验的决策逻辑



▶ 无法有效地解决系统变化性

- ▶ 设备故障 / 计划外事件
- ▶ 工人 / 任务时间 / 物料可用性



Simio不仅能解决这些问题，
还有更多功能！

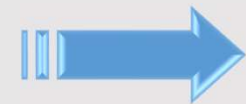
为什么系统波动至关重要

设计2小时的数码冲印店

到达间隔平均
为60分钟



确保“2小时”
相片处理



平均服务时间是
55分钟

系统的表现如何?
24x7模式下顾客的等待时间?



到达间隔

常量
随机*
随机*

服务时间

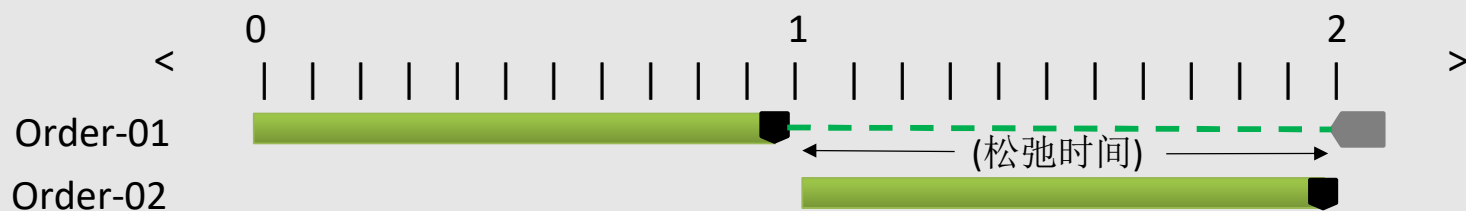
常量
常量
随机*

等待时间?

我们来模拟
一下...

* 指数分布

为什么系统波动至关重要



▶ 计划是基于“期望”的确定性数值（平均60分钟到达和平均55分钟服务时间)

▶ 92% 利用率

▶ 静态分析显示:

没有未准时完成的订单



▶ 动态分析显示:

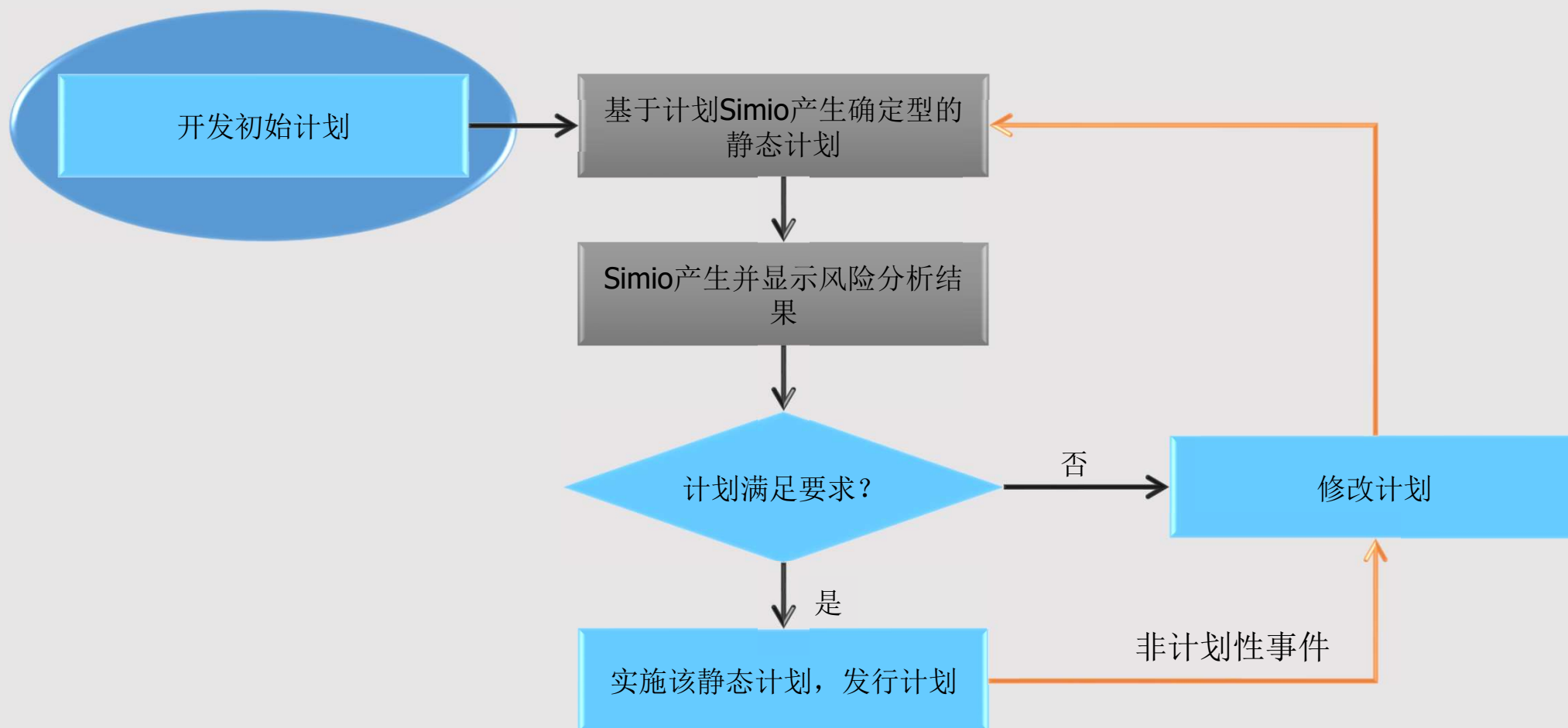
84%的订单交付都延误了!

变化性对系统的影响巨大!

Simio独特的解决方案

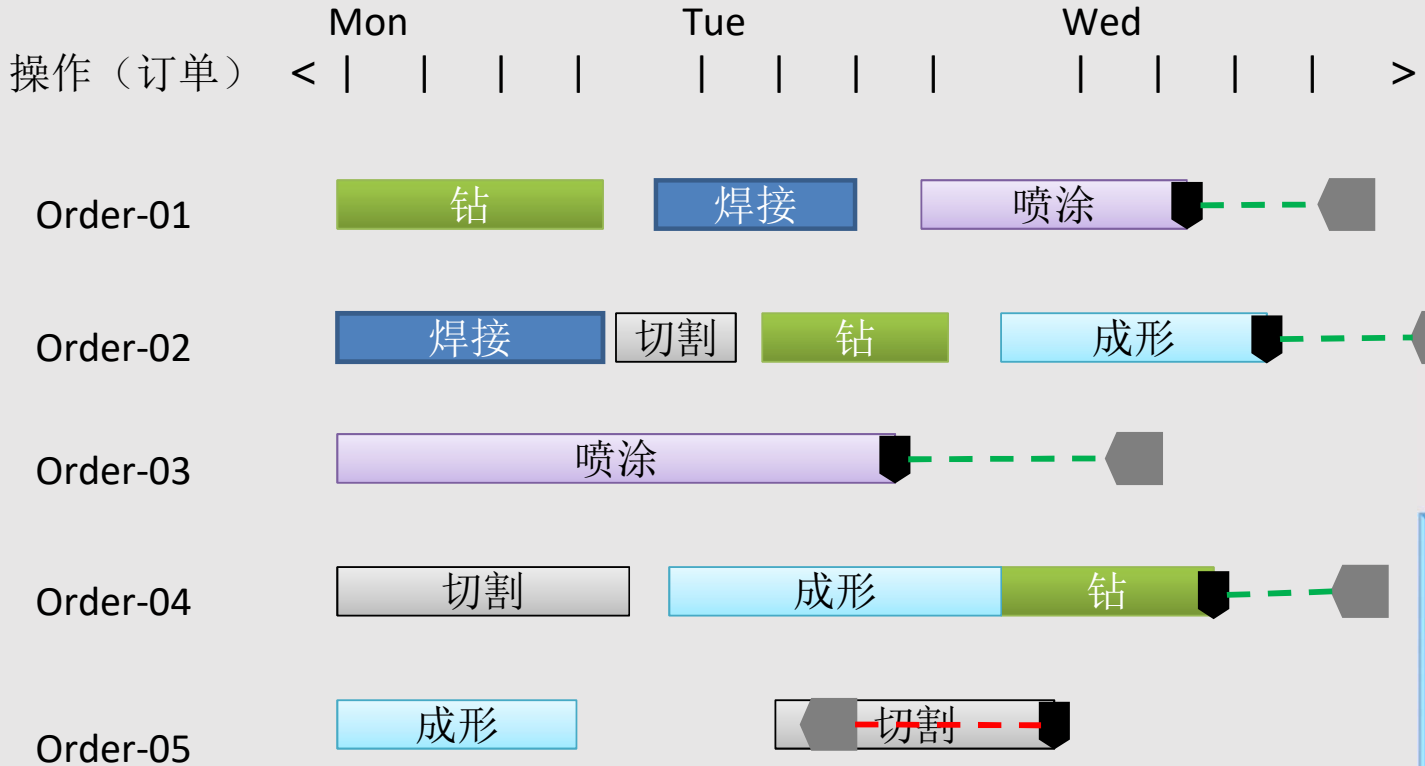
- ▶ **Simio**基于风险的计划和调度 (RPS).
 - ▶ 基于仿真内核的计划和调度方法
 - ▶ 高度灵活的模拟，可配置
 - ▶ 捕捉风险和不确定性
- ▶ 模型用于：
 - ▶ 首先产生细节化的确定性计划和调度方案。
 - ▶ 其次，根据系统中所给的内在的变化性，评估相关的风险。
- ▶ 很有信心地管控复杂，动态和高度可变的环境。
- ▶ 使用**3D**动画沟通结果。

计划和风险分析 workflow



基于风险的计划调度实例

➔ 确定性计划



如何保证该订单有90%的概率能够准时交付?

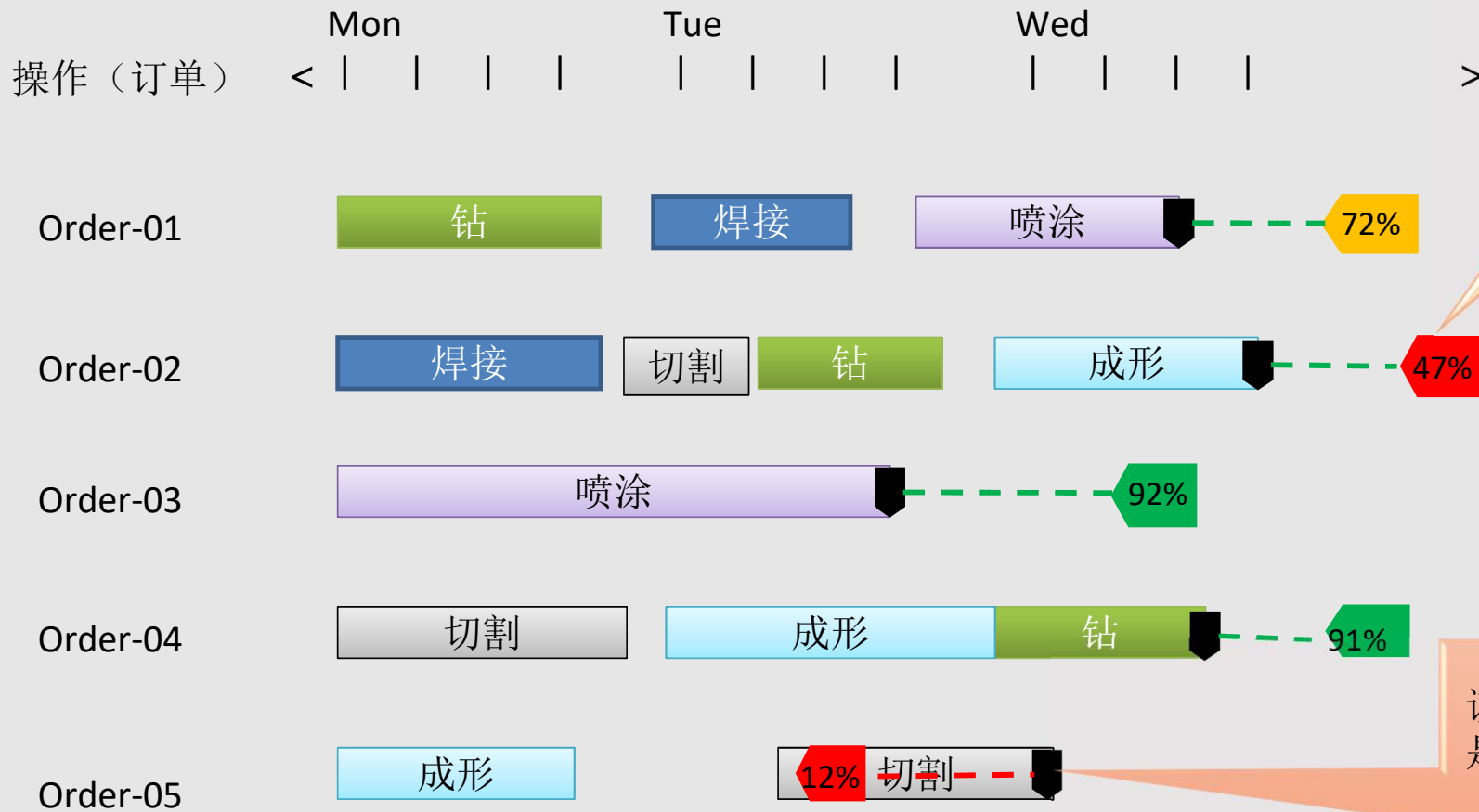
计划方案随着时间而变得不可行, 当机 - 物料延迟 - 人员配备 ...

Simio风险和约束分析

- ▶ 自动产生风险和约束的衡量指标。
- ▶ 用户自定义一个或多个“目标”，比如发运日。
- ▶ Simio增加加工时间的变动，引入非计划性事件的影响等。
- ▶ Simio自动运行模型的多个副本（多处理器并行),按照特殊的算法产生风险指标值。
 - ▶ 达成自定义目标的概率.
 - ▶ 期望的完工时间.
 - ▶ 悲观的完工时间 (比如 80分位).
 - ▶ 乐观的完工时间 (比如 20分位).
- ▶ Simio自动追踪系统内的约束，这些约束往往给计划执行带来大量不增值时间（NVA）。

给计划增加风险指标分析

➔ 增加变化性和风险分析



该订单还有松弛时间，但是有比较高的风险会拖延！

该订单显示在计划上是按时的，但有很可能会延迟。

调度项目实施方法

调度项目的实施可以采用下面基本方法之一

模型驱动方法

数据驱动方法

整个设施还在设计和建筑阶段。

设施已经存在，但是MES还在实施前期。

设施已经存在，但是没有MES系统。

设施已经存在，MES已经安装和正常运作。

设施已经存在，但是MES还在实施阶段或未来阶段。

设施已经存在，但是Simio需要和ERP/MRP系统整合。

Simio调度数据表

▶ Simio插件创建符合 B2MML*规范的数据表:

- 资源表 (Resources table)
- 制造订单表 (Manufacturing Orders table)
- 采购物料表 (Purchased Materials table)
- 物料表 (Material table)
- 工艺线路表 (Routings table)
- 物料清单表 (Bill Of Material table)
- 在制品表 (Work In Process table)
- 制造订单输出表 (Manufacturing Orders Output table)

**(B2MML=Business To Manufacturing Markup Language)*

Simio数据表实例

► 资源

- 加工步骤需要的固定资源的列表

字段名	描述
Resource Name	资源的唯一名称
Description	资源的描述
Update Location	布尔型变量，定义该资源是否被更新到此位置
X Location	沿着X轴多少米的位置
Z Location	沿着Z轴多少米的位置
Parent Resource Name	机器组的上级资源的名称
Object Type	对象类型指定为WorkStation或者TransferNode
Node List	父节点指定和机器组关联的节点列表

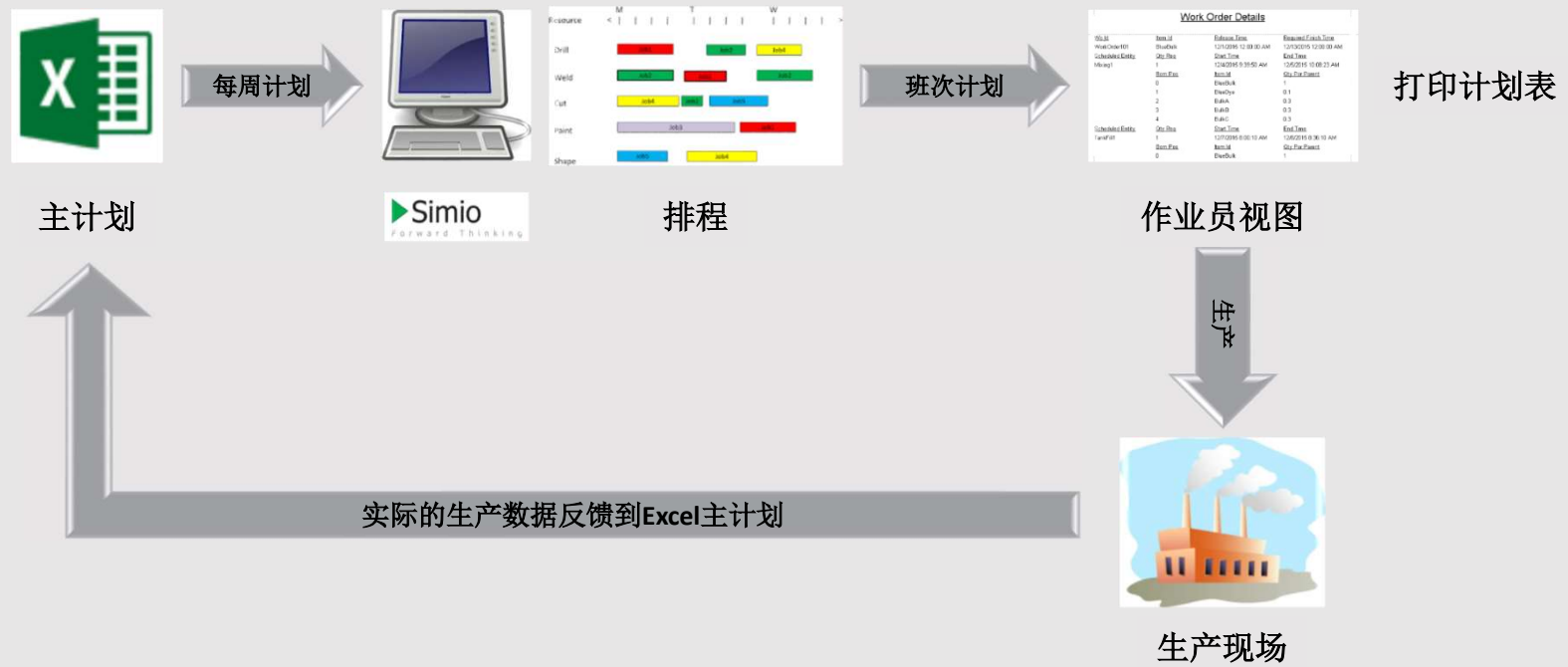
► 工艺线路

- 为每个物料指定资源，为每个任务指定服务时间的工艺线路清单

字段名	描述
Routing Key	工艺步骤的唯一名称
Material Name	对于生产物料的材料表进行的外键应用
Route Number	工艺步骤编号
Sequence	该步骤被执行的资源序列位置
Process Time	该步骤的加工时间
Setup Time	在该过程开始前必须的准备时间

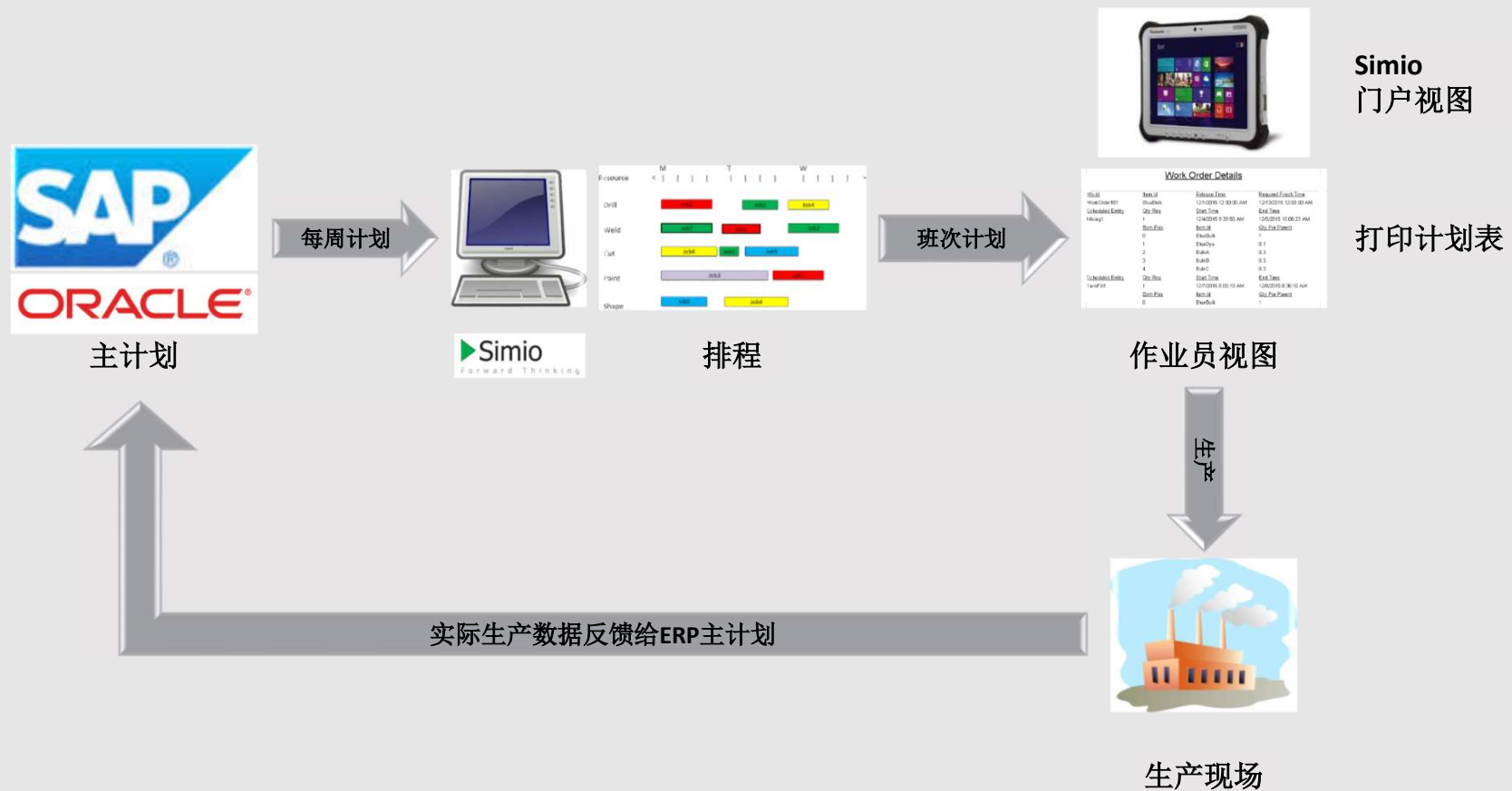
调度项目实施方法

独立模式（基础模式）



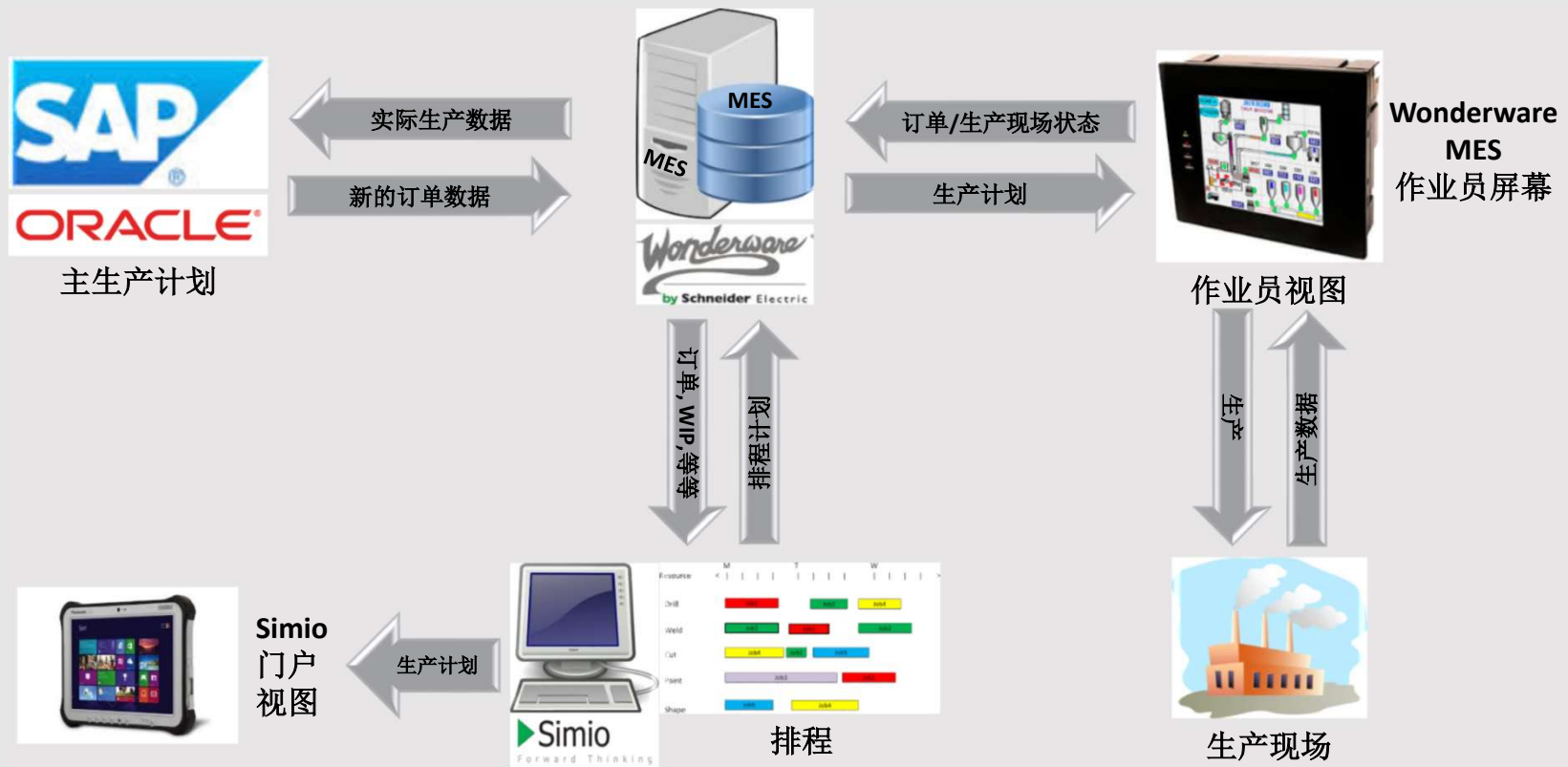
调度项目实施方法

ERP整合模式 (基础整合)



调度项目实施方法

ERP & MES 整合的模式 (最终模式)



Simio门户版本

- ▶ **Simio门户版本是SaaS模式的(软件即服务)**

- ▶ 微软Azure云平台
- ▶ 高度可扩展，可靠和安全

- ▶ **增强了结果的可视性**

- ▶ 在web上发布和分享仿真和调度的结果
- ▶ 可供无限数量的用户观看模型结果

- ▶ **增加了仿真实验的入口**

- ▶ 建模者允许其他的团队成员实施仿真实验，而不需要Simio桌面软件

- ▶ **仿真实验和调度进行整合**

- ▶ 可以运行仿真实验，也可以运行所有方案或某些方案的计划。

- ▶ **Simio门户版本的接入只需要标准的网页浏览器或触摸屏的移动终端设备，比如智能手机和平板。**



第十章总结

- ▶ **Simio RPS** 基于风险的计划调度的功能是独特的。
- ▶ 一个三维模型同时用于设计和运营用途。
- ▶ 复杂环境的强大解决方案。
- ▶ 可以和**ERP/MES**系统整合。
- ▶ 独特的价值主张:
 - ▶ 系统设计和分析支持
 - ▶ 计划调度和运营支持
- ▶ 可以用数据驱动或模型驱动的方式实施。
- ▶ 开发的架构和**API**支持定制化的**.NET** 扩展。
- ▶ 通过网络浏览器和移动设备在企业各个部门间发布结果。
- ▶ 绑定到关系型数据库，例如**SQL Server, Access**和**Excel**。



第十一章



输入数据分析



(0510) 82132584



无锡迅合信息科技有限公司©



<http://www.simio-china.com/>

分析输入数据

处理不同的数据种类

拟合分布

常用分布函数的总结

模拟客户到达

模拟任务时间

结果的敏感度

系统名称	典型的输入数据
排队系统	顾客到达的间隔时间
	顾客被服务的时间
库存系统	需求的分布（按客户）
	顾客订购量的分布
	物料订货提前期的分布
生产系统	作业到达的时间间隔
	作业类型的概率
	每种作业每道工序服务时间分布
	设备当机的频率和修复时间 (或MTBF, MTTR)

模型输入数据

模型具有结构和输入数据。

模型结构和输入数据对仿真结果都有重要影响。

数据可能是仿真模拟项目中问题多发区。

典型的数据实例

不存在数据。

大量好的数据。

错误形式的数据。

不存在数据

关于活动时间可以考虑使用三角分布或者Pert分布 (最小值, 中间值, 最大值)。

根据内在的流程, 选择假设分布, 合理地估计参数。

不要试图用平均值来代替分布函数。

评估结果对于参数的灵敏度。

没有数据：响应敏度分析

确定哪些数据是关键。

计算每个响应变量对于模型输入参数的敏度。

预估的数据参数是有用的。

显示哪些输入参数对响应变量有最大的影响。

在仿真项目早期被使用，帮助引导决策，投入时间和经费在初期就搜集真实的观察数据，用于输入参数。

错误形式的的数据

- ▶ 数据从不同的现实世界的流程中观察。
 - TBF/故障是计数的
 - TTR/修理是资源约束的
 - 在“空闲”和“忙碌”时间记录的数据
- ▶ 从多个流程中获得的数值没有区别性的信息（比如：修理时间，但是没有告知停机类型）。
- ▶ 用已经存在的数据来人为地猜测必要的的数据。

存在大量数据-三种方法

使用输入参数，从表格中的数据直接取样

使用经验分布

使用理论分布

- 简洁, 快速, 容易修改
- 可根据内在的流程假设一个分布
- 使用拟合软件来测试假设检验和评估参数

数据拟合程序

前提：评价IID假设.

- 样本间独立
- 同分布

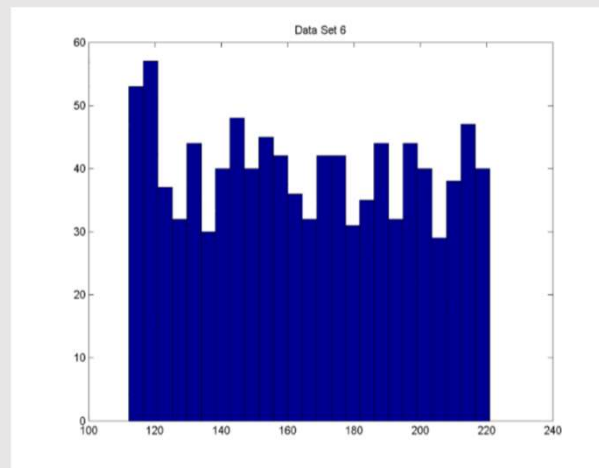
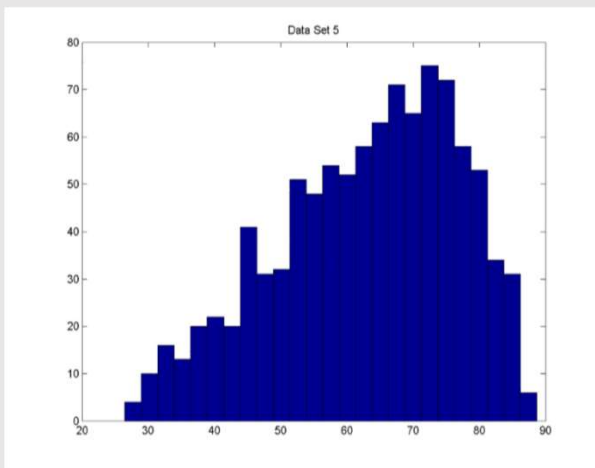
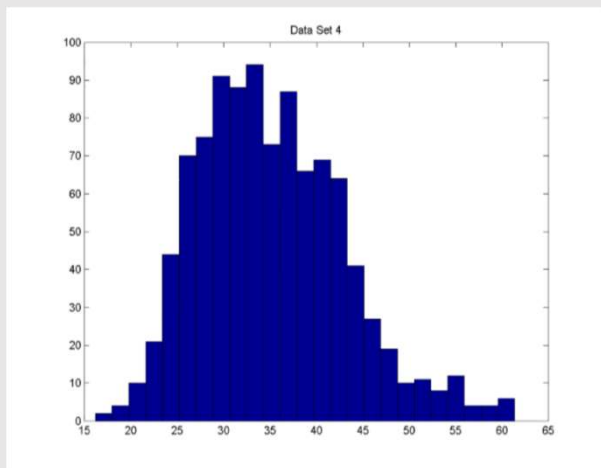
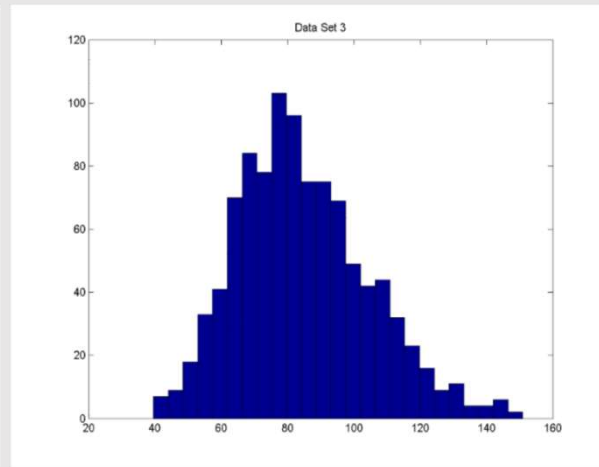
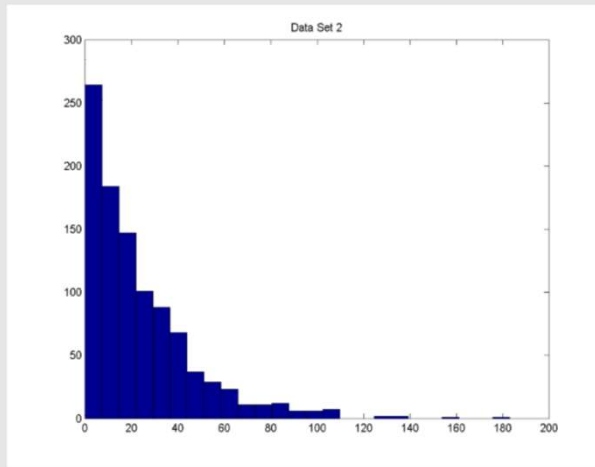
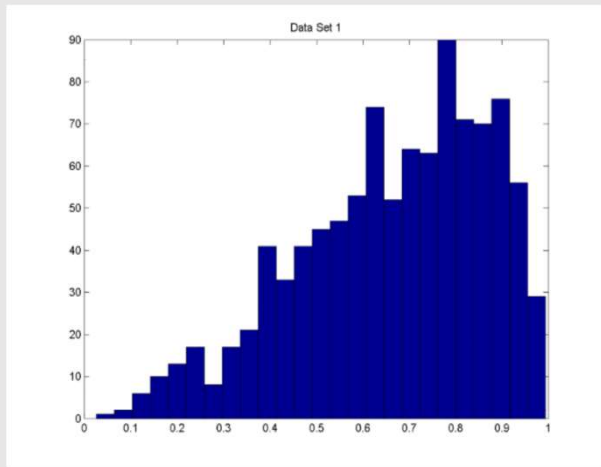
使用软件，用柱状图来观察数据

假设一个分布族/分布形式

使用专业拟合软件来:

- 评估分布的参数
- 评估拟合质量

样本数据集



常见分布函数

二项
分布

泊松
分布

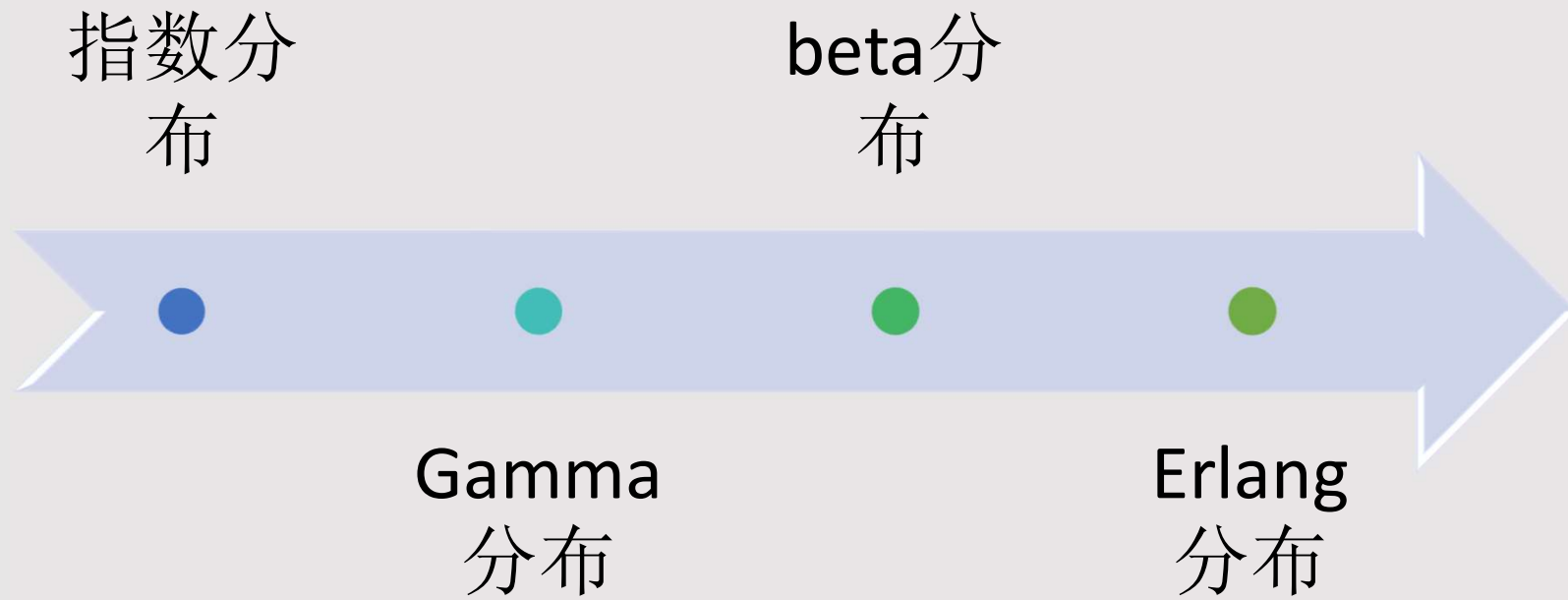
对数正
态分布



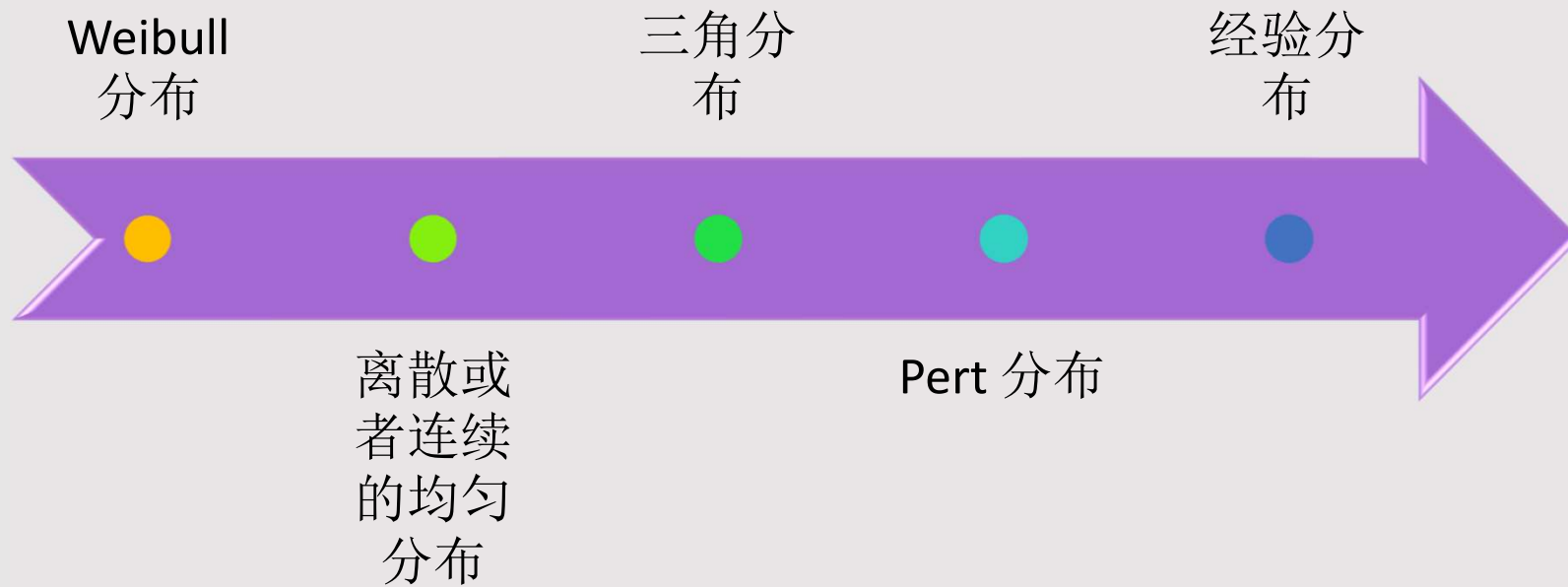
负二
项分
布

正态
分布

常见分布函数



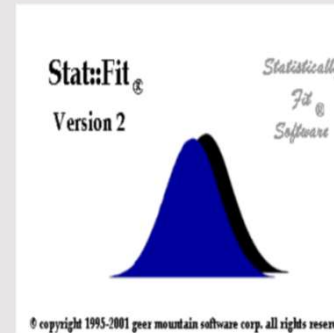
常见分布函数



拟合软件选项

▶ 仿真软件特定的

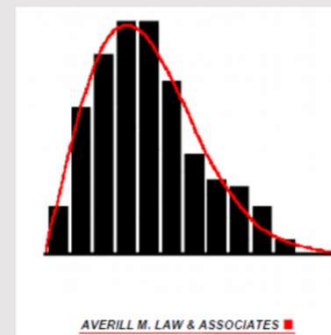
- Stat::Fit (www.geerms.com)*
- ExpertFit (www.averill-law.com)



以Stat::Fit®为例

▶ 通用软件包

- R (<http://www.r-project.org/>)
- EasyFit (www.mathwave.com)
- BestFit (<http://www.palisade.com/>)



注意可能会有不匹配的参数!

* 教科书配套软件在教材的网站上下载。

模拟到达

如果到达是独立和随机的，那么它们服从泊松分布。

- 在固定的时间段内到达的数量服从泊松分布
- 到达间隔是服从指数分布

在有些情况下，到达率会随着时间而变化-Simio支持分段的线性到达率（使用速率表）。

模拟任务时间

使用带范围的分布（范围 ≥ 0 ），例如不会是正态分布或者JohnsonUB分布。

数据缺乏的时候，三角分布和Pert分布是可选分布。

具有数据支持的情况下，Gamma，LogNormal，Weibull，LogLogistic，Beta，PearsonIV，JohnsonSB分布都是可以选择的。

Gamma, LogNormal, Weibull

gamma

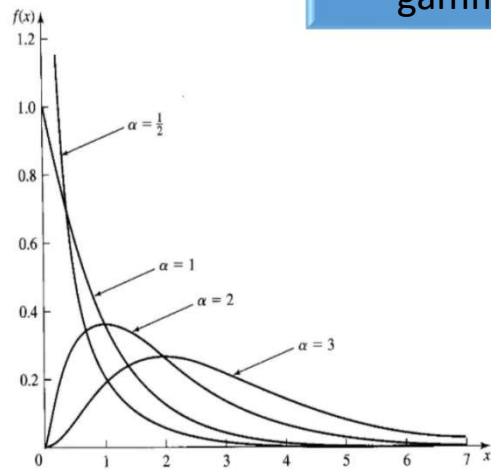


FIGURE 6.7
gamma($\alpha, 1$) density functions.

Log Normal

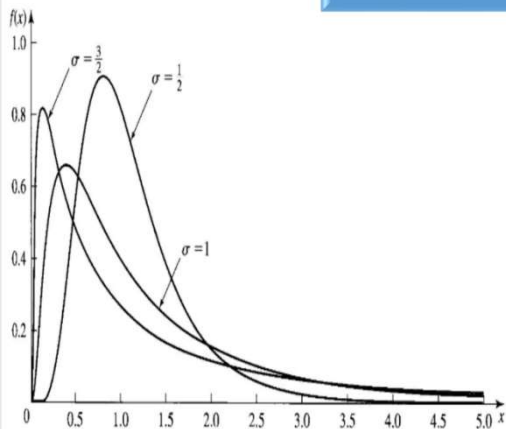


FIGURE 6.10
LN($0, \sigma^2$) density functions.

Weibull

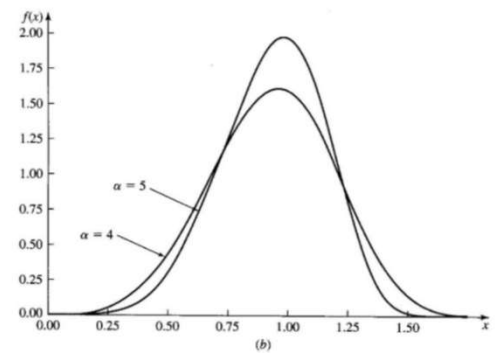
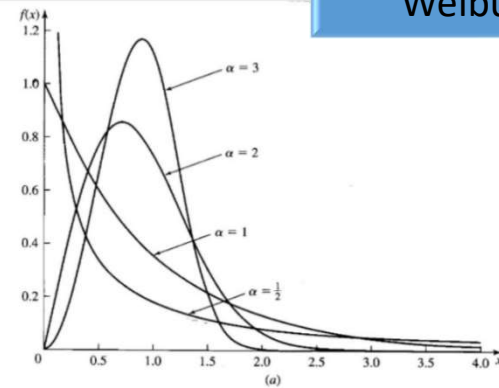


FIGURE 6.8
Weibull($\alpha, 1$) density functions.

第十一章总结

- ▶ 分布函数是获取系统中变化性的主要方法。
- ▶ 不要试图用“平均值”来代替随机分布。
- ▶ 当数据存在时，假设一个分布，估计参数，并使用拟合软件来测试符合性。
- ▶ 如果缺乏数据，使用合适的分布函数。
 - 到达 – 到达间隔是指数分布，或者非静态泊松
 - 活动 – 三角分布或**Pert**分布
 - 设备失效 – **Erlang**分布或**Weibull**分布
 - 使用响应敏度分析来评估
- ▶ 使用模型确定最关键的数据元素。

培训师介绍



Zhou Zheng 周峥



高级咨询师(系统仿真建模)

M: +86 137-7150-8461

E: jeffzz111@163.com

微信: jeffzz111wx



扫一扫加我微信

我的仿真工具箱

SIMIO

SIMUL8

ExtendSIM

(1976-) 江苏无锡人。上海交通大学毕业，获硕士学位。系统仿真高级咨询师，无锡迅合信息科技有限公司联合创始人。研究领域：工业工程和物流工程领域系统仿真，项目组织仿真设计。

对上述领域有持久、深入的研究和实际工作经验。系统仿真项目经验10+年。曾在多个世界500强企业担任中高级管理职位，从业经历20+年。在精益制造系统，物流系统，供应链系统方面均有丰富项目经验，行业涵盖消费电子，机械制造，汽车制造，制药，物流，咨询教育等行业。熟悉仿真软件

Simio, Simul8, ExtendSIM开发，熟悉开发语言C#, 数据库SQL Server，大数据挖掘(jmp, spss, Tableau®等)。

TRAINING - SIMULATION

PROVIDE YOUR
TEAM WITH THE
TOOLS TO
SUCCEED

Simio



THANKS
感谢在座各位!