

WITNESS 工业物流仿真平台基础教程

LANNER 北京代表处

北京威特尼斯科技中心



目 录

<u>第一章 离散事件系统仿真简介</u>2

| <u>1.1 仿真技术的产生与发展</u> | 9 |
|----------------------------|----|
| <u>1.1.1 仿真软件的发展</u> | 10 |
| <u>1.1.2 仿真建模方法学的发展</u> | 12 |
| <u>1.2 系统、模型与仿真</u> | 13 |
| <u>1.2.1 系统</u> | 13 |
| <u>1.2.2 模型</u> | 14 |
| <u>1.2.3 仿真</u> | 14 |
| <u>1.3 离散事件系统仿真</u> | 16 |
| <u>1.3.1 基本概念</u> | 16 |
| <u>1.3.2 离散事件系统仿真方法</u> | |
| <u>1.3.3 仿真策略</u> | 19 |
| <u>1.4 系统仿真的一般步骤</u> | 19 |
| <u>1.4.1 问题的定义</u> | 20 |
| <u>1.4.2 制定目标和定义系统效能测度</u> | 21 |
| <u>1.4.3 描述系统和列出假设</u> | 21 |
| <u>1.4.4 列举可能的替代方案</u> | 22 |
| <u>1.4.5 收集数据和信息</u> | 22 |
| <u>1.4.6 构造计算机模型</u> | 22 |
| <u>1.4.7 验证和确认模型</u> | 22 |
| <u>1.4.8 运行可替代实验</u> | 24 |
| <u>1.4.9 输出分析</u> | 24 |
| | |

<u>第二章 随机分布</u>25

| <u>2.1 概率统计基本概念</u> | 25 |
|------------------------|----|
| 2.1.1 确定事件和随机事件 | 25 |
| <u>2.1.2 随机变量与概率</u> | 25 |
| 2.2 离散事件系统仿真中常见的概率分布 | |
| <u>2.2.1 排队系统</u> | |
| <u>2.2.2 库存系统</u> | 27 |
| <u>2.2.3 可靠性与维修性</u> | 27 |
| <u>2.3 随机数和随机变量的产生</u> | |
| <u>2.3.1 伪随机数</u> | 29 |



| 2 | <u>3.2 伪随机数产生方法</u> | . 30 |
|-----|--|------|
| 2.4 | WITNESS 系统标准随机分布函数 | . 34 |
| 2 | .4.1 伪随机数流 (PRNS: pseudo-random number stream) | . 34 |
| 2 | .4.2 随机分布函数详解 | . 35 |

第三章 WITNESS 仿真系统软件综述 46

| <u>3.1 WITNESS 的安装与启动</u> | 46 |
|---|-------|
| <u>3.1.1 安装环境</u> | |
| <u>3.1.2 安装步骤</u> | 46 |
| <u>3.1.3 启动 WITNESS 2003</u> | 48 |
| <u>3.1.4 退出系统</u> | 定义书签。 |
| <u>3.2 WITNESS 2003 用户界面</u> | 49 |
| <u>3.2.1 标题栏</u> | 49 |
| <u>3.2.2 菜单栏</u> | 50 |
| <u>3.2.3 工具栏</u> | 50 |
| <u>3.2.4 元素选择窗口</u> | 51 |
| <u>3.2.5 状态栏</u> | 51 |
| <u>3.2.6 用户元素窗口 (Designer Elements)</u> | 51 |
| <u>3.2.7 系统布局区</u> | 51 |
| <u>3.3 WITNESS 建模元素</u> | 52 |
| <u>3.3.1 离散型元素</u> | 52 |
| | 53 |
| <u>3.3.3 运输逻辑型元素</u> | 53 |
| | 53 |
| | 53 |
| | 53 |
| | 54 |

<u>第四章 元素——模型的组成部分</u> 57

| 4.1 | 离散型元素 | |
|-----|---------------------|----|
| 4 | 4.1.1 零部件 (Part) | |
| 4 | 4.1.2 机器(Machine) | |
| 4 | 4.1.3 输送链(Conveyor) | |
| 4 | 4.1.4 缓冲区(Buffer) | |
| 4 | 4.1.5 车辆(Vehicle) | 59 |
| 4 | | 59 |



| <u>4.1.7 劳动者(Labor)</u> | 59 |
|--------------------------------|----|
| <u>4.1.8 路径(Path)</u> | 60 |
| <u>4.1.9 模块(Module)</u> | 60 |
| 4.2 连续型元素 | 60 |
| <u>4.3 运输逻辑型元素</u> | 61 |
| <u>4.3.1 运输网络(Network)</u> | 61 |
| <u>4.3.2 单件运输小车 (Carriers)</u> | 61 |
| <u>4.3.3 路线集 (Section)</u> | 62 |
| <u>4.3.4 工作站(Station)</u> | 62 |
| <u>4.4 逻辑元素</u> | 62 |
| <u>4.4.1 属性(Attribute)</u> | 63 |
| <u>4.4.2 变量 (Variable)</u> | 63 |
| <u>4.4.3 分布 (Distribution)</u> | 64 |
| <u>4.4.4 函数 (Function)</u> | 64 |
| <u>4.4.5 文件(File)</u> | 65 |
| 4.4.6 零部件文件 (Part file) | 65 |
| <u>4.4.7 班次 (Shift)</u> | 65 |
| <u>4.5 图形元素</u> | 66 |
| 4.5.1 时间序列图 (Timeseries) | 66 |
| 4.5.2 饼状图 (Pie chart) | 66 |
| <u>4.5.3 直方图 (Histogram)</u> | 66 |

<u>第五章 规则</u> 68

| 5.1 | 输入规则(INPUT RULE) | |
|-----|--------------------|----|
| 5.2 | 输出规则(OUTPUT RULES) | |
| 5.3 | 劳动者规则(LABOR RULES) | |
| 5 | .3.1 劳动者规则概述 | |
| 5 | .3.2 三种劳动者规则 | |
| 5 | | 74 |

<u>第六章 WITNESS 程序设计基础</u> 76

| <u>6.1 变量类型</u> | |
|-----------------------|--|
| | |
| | |
| <u>6.1.3 名型(name)</u> | |
| | |



| 6 | . 2 | 运算 | 算符及表i | 达式 | |
|---|-----|---------|--------------|---------------|--|
| | 6 | . 2 . 1 | 算术运 | <u>算符</u> | |
| | 6 | . 2 . 2 | 2 关系运 | <u>算符</u> | |
| | 6 | . 2 . 3 | 3 逻辑运 | <u>算符</u> | |
| | 6 | . 2 . 4 | 转换运 | <u>算符</u> | |
| 6 | . 3 | 程度 | <u> 三种基</u> | <u>本结构</u> . | |
| | 6 | . 3 . 1 | 顺序结 | <u>构</u> | |
| | 6 | . 3 . 2 | 2 分支结 | <u>构</u> | |
| | 6 | . 3 . 3 | 3 循环结 | <u>构</u> | |

<u>第七章 可视化仿真项目的设计及运行示例</u>82

| <u>7.1 流水线仿真系统</u> | 82 |
|-----------------------------------|-----|
| <u>7.1.1 引言 (Introduction)</u> | 82 |
| <u>7.1.2 模型概述</u> | 82 |
| <u>7.1.3 构建第一阶段(Stage1.mod)模型</u> | 83 |
| <u>7.1.4 构建第二阶段(Stage2)模型</u> | 89 |
| <u>7.1.5 构建第三阶段(Stage3)模型</u> | |
| <u>7.1.6 构建第四阶段(Stage4)模型</u> | |
| <u>7.1.7 构建第五阶段(Stage5)模型</u> | |
| <u>7.1.8 构建第六阶段(Stage6)模型</u> | 99 |
| <u>7.2 装配模型</u> | 100 |
| <u>7.2.1 模型流程概述</u> | 100 |
| <u>7.2.2 定义元素</u> | 100 |
| <u>7.2.3 可视化元素</u> | 101 |
| <u>7.2.4 详细定义元素</u> | 103 |
| <u>7.2.5 仿真运行</u> | 105 |
| <u>7.2.6 小结</u> | 105 |
| <u>7.3 属性模型</u> | 105 |
| <u>7.3.1 模型概述</u> | 105 |
| <u>7.3.2 元素定义</u> | 106 |
| <u>7.3.3 可视化元素</u> | 106 |
| <u>7.3.4 详细定义</u> | 107 |
| <u>7.3.5 运行模型</u> | 108 |
| <u>7.3.6 小结</u> | 108 |
| <u>7.4 输送链模型</u> | 109 |
| <u>7.4.1 模型概述</u> | 109 |
| 7.4.2 元素定义 | 109 |



| <u>7.4.3 可视化元素</u> | |
|----------------------------------|--|
| <u>7.4.4 Conveyor 详细对话框介绍</u> | |
| <u>7.4.5 详细定义</u> | |
| <u>7.4.6 运行模型及分析</u> | |
| 7.4.7 输送链类型不同的差异 | |
| 7.4.8 小结 | |
| | |
| <u>7.5.1 模型概述</u> | |
| | |
| | |
| | |
| | |
| 7.5.6 小结 | |
| | |
| 7.6.1 模型概述 | |
| 7.6.2 修改并添加元素 process time 分布的步骤 | |
| 7.6.3 详细定义 | |
| | |
| | |
| | |
| | |
| | |
| <u>7.7.4 详细定义</u> | |
| | |
| | |
| | |
| | |
| 7.8.3 定义元素 | |
| | |
| | |
| | |

<u>第八章 归档器及其应用</u> 136

| 8.1 | <u>归档器对话框介绍</u> | |
|----------|-----------------|--|
| 8.2 | 选择形成报告的数据 | |
| <u>8</u> | .2.1 添加仿真对象数据 | |
| 8 | .2.2 删除所选定的对象 | |



| 8 | 2.3 重排 report 列表中对象的次序1 | 137 |
|-----|-------------------------|-----|
| 8.3 | 生成报告1 | 137 |

<u>第九章 优化器简介和应用</u> 139

| <u>9.1 示例模型流程介绍</u> | |
|-------------------------------------|--|
| 9.2 优化变量 | |
| 9.3 相关时间值 | |
| <u>9.4 员工选择规则</u> | |
| <u>9.4.1 订单记录处理</u> | |
| <u>9.4.2 订单确认处理</u> | |
| <u>9.4.3 电脑组装处理</u> | |
| <u>9.4.4 检测处理</u> | |
| <u>9.5 目标函数</u> | |
| <u>9.5.1 函数体程序及其解释</u> | |
| <u>9.5.2 系统函数介绍</u> | |
| 9.6 优化 | |
| | |
| 9.6.2 Model Optimization 4.0 设置窗口介绍 | |
| 9.6.3 进行优化设置 | |
| <u>9.6.4 运行优化与结果分析</u> | |

<u>第十章 多产品多阶段加工仿真系统设计</u> 147

| <u>10.1 加工系统描述</u> | |
|---------------------------|--|
| <u>10.2 加工系统数据</u> | |
| <u>10.3 仿真模型的建立</u> | |
| <u>10.3.1 元素定义 define</u> | |
| | |
| | |
| | |
| | |
| 10.5 改善 | |

<u>第十一章 垃圾回收物流仿真系统设计</u> 153

| 11.1 | 垃圾回收物流系统介绍 | 153 |
|------|------------|-----|
| 11.2 | | 154 |
| 11. | | 154 |



| <u>11.2.2 与收集车辆有关的信息</u> | |
|---------------------------------|--|
| <u>11.2.3 与垃圾相关的信息</u> | |
| <u>11.3 收集垃圾的成本函数</u> | |
| <u>11.4 系统逻辑结构</u> | |
| <u>11.5 仿真模型的建立</u> | |
| <u>11.5.1 元素定义 Difine</u> | |
| <u>11.5.2 元素显示 display</u> | |
| <u>11.5.3 元素详细设计</u> | |
| <u>11.5.5 数据处理子模块</u> | |
| <u>11.5.6 目标函数 objfun()中的程序</u> | |
| <u>11.6 仿真运行与结果分析</u> | |



第一章 离散事件系统仿真简介

在现实世界中,事物并不是孤立存在。它们之间存在着内在的和有机的联系。 作为一个研究对象,我们将这种由相互联系、相互作用的事物或元素构成的统一整 体称为系统。

随着人们认识自然和改造自然的能力和手段的不断增强,作为实践经验总结的 科学和技术水平也有了迅猛的发展。从开普勒的行星运动三大定律,到牛顿的万有 引力定律,再到爱因斯坦的相对论,利用数学手段对事物描述的理论越来越完善, 而且研究的范围也越来越广,从身边发生的自然现象到广袤无垠的宇宙奇观。长期 以来,人们已经充分认识到利用数学模型去描述所研究系统的优越性、并且逐渐地 发展了系统研究和系统分析理论。但是,由于数学手段的限制,人们对复杂事物和 复杂系统建立数学模型并进行求解的能力是非常有限的。电子计算机的出现,对科 学技术的发展产生了无可估量的和深远的影响。许多复杂的数学模型可以通过计算 机来进行计算求解。由此,利用数学模型描述系统的特征并进行求解的手段逐步发 展成为现代的计算机仿真技术,计算机仿真技术有着巨大的优越性,利用它可以求 解许多复杂而无法用数学手段解析求解的问题,利用它可以预演或再现系统的运动 规律或运动过程,利用它可以对无法直接进行实验的系统进行仿真试验研究,从而 节省大量的能源和费用。由于计算机仿真技术的优越性。它的应用领域已经非常广 泛,而且也越来越受到普遍的重视。诚然,计算仿真技术中仍然存在着许多需要解 决的问题,需要不断进行努力探索。

1.1 仿真技术的产生与发展

从一般意义上讲,系统仿真可以被理解为在对一个已经存在或尚不存在但正在 开发的系统进行研究的过程中,为了了解系统的内在特性,必须进行一定的实验; 而由于系统不存在或其它一些原因,无法在原系统上直接进行实验,只能设法构造 既能反映系统特征又能符合系统实验要求的系统模型,并在该系统模型上进行实验, 以达到了解或设计系统的目的,由此可以看出,系统仿真本质上是由三个要素构成 的,即系统,系统模型和实验。系统是问题的本源,是系统分析的目的,实验是解 决问题达到目的的手段,而系统模型则是连接系统和实验(目的和手段)之间的桥梁。

显然,系统仿真是一项社会实践活动。凡是包含系统、系统模型和系统实验三 个要素的活动都可以广义地理解为系统仿真活动。

系统仿真方法的研究和应用已经有了很长的历史。在古代,人们已经从长期的 生产劳动实践活动中总结出了朴素的仿真思想。例如,古代的房屋屋顶多数为桁梁 式建筑,在建房过程中需要使用大量的木料。为了使屋顶稳定牢靠,除了要选择材 质较好粗细适当的木料外,整个屋顶的桁架结构也必须满足一定的几何形状要求; 也就是说,桁梁上的每一根木料都有确定的长度尺寸要求,不论木料长了还是短了



都可能影响整个屋顶结构的稳定性。那么如何来确定屋顶上每一根木料的具体长度 呢,显然不能拿实际的木料到屋顶上去试。这样既花费工时又可能造成木料不必要 的浪费。这个问题对现代人来说是非常简单的,利用几何和三角学的原理立刻可以 解决。但在古代科学尚不发达的情况下、解决的办法只有一个,即在地面上按实际 尺寸的一定比例模拟制作一个屋顶。经过若干次实验确定了稳定的结构之后,量出 模拟屋顶上每一根相应木料的长度,再按比例放大,即可得到实际木料所需的长度。 这是一个很典型的通过构造模型并进行实验从而获得系统特性的系统仿真实例。

仿真作为一门技术科学是在 19 世纪末 20 世纪初工业技术有了长足的发展之后 而确定下来的。而且伴随着工业技术的进步,仿真技术也在不断地发展。例如,随 着电子技术的发展,人们发现可以利用模拟电路去研究工业控制过程中的实际问题, 由此而产生了现代控制理论。而这个模拟电路就是工业控制系统的一个模型,通过 在这个模型上进行实验,就可以解决实际控制过程中产生的问题。又例如在飞机设 计过程中,对飞机的外形要求是非常严格的,因为气动外形将最终影响整个飞机的 飞行特性。由于飞机造价的昂贵。用真实的飞机去进行实验是不现实的。为了获得 飞机外形的气动数据,尤其是飞机机翼的气动数据,必须制作各种不同形状的机翼 模型放到风洞中进行实验。风洞实验的结果改进了飞机的设计理论,而利用这个理 论又可以去设计新型的飞机。在这个时期,人们在利用仿真方法研究或求解问题时, 都是利用实物去构造与实际系统成比例的物理模型,再在这个模型上进行实验。如 果这种实验是破坏性的,那么每次实验都要重新构造实物模型,带来很大的麻烦和 浪费。

1946年,世界上第一台电子计算机在美国诞生。在随后的 50 年中,计算机技术 的发展速度惊人,当今计算机的计算能力和信息处理能力已经比最初的那台笨重的 以电子管为主体的机器提高了成千上万倍。如果说早期的仿真主要是利用实际物理 模型的比例仿真,那么,现代仿真技术则是与计算机的发展密切相关的。目前通常 所讲的仿真技术一般就是指计算机仿真技术。随着计算机硬件和软件水平的提高, 计算机仿真技术也得到了很大的发展。

1.1.1 仿真软件的发展

数字仿真软件泛指一类面向仿真用途的应用软件。它的特点是面向问题和面向用 户。它的功能包括模型描述的规范及处理、仿真实验的执行与控制、仿真结果的分 析与演示、模型和数据的存储与检索。根据功能仿真软件可以分为仿真程序包、仿 真语言及仿真环境三大类。仿真软件的发展是离不开计算机软件尤其是计算机程序 设计语言的发展的。随着计算机从电子管到晶体管再到大规模集成电路不断地发展 和进步,计算机的运算速度和存储能力都有显著的提高,因此也就有了计算机程序 设计语言从机器语言到汇编语言再到高级程序设计语言的发展历程。而这一切又为 仿真软件的产生与发展提供了必要的条件。

历史上第一个仿真软件是由塞尔弗里奇(R. G. Selfridge) 在 1955 年开发的。他完



成了利用辛普森方法进行数值积分的仿真程序设计工作。从那之后,仿真软件的发 展经历了四个阶段:

第一阶段是从 50 年代到 60 年代初期,以 Fortran 语言为代表的通用程序设计语言阶段,Fortran 语言是达到成熟的第一个高级程序设计语言。当时几乎所有用于求解数学表达式的程序都是用 Fortran 语言编写成的,即使在目前,也有许多大型的通用仿真语言是基于 Fortran 语言编制的。

第二阶段是 60 年代到 70 年代,出现了多种仿真程序包及初级仿真语言。这个时 期仿真软件主要解决的问题是利用数字仿真方法求解常微分方程组。例如 1961 年由 贝尔实验室开发的用于实现数据采集系统仿真的面向框图的程序 BLODI (Block Diagram compiler),1962 年为了工业动力学系统仿真专门开发的语言 DYNAMO (DYNAmic Models)。1983 发表的用于求解常微分方程组的仿真程序 MIDAS (Modified Integration Digital Analog Simulator)等等,直到 1967 年,为了促进已有的 几十种数字仿真语言的标准化,美国计算机仿真学会 SCS 提出了 CSSL(Continuous System Simulation Language)标准,后来开发的仿真语言大都遵循这个标准。在此阶 段的仿真语言中,比较典型的还有 1964 年由 IBM 公司的 G.戈登(Gordon)开发的高 度结构化的利用进程交互法进行排队问题仿真的专用仿真语言 GPSS(General Purpose Simulation System)。

第三阶段在 70 年代到 80 年代初期,出现了高级完善的商品化仿真语言。这个阶段仿真语言的特点是在以下几个方面比早期的仿真语言更加成熟和全面。

- ·模型的表达能力
- · 数值性能和算法
- ·语言的结构特征
- ·模型验证
- ·程序执行方式
- ·数据管理和处理能力
- ·输入输出特性

例如在 7 0 年代中期推出的算法全面,功能强大的求解常微分方程和差分方程 问题的仿真语言 CSSL-IV 和 ACSL(Advanced Continuous Simulation Language),以及 1971 年推出的用于离散事件仿真的可以用类似自然语言自由格式描述系统模型的仿 真语言 Simscript I.5 和应用广泛的随机网络建模的 SLAM 仿真语言。

第四阶段是 80 年代中期开始的一体化建模与仿真环境研究。其背景是:

·随着建模与仿真工作要求的提高,已开发的各种仿真软件经常不能协调地工作;

- ·对仿真语言的要求越来越复杂;
- ·存在大量的数据处理及文档化工作;
- ·不同的用户(建模者,仿真实验人员,决策者)对仿真工具有不同的要求;
- 计算机网络技术和数据库技术有较大的发展;
 - 一体化建模与仿真环境的主要性能表现在:
- ·支持建模与仿真的全寿命周期活动;



- 集成化程度高;
- 方便友好的用户接口;
- ·初步的知识处理能力;
- ·模型与仿真的质量保证措施;
- ·开放性;

在当今市面上,仿真可用使用专用软件来实现。下面列举了一些仿真软件: 20-sim、arena、Automod、Awesim、Easy5、Idef、Intrax、Manufacturing Engineering、 Matlab、Modsim、Promodel、Prosolvia、Quest、SDI supply chain 以及 Witness。

Witness(SDX):该软件提供离散事件仿真。该软件具备的多种工具使得对自动 化制造系统进行仿真非常容易。周转时间、损坏模式和定时,调整模式和定时,缓 冲设备容量和保存时间,机器类型等连同路径信息都为仿真提供了方便性。该软件 还包括物料流动优化,虚拟现实功能,有效地物流流动可以最小化设备间物料和产 品流动的费用。更多资料可以参看公司的网站:<u>www.lanner.com</u>。

1.1.2 仿真建模方法学的发展

仿真是在系统模型上进行实验的过程。利用计算机进行仿真就必须建立能够被 计算机识别并在计算机上运行的系统模型,也就是说,通过对系统进行分析,首先 建立描述系统行为规律的系统模型,再将其转换为计算机仿真程序。仿真运算过程 就是对系统模型求解的过程。为了通过仿真分析能够准确地掌握系统的内在运动规 律,在仿真中以下两个方面是非常重要的。一是建立准确的系统模型,二是获得正 确的仿真结果。

早期计算机仿真的对象是对工程技术领域中的实际物理过程进行仿真。该领域 中的问题(例如系统的控制和优化)涉及机械、电子、制造、航空等诸多背景,这些问 题的特点是可以建立起以时间为基准的数学模型,即连续时间模型和离散时间模型, 包括常微分方程、偏微分方程和差分方程等。利用实际工程背景中的原理和定理可 以推导出所研究问题的时间微分或差分方程模型,而根据系统自身的特征和试验数 据可以确定模型中的参数。在 50 年代至 60 年代,人们在差分方程和微分方程模型 的结构特征化和参数辨识方面花费了相当的精力并取得了很大的成果,与此同时, 对求解这些方程的算法的研究也在不断发展。产生了能够满足快速和实时等多种不 同要求的仿真算法和仿真软件。

进人 70 年代,仿真逐步向政治、经济、军事等社会科学领域渗透,出现了许多 用于求解这些领域中问题的数学模型。而随着对这些问题的深入分析和了解,数学 模型从早期的微分方程和差分方程模型逐渐向能够反映问题离散和随机特点的离散 事件逻辑流图和网络图模型过渡。同时,从求解静态模型的蒙特卜罗(Monte Carlo) 法到研究系统动态模型的以事件调度法、活动扫描法和进程交互法为代表的仿真策 略,离散事件模型的仿真算法研究也取得了很大的发展。

由于离散事件模型的构造比微分方程和差分方程复杂,而且建立的模型多种多



样,即使对同一个系统也可以建立许多不同的模型,因此人们一直在探索用统一的 建模方法来指导和简化离散事件模型的建模过程。70 年代中期 B. P.齐格勒 CB. P. Zeigler)提出了模型的规范化和形式化描述理论,使得建模方法学前进了一大步。从 那时起,结合计算机软件方法学的发展,系统建模理论中引入了层次化模块化方法 和面向对象的思想,为建立集成化交互式建模环境提供了良好的基础。

1.2 系统、模型与仿真

1.2.1 系统

系统仿真的研究对象是具有独立行为规律的系统。所谓系统是指相互联系又相 互作用着的对象的有机组合。从广义上讲,系统的概念是非常广阔的。大到无垠的 宇宙世界,小到分子原子,我们都可以称之为系统。

根据系统的物理特征可以将系统划分为两大类,即工程系统和非工程系统。所 谓非工程系统是指自然和社会在发展过程中形成的,被人们在长期的生产劳动和社 会实践中逐渐认识的系统。例如社会、经济、管理、交通、生物系统等属于非工程 系统,所谓工程系统是指人们为满足某种需要或实现某个预定的功能,利用某种手 段构造而成的系统。工程系统的例子非常多,如机械、电气,动力、化工、武器系 统等。

对于一个系统来说,不论它是大是小,都必然存在三个要素,即实体、属性和 活动。

所谓实体是指组成系统的具体对象。例如,在商品销售系统中的实体有经理、 部门、商品货币、仓库等。系统中的各个实体既具有一定的相对独立性,又相互联 系构成一个整体。

所谓属性是指实体所具有的每一项有效特性。例如,商品的属性有生产日期、 进货价格、销售日期、售价等。

所谓活动是指随着时间的推移、在系统内部由于各种原因而发生的变化过程。 例如零售商品价格的增长等。

系统是在不断地运动、发展、变化的。由于组成系统的实体之间相互作用而引 起实体属性的变化,使得在不同的时刻,系统中的实体和实体属性都可能会有所不 同,这种变化通常用状态的概念来描述。在任意给定时刻,系统中实体、属性以及 活动的信息总和称为系统在该时刻的状态;用于表示系统状态的变量称为状态变量。

系统不是孤立存在的。自然界中的一切事物都存在着相互联系和相互影响。任 何一个系统都将经常由于系统之外出现的变化而受到影响。这种对系统的活动结果 产生影响的外界因素称为系统的环境。在对一个系统进行分析时,必须考虑系统所 处的环境,而首要的便是划分系统与其所处环境之间的边界。系统边界包围系统中 的所有实体。



系统边界的划分在很大程度上取决于系统研究的目的。例如在商品销售系统中, 如果仅考虑商品仓库库存量的变化情况,那么系统只需包括采购部门、仓库以及销 售部门即可。但若要研究商品进货与销售的关系时,系统中还要包括市场调查部门, 因为商品销售状况及对进货的影响这部分职能是由该部门完成的。

另一方面,系统在某些条件下是可以分解的。也就是说,构成系统的某个实体 本身也可以看成为一个单独的系统来进行分析研究,这个系统称为原系统的一个子 系统或分系统。

系统研究包括系统分析、系统综合和系统预测等方面。研究系统首先需要描述 清楚所研究系统的实体、属性、活动及环境。因为系统的概念不仅与实体有关。而 且与研究者的目的有关,只有在对实体、属性、活动、环境作了明确的描述之后, 系统才是确定的。

1.2.2 模型

系统模型可以定义为:为了达到系统研究的目的,用于收集和描述系统有关信息 的实体。

模型是对相应的真实对象和真实关系中那些有用的和令人感兴趣的特性的抽象,是对系统某些本质方面的描述,它以各种可用的形式提供被研究系统的信息。 模型描述可视为是对真实世界中的物体或过程相关信息进行形式化的结果。模型在 所研究系统的某一侧面具有与系统相似的数学描述或物理描述。从某种意义上说, 模型是系统的代表,同时也是对系统的简化。另一方面,模型应足够详细,以便从 模型的实验中取得关于实际系统的有效结论。

一般来说,系统模型的结构具有相似性、简单性、多面性等性质。

1.2.3 仿真

给出一个系统的数学模型之后,有时用分析手段就可以求解系统有关的信息,但 是当不能应用分析法的时候,就需要应用仿真方法求解。

与应用数学分析方法求解问题相比较,仿真方法求解问题的主要缺点是很明显 的,即它只能给出问题的特解而不能给出问题的通解。然而,能用数学分析法求解 的问题的范围毕竟是有限的。用数学分析法求解问题时,要对系统加以抽象和近似 处理。以使模型适于用数学分析方法求解,在许多方面,理想情况下是把仿真方法 的应用作为已经得到的、因过于简化的数学分析解答的一种补充。系统、模型与仿 真三者之间有着十分密切的关系,系统是研究对象,模型是系统特性的描述,仿真 则包含建立模型及对模型进行试验两个过程。

1.2.3.1 仿真的分类

根据模型的类型,系统仿真可以分成物理仿真、数学仿真和物理-数学仿真。 按照真实系统的物理性质构造系统的物理模型,并在物理模型上进行实验,就

称为物理仿真。而按照真实系统的数学关系构造系统的数学模型,并在数学模型上 进行实验,就称为数学仿真。把系统的一部分写成数学模型,而另一部部分则构造 其物理模型,然后将它们联接成系统模型进行实验,就称为物理-数学仿真,也称为 半实物仿真。

物理仿真的优点是直观且形象化,但建模周期长、花费大。数学仿真的特点是 经济、方便。计算机为数学模型的建立与实验提供了较大的灵活性,目前数学仿真 一般就是在计算机上建立系统的数学模型并进行实验。因此数学仿真通常也称为计 算机仿真。

根据仿真中所用计算机的类型,计算机仿真又可以分为模拟仿真,数字仿真和 混合仿真。

模拟仿真是基于数学模型相似原理上的一种方法,仿真的主要工具是模拟计算机,模拟仿真的特点是直观、运算速度快,但精度较差。

数字仿真基于数值计算原理,仿真的主要工具是数字计算机和仿真软件。数字 仿真自动化程度高,具有复杂逻辑判断的能力,而且可以获得较高的精度。

混合仿真是将模拟仿真和数字仿真相结合的一种方法,仿真的主要工具是混合 计算机系统。混合仿真兼备模拟仿真和数字仿真的优点,可以快速地进行多次仿真 研究,因此特别适用于参数寻优,统计分析等方面的应用,尤其是在复杂系统的实 时仿真方面体现出极大的优越性。

根据仿真的研究对象,系统仿真可以分成连续系统仿真和离散事件系统仿真, 连续系统是指系统的状态随时间连续变化的系统。这里要注意有些连续系统如数据 采集系统的状态数据是在离散时间点上获得的,是非连续的,但其状态本身则是连 续变化的。连续系统的模型可以用一组连续的方程描述。

离散事件系统的特点是系统的状态变化只在离散的时间点上发生,且发生时刻 往往是随机的,系统的状态变化是由随机事件驱动的。

1.2.3.2 计算机仿真

w/

计算机仿真就是采用计算机对数学模型进行仿真实验。现代的仿真系统主体(包括硬件和软件)都离不开计算机,计算机仿真方法主要解决下述两个问题:

● 提供计算机能接受的仿真模型;

提供在计算机上运行计算和进行仿真研究的方法。

计算机仿真摆脱了物理模型的传统概念,不同的数学模型可在同一台计算机上运行。仿真研究要求借助计算机实现便于进行实验的"活"的数学模型,即提供便于程序设计的方法,使仿真人员能集中精力对仿真结果进行分析和处理,由于可以对物理性质截然不同的各种系统进行准确、灵活、可靠的研究,这就使现代科学实验技术提高到一个新的水平。

现代仿真技术的发展是与计算机应用和发展紧密相联系的,在计算机尚未问世 之前,由于只有物理仿真,因此系统仿真是附属在与所研究系统有关的学科中的。 而只有在计算机出现以后,由于数学仿真的发展,提出了大量共同性的技术问题,



系统仿真才逐渐发展为一门以计算机仿真为代表的独立的学科。从 40 年代末的模拟 计算机仿真开始,逐渐发展到采用混合计算机、数字计算机和全数字并行处理机的 仿真,其应用领域越来越广泛。至今,数字计算机已成为系统仿真的主要工具。多 数情况下系统仿真即指计算机仿真,尤其特指数字计算机仿真。

计算机仿真包括三个要素,即系统、模型和计算机。联系这三个要素的有三个 基本活动:系统模型建立、仿真模型建立和仿真实验。图 1 描述了计算机仿真三要 素及三个基本活动的关系。

建模活动是通过对实际系统的观测和检测,在忽略次要因素及不可检测变量的 基础上,用物理或数学的方法进行描述,从而获得实际系统的简化近似模型。仿真 模型反映了系统模型同仿真器或计算机之间的关系,能为仿真器及计算机所接受并 在其上运行。仿真实验就是将系统的仿真模型置于计算机上运行的过程。系统仿真 是通过实验来研究实际系统的一种技术,通过仿真活动可以弄清系统内在结构变量 和环境条件的影响。



图 1.1 计算机仿真三要素及其关系

1.3 离散事件系统仿真

离散事件系统的仿真就是按照实际的工作流程,在规定时间内顺序地改变实体 或设备的状态。所谓工作流程是指实体在整个仿真过程中活动的顺序。每发生一个 事件,系统的状态就发生一次变化,在实际活动中、事件的发生不是连续的,发生 时间的间隔也不相等,而是具有某种随机性。

1.3.1 基本概念

为了了解系统仿真的基本方法,首先需要掌握与系统仿真有关的一些基本概念: 1.3.1.1 **事件**

事件是描述系统的另一基本要素。时间是指引起系统状态变化的行为,系统的 动态过程是靠事件来驱动的。例如,在物流系统中,工件到达可以定义为一类事件。 因为工件到达仓库,进行入库时,仓库货位的状态会从空变为满,或者引起原来等 待入库的队列长度的变化。

事件一般分为两类:必然事件和条件事件。只与时间有关的事件称为必然事件。



如果事件发生不仅与时间因素有关,而且还与其它条件有关,则称为条件事件。系 统仿真过程,最主要的工作就是分析这些必然事件和条件事件。

1.3.1.2 成分

描述系统的第三个基本要素是成分。成分与实体是同一概念,只是根据习惯, 在描述系统时用实体而在模型描述中用成分。成分分为主动成分和被动成分。可以 主动产生活动的成分称为主动成分,如物流系统中的工件,它的到达将产生入库活 动或排队活动。本身不产生活动,只在主动成分作用下才产生状态变化的那些成分 称为被动成分。

1.3.1.3 进程

若干事件与若干活动组成的过程称为进程。它描述了各事件活动发生的相互逻辑关系及时序关系,例如,工件由车辆装入进货台,经装卸搬运进入仓库,经保管、加工到配送至客户的过程(如图 1.2)。



1.3.1.4 仿真钟

仿真钟用于表示仿真事件的变化。在离散事件系统仿真中,由于系统状态变化 是不连续得,在相邻两个事件发生之间,系统状态不发生变化,因而仿真钟可以跨 越这些"不活动"区域。从一个事件发生时刻,推进到下一个事件发生时刻。仿真 钟的推进成跳跃性,推进速度具有随机性。由于仿真实质上是对系统状态在一定时 间序列的动态描述,因此,仿真钟一般是仿真的主要自变量,仿真钟的推进是系统 仿真程序的核心部分。

应当指出,仿真钟所显示的是仿真系统对应实际系统的运行时间,而不是计算机 运行仿真模型的时间。仿真时间与真实时间将设定成一定比例关系,使得像物流系 统这样复杂的系统,真实系统运行若干天,若干月,计算机仿真只需要几分钟就可 以完成。

1.3.1.5 随机变量

复杂的现实系统常常包含有随机的因素。在物流系统中工件的到达、运输车辆 的到达和运输事件等一般都是随机的。这些复杂的随机系统很难找到响应的解析驶 来描述和求解。系统仿真技术成了解决这类问题的有效方法。



对于有随机因素影响的系统进行仿真时,首先要建立随机变量模型,即确定系统的随机变量并确定这些随机变量的分布类型和参数。对于分布类型是已知或者是可以根据经验确定的随机变量,只要确定它们的参数就可以了。无论是确定随机变量的分布类型还是确定其参数,都要以调研观测的数据为依据。

1.3.2 离散事件系统仿真方法

离散事件系统仿真与连续系统仿真的方法很不相同。

离散事件系统模型只是一种稳态模型,无须研究状态变量从一种状态变化到 另一种状态的过程。而对于连续系统,主要是研究其动态过程,连续系统模型一般 要用微分方程描述。

离散事件系统中的变量大多数是随机的,例如实体的"到达"和"服务"时 间都是随机变量。仿真实验的目的是力图用大量抽样的统计结果来逼近总体分布的 统计特征值,因而需要进行多次仿真和较长时间仿真。

连续系统仿真中采用均匀步长推进仿真钟的原则,则离散事件系统仿真中时 间的推进是不确定的,它决定于系统的状态条件和事件发生的可能性。

离散事件系统仿真实质上是对那些由随机系统定义的,用数值方式或逻辑方式 描述的动态模型的处理过程。从处理手段上看,离散事件系统仿真方法可分为两类。

面向过程的离散事件系统仿真

面向过程的仿真方法主要研究仿真过程中发生的事件以及模型中实体的活动; 这些事件或活动的发生是顺序的。而仿真时钟的推进正是依赖于这些事件和活动的 发生顺序,在当前仿真时刻,仿真进程需要判断下一个事件发生的时刻或者判断触 发实体活动开始和停止的条件是否满足,在处理完当前仿真时刻系统状态变化操作 后,将仿真时钟推进到下一事件发生时刻或下一个最早的活动开始或停止时刻。仿 真进程就是不断按发生时间排列事件序列,并处理系统状态变化的过程。

面向对象的离散事件系统仿真

在面向对象仿真中,组成系统的实体以对象来描述。对象有三个基本的描述部 分,即属性、活动和消息。每个对象都是一个封装了对象的属性及对象状态变化操 作的自主的模块,对象之间靠消息传递来建立联系以协调活动。对象内部不仅封装 了对象的属性还封装了描述对象运动及变化规律的内部和外部转换函数。这些函数 以消息或时间来激活,在满足一定条件时产生相应的活动。消息和活动可以同时产 生,即所谓的并发,但在单 CPU 计算机上,仍须按一定的仿真策略进行调度。在并 行计算机和分布式仿真环境中,仿真策略则可以更加灵活、方便。面向对象的仿真 尤其适用于各实体相对独立、以信息建立相互联系的系统中,如航空管理系统、机 械制造加工系统、以及武器攻防对抗系统等。

18



1.3.3 仿真策略

离散事件系统仿真方法适用于状态变量是离散变化、时间连续变化的一类系统 的仿真问题。随机时刻点上发生的事件引起系统中实体的状态变化。描述这类系统 的模型一般不是一组数学表达式,而是一幅表示数量关系和逻辑关系的流程图。离 散事件系统的算法体现在其建模框架和仿真策略之中。有三类基本仿真策略:

1.3.3.1 事件调度法

按这种策略建立模型时,所有事件均放在事件表中,模型中设有一个时间控制 成分,该成分从事件表中选择具有最早发生时间的事件,并将仿真钟修改到该事件 发生的时间,再调用与该事件相应的事件处理模块,该事件处理完后返回时间控制 成分。这样,事件的选择与处理不断地进行,直到仿真终止的条件或程序事件产生 为止。

1.3.3.2 活动扫描法

在此方法中,系统由部件组成,而部件包含着运动,这些活动的发生应当满足 规定事件发生的条件。每一个成分均有一个激活条件,若条件满足,则激活该成分 的活动例程。仿真过程中,活动的发生时间也作为条件之一,而且较之其它条件具 有更高的优先权。即在判断激活条件时首先判断该活动发生的时间是否满足,然后 再判断其它条件。对活动的扫描循环进行,直到仿真终止为止。

1.3.3.3 进程交互法

这种方法的特点是系统仿真钟的控制程序采用两张事件表,其一是当前事件表 (FEL:Current Events List),它包含了从当前时间点开始有资格执行的事件记录,但 是该事件是否发生的条件尚未判断。其二是将来事件表(FEL:Future Events List), 它包含在将来某个仿真时刻发生的事件记录。每一个事件记录中包含该事件的若干 属性,其中必有一个属性,说明该事件在过程中所处位置的指针。

进程交互法首先按一定分布产生到达实体并置于 FEL 中,实体进人排队等待。 然后对当前事件表进行扫描,判断各种条件是否满足,再对满足条件的活动进行处 理。仿真钟推进到服务结束并将该实体从系统中清除,最后将 FEL 中为当前事件的 实体移到当前事件表中。

1.4 系统仿真的一般步骤

对于每一个成功的仿真研究项目,其应用都包含着特定的步骤。不论该研究的类型和目的,仿真的过程是保持不变的。一般要进行如下9步

- 问题定义
- 制定目标
- 描述系统并对所有假设列表



- 罗列出所有可能替代方案
- 收集数据和信息
- 建立计算机模型
- 校验和确认模型
- 运行模型
- 分析输出

下面对这九步作简洁的定义和说明。它不是为了引出详细的讨论,仅仅起到抛砖引玉的作用。注意仿真研究不能简单遵循这九步的排序,有些项目在获得系统的内在细节之后,可能要返回到先前的步骤中去。同时,验证和确认将贯穿于仿真工程的每一个步骤当中,如图 1.3 所示。

1.4.1 问题的定义

一个模型不可能呈现被模拟的现实系统的所有方面,有时是因为太昂贵。另外, 假如一个表现真实系统所有细节的模型也常常是非常差的模型,因为它将过于复杂 和难于理解。因此,明智的做法是:先定义问题,再制定目标,然后构建一个能够 完全解决问题的模型。在问题定义阶段,对于假设要小心谨慎,不要做出错误的假 设。例如,假设叉车等待时间较长,比假设没有足够的接收码头要好。作为仿真大 纲,对问题的陈述越普通越好。



图 1.3 系统仿真流程图



1.4.2 制定目标和定义系统效能测度

没有目标的仿真研究是毫无用途的。目标是仿真工程所有步骤的导向。系统的 定义是基于系统目标的;目标决定了该作出怎样的假设;目标决定了应该收集那些 信息和数据;模型的建立和确认专门是考虑是否满足目标的需求。目标需要清楚、 明确和切实可行。目标经常被描述成像这样的问题"通过添加机器或延长工时,能 够获得更多的利润吗?"在定义目标时,详细说明那些将要被用来决定目标是否实 现的性能测度是非常必要的。每小时的产出率、工人利用率、平均排队时间、以及 最大队列长度是最常见的系统性能测度。

最后,列出仿真结果的先决条件。如,必须通过利用现有设备来实现目标,或 最高投资额要在限度内,或产品订货提前期不能延长等。

1.4.3 描述系统和列出假设

简单点说,仿真模型降低完成工作的时间。系统中的时间被划分成处理时间、 运输时间和排队时间。不论模型是一个物流系统、制造工厂、或服务机构,清楚明 了的定义如下建模要素都是非常必要的:资源、流动项目(产品、顾客或信息)路 径、项目运输、流程控制、加工时间,资源故障时间。下面是对各要素的简要描述:

仿真将现实系统资源分成四类:处理器、队列、运输和共享资源如操作员。流 动项目的到达和预载的必要条件必须定义,如:到达时间、到达模式和该项目的类 型等属性。在定义流动路径时,合并和转移需要详细的描述。项目的转变包括属性 变化、装配操作(项目和并)拆卸操作(项目分离)。在系统中,常常有必要控制 项目的流动。如:一个项目只有在某种条件或某一时刻到来时才能移动,以及一些 特定的规则。所有的处理时间都要被定义,并且要清楚表明那些操作是机器自动完 成,哪些操作是人工独立完成,哪些操作需要人机协同完成。资源可能有计划故障 时间和意外故障时间。计划故障时间通常指午餐时间,中场休息,和预防性维护等。 意外故障时间是随机发生的故障所需的时间,包括失效平均间隔时间和维修平均间 隔时间。

在这些工作完成之后,需要将现实系统作模型描述,它远比模型描述向计算机 模型转化困难。现实向模型的转化意味着你已经对现实有了非常彻底的理解,并且 能将其完美的描述出来。这一阶段,将此转换过程中所作的所有假设作详细说明非 常有必要。事实上,在整个仿真研究过程中,所有假设列表保持在可获得状态是个 很好的主意,因为这个假设列表随着仿真的递进还要逐步增长。假如描述系统这一 步做得非常棒,建立计算机模型这一阶段将非常简便。

注意,获得足够的,能够体现特定仿真目的的<u>系统</u>本质的材料是必要的,但是 不需要获得与真实系统一一对应的模型的描述。正如爱因斯坦所说"做到不能再简 单为止"。



1.4.4 列举可能的替代方案

在仿真研究中,确定模型早期运行的可置换场景是很重要的。它将影响着模型的建立。在初期阶段考虑替代方案,模型可能被设计成可以非常容易的转换到替换 系统。

1.4.5 收集数据和信息

收集数据和信息,除了为模型参数输入数据外,在验证模型阶段,还可以提供 实际数据与模型的性能测度数据进行比较。

数据可以通过历史纪录、经验、和计算得到。这些粗糙的数据将为模型输入参数提供基础,同时将有助于一些需要较精确输入参数数据的收集。

现存数据并不是总是有用的,通过测量收集数据可能要费时、费钱。同立马行 动,收集系统输入参数数据相比,除了分析中,模型参数需要极为精确的输入数据外, 采用估计方法来产生输入数据更为高效。估计值可以通过少数快速测量或者通过咨 询熟悉系统的系统专家来得到。即使是使用较为粗糙的数据,根据最小值、最大值 和最可能取值定义一个三角分布,不仅仅采用平均值仿真效果都要好得多。有时候 采用估计值也能够很好的满足仿真研究的目的。例如,仿真可能被简单的用来指导 人员了解系统中特定的因果关系。在这种情况下,估计值就可以满足要求。

当需要可靠数据时,花费较多时间收集和统计大量数据,用以定义出能够准确 反映现实的概率分布函数就是非常必要的。需要的数据量的大小取决于变量的变异 程度,但是也有通用的规则,大拇指法指出至少需要三十甚至上百的数据。假如要 获得随即停机时间的输入参数,必须要在一个较长时间段内捕获足够多的数据。

1.4.6 构造计算机模型

构造计算机模型的过程中,仿真研究目标要记在心中。首先构建小的测试模型 来证明复杂部件建模是合适的。一般建模过程是呈现阶段性的,在进行下一阶段建 模之前,验证本阶段的模型工作正常。而从不会将整个系统模型构建起来,然后点 击"运行"按钮来进行系统的仿真。在建模过程中运行和调试每一阶段的模型。我 们可能想对同一现实系统构建多个计算机模型,每个模型的抽象程度都不相同。抽 象模型有助于定义系统的重要部分,并可以引导为后续模型的详细化而进行的数据 收集活动

1.4.7 验证和确认模型

验证是确认模型的功能是否同设想的系统功能相符合。模型是否同我们想构建的模型相吻合?产品的处理时间、流向是否正确?

W

确认范围更广泛。它包括:确认模型是否能够正确反映现实?评估模型仿真结 果的可信度有多大?

1.4.7.1 验证模型

现在有很多技术可以用来验证模型。最最重要的、首要的是在仿真低速运行时, 观看动画和仿真钟是否同步运行,它可以发现物料流程及其处理时间的总的差异。

另一种验证技术是在模型运行过程中,通过交互命令窗口,显示动态图表来询 问资源和流动项目的属性和状态。

通过"步进"方式运行模型,和动态查看轨迹文件可以帮助人们调试模型。运 行仿真时,通过输入多组仿真输入参数值,来验证仿真结果是否合理也是一种很好 的方法。在某些情况下,对系统性能的一些简单测量可以通过手工或使用对比而来 获得。对模型中特定区域要素的使用率和产出率通常是非常容易计算出来的。

在调试模型中是否存在着某种特定问题时,推荐使用同一随机数流,这样可以 保证仿真结果的变化是由对模型所做的修改引起的,同时对随机数流不做改动,有 时对于模型运行在一些简单化假设下,非常有帮助,这些假设是为了更加简便的计 算或预测系统性能。

1.4.7.2 确认模型

模型确认建立模型的可信度。但是,现在还没有哪一种确认技术可以对模型的 结果作出 100%的确定。我们永远不可能证明模型的行为就是现实的真实行为。如果 我们能够做到这一步,可能就不需要进行仿真研究的第一步(问题的定义)了。我 们尽力去做的,最多只能是保证模型的行为同现实不会相互抵触罢了。

通过确认,我们试着判断模型的有效程度。假如一个模型在得到我们提供的相 关正确数据之后,其输出满足我们的目标,那么它就是好的。模型只要在必要范围 内有效就可以了,而不需要尽可能的有效。在模型结果的正确性同获得这些结果所 需要的费用之间总存在着权衡。

下面是判断模型有效性的一些指标:

- 損型性能测度是否同真实系统性能测度匹配
- 2) 如果没有现实系统来对比,可以将仿真结果同相近现实系统的仿真模型的相 关运行结果作对比。
- 3)利用系统专家的经验和直觉来假设复杂系统特定部分模型的运行状况。
- 4) 在每一主要人物确认模型的输入和假设都是正确的,模型的性能测度都是可 以测量的之前,对模型作各部分的随意测试。整个团队的智慧都对模型的有 效性作出了贡献。
- 5) 模型的行为是否同理论相一致?确定结果的理论最大值和最小值,然后验证 模型结果是否落入两值之间。
- 6) 对于我们了解改变输入值,其输出的变化方向的特定性能测度,可以通过改 变其输入参数,来验证一致性。
- 7)模型是否能够准确的预测结果?这项技术用来对正在运行中的模型进行连续



的有效性验证。

8) 是否有其他仿真模拟器模拟了这个模型?要是有的话那就再好不过了,可以 将两个模型的运行结果进行对比。

1.4.8 运行可替代实验

当系统具有随机性时,就需要对实验做多次运行。因为,随机输入导致随机输 出。如果可能,在第二步中应当计算出已经定义的每一性能测度的置信区间。可替 代环境能够单独构建,并可以通过手工使用 WITNESS 软件中的 SDX 模块来进行模 拟,或通过使用"OPTIMIZER"模块自动运行模拟。

在选择仿真运行长度时,考虑启动时间,资源失效可能间隔时间,处理时间或 到达时间的时间或季节性差异,或其他需要系统运行足够长时间才能出现效果的系 统特征变量,是非常重要的。

1.4.9 输出分析

报表、图形、表格和置信区间点图将被用于输出结果分析。置信区间指出性能 测度依赖的范围。这个使用上、下限来表示。上限和下限之差称为精度。精度的可 靠性用百分比来表示。统计技术用来分析不同场景的模拟结果。一旦分析结果并得 出结论,我们要能够根据模拟的目标来解释这些结果。通常,使用结果和方案的矩 阵非常有帮助。



第二章 随机分布

在离散事件系统仿真中,系统内部实体的活动是不确定的,事件的发生具有随 机的性质,对于这种随机的过程需要采用概率及统计的方法描述,离散事件系统模 型一般与概率分布有关,因此建模者需要从已知的数据中选择合适的分布形式,作 该分布的参数估计,然后进行检验以观察是否获得合适的拟合,最终得到离散事件 系统的概率模型。随机变量在离散事件系统仿真中起着重要的作用,因此掌握随机 变量的有关知识是非常必要的。在本章中,首先对仿真中概率统计的基本知识做简 要说明,然后介绍在离散事件系统仿真中常见的概率分布形式,以及随机数和随机 变量的产生方法,最后介绍 Witness 提供的标准随机分布函数。

2.1 概率统计基本概念

2.1.1 确定事件和随机事件

确定事件:在给定条件下进行的试验中,一定发生或一定不发生的事件分别称为 必然事件和不可能事件,这类事件是确定性的,总称为确定事件。

随机事件:在给定条件下进行的试验中,可能发生也可能不发生,而在大量重复 试验中却具有某种规律性的事件,称为随机事件。

2.1.2 随机变量与概率

随机变量:如果试验的每个结果用变量的一个值来表示,即E的值根据试验结果 来确定,因而它取什么值是随机的,而且对任意实数, < 是一个随机事禅,这 种变量称为随机变量。

概率:如果对某项试验重复进行了 n 次,事件 A 发生了 m 次,则称 A 在这次试验中的概率为 m/n。在大量重复某项试验时,就会发现事件的频率在试验次数很大而且不断增大的过程中呈现稳定性。这种统计规律性表明:事件发生的可能性大小是事件本身固有的客观属性。称事件 A 发生可能性的大小为事件 A 的概率,记为 P(A)。当试验次数 n 足够大时,可以用事件的频率作为事件概率的近似值,即

P(A) m/n

概率是一个在区间[0,1]上取值的实数。

随机变量 取值小于实数 的可能性大小,即随机事件{ < }发生的概率,介 于数0和1之间:

0 P(<) 1



2.2 离散事件系统仿真中常见的概率分布

在离散事件系统的建模过程中,经常要研究一些不确定的随机事件,需要引人 概率分布,这在典型的离散事件系统如排队系统、库存系统以及系统可靠性、维修 性研究中都有很多的实例。在排队系统中,实体到达的时间间隔和接受服务的时间 通常是不确定的。在存储系统之中,需求量以及订货到收货的时间间隔也是不确定 的。在系统可靠性模型中,也有很多不确定的因素,如系统内部件发生故障的时间 以及发生故障部件的修复时间都是不确定的。在这些系统中,随机事件发生的概率 必然满足一定的分布,而系统建模的任务之一就是确定系统中各种随机事件发生的 概率分布形式。完全准确地描述这些分布形式是困难的,一般是从己知的概率分布 形式中寻找近似者。各种已知的概率分布都有其各不相同的特性,因此,选择适宜 的随机变量概率分布形式,建立合理的系统概率统计模型是系统建模的一个重要方 面。本节讨论在上述几种典型的离散事件系统应用领域中适用的概率统计模型。

2.2.1 排队系统

在排队系统中,主要有两种类型的活动,即实体到达和实体接受服务。一般情况下,实体到达的时间间隔是不确定的,从而在一定时间内到达的实体数目也是一个随机变量;另一方面,实体接受服务的时间也总是不确定的,从而造成队列的长短也是随机的。

实体到达的模式一般用泊松分布来描述,即在固定的时间内到达系统的实体数 目服从泊松分布。这种模式的特点是:

在一定时间间隔内到达实体的数目仅与时间间隔的长短有关,而与这段时间间隔的起始时刻无关。

在某个时间间隔内,到达的实体数目与在此之前到达的实体数目无关,也不 影响在此之后实体的到达。

不存在两个或两个以上实体同时到达的情况。

若在一定时间内到达系统的实体数目 服从参数为 A 的泊松分布,则相邻到达的两个实体之间的到达时间间隔 T 服从参数为 λ 的指数分布。

如果服务时间完全是随机的,通常在建模过程中用指数分布描述。服务时间也 可能是在某个常数附近波动,例如同样产品的加工时间应当总是相同的,但是由于 产品自身或加工工具的原因可能引起加工时间稍有不同。在这种情况下,服务时间 可以用正态分布描述。

T 分布和威布尔分布也可以被用于模仿到达时间间隔和服务时间。实际上,指 数分布可以看成是 T 分布和威布尔分布的特殊情况。



2.2.2 库存系统

在现实的库存系统中有三个随机变量:

每次订货或单位周期内的需求量,

两次需求间的时间间隔,

发出订单和收到订货间的时间间隔,也称为提前期。

在最简单的库存系统数学模型中,需求量始终为常数,而提前期也是零或常数。 但大多数实际情况中需求的时间是随机出现的,而每次的需求量也是随机的。

从一般意义上讲,订货提前期的分布通常和 T 分布是非常拟合的。但在研究库 存系统时,考虑的主要因素是需求量而不是提前期,提前期一般只视为对库存策略 的一种制约,因此在多数情况下,将提前期的分布简化为用均匀分布来描述或者用 正态分布来描述。

几何分布、二项分布以及泊松分布都可以用来描述需求的分布,它们分别提供 了满足各种需求模式的分布形式,即在一定周期内的需求量的分布。几何分布是特 殊情况的二项分布,它描述了至少出现一次需求的概率。常用的描述需求的分布形 式是泊松分布。

2.2.3 可靠性与维修性

可靠性对于那些一旦发生故障就会造成重大损失的系统来说是至关重要的。在 系统可靠性与维修性建模中,优先考虑的随机变量是系统中部件的无故障工作时间 和故障后的修复时间。通常部件发生故障的时间分布和修复时间的分布用指数分布 来描述,也可以用 T 分布和威布尔分布。如果故障是完全随机的,则可以采用指数 分布建模。如果部件有储备,且每个备件的故障发生时间服从指数分布,则可以采 用 T 分布来建模。威布尔分布已经广泛用于描述故障发生时间,原因是它逼近许多 观察结果,当系统中有许多部件的故障是由子大量元件的严重失效或可能失效造成 时,适合采用威布尔分布建立模型。正态分布适用于那些大多数故障是由于磨损产 生的系统。

综上所述可以看出,在离散事件系统建模过程中,随机变量较常采用的分布形 式是指数分布和正态分布,其主要原因为:

指数分布简单,数据处理容易。

指数分布的一个重要特点是无记忆性,这与大多数与时间有关的随机现象是 一致的。例如在排队服务系统中,经过很长时间才有顾客到达,并不会对下一个顾 客到达的时间产生影响。同样,服务台不会因为已经工作了很长时间就会缩短为下 一个顾客的服务时间。又如电子元器件在工作了一段时间后继续使用,其在固定的 时间内发生故障的概率与一个新的元器件完全一样。

指数分布与许多其它的分布形式有关,一般作为特例,如泊松分布、威布尔 分布、T 分布等。作为指数分布的补充,这些分布形式也大量地应用在离散事件系



统的建模过程中。

正态分布虽然没有指数分布那么多的优点,但也是描述随机变量所必不可少 的一种分布形式。正态分布的特点之一是可以使随机变量取值相对集中或者使随机 变量散布很大甚至接近于均匀分布,这个特点使它具有很广的应用范围。

实际上,指数分布和正态分布属于完全不同的两种描述随机变量的分布形式。 服从指数分布的随机变量的取值概率是从大到小或从小到大(在负指数分布情况下); 而服从正态分布的随机变量概率分布有一个最大值,随机变量的大多数取值都在这 个最大值附近的区域之内。例如一个电子元件的平均寿命是 1000h ,如果它服从正 态分布,那么这个电子元件的实际寿命应当离 1 000h 不远(概率很大),但若它服从 指数分布,那么这个电子元件的实际寿命就可能与 1000h 相差很远。

在描述时间间隔这类随机变量时,如果没有任何限制,则可以用指数分布或指数分布的相关分布〔泊松分布、威布尔分布等〕来对此时间间隔进行描述。而如果时间间隔接近于一个常数或多数时间间隔都在一个固定的时间范围之内,那么这个随机变量最好用正态分布来描述。

选择描述离散事件系统中随机变量的分布形式,其原则是简单适用。在具体分 布形式选定之后,还需对其进行拟合度检验。

2.3 随机数和随机变量的产生

在许多仿真过程中,事件的发生是随机的,或者事件的属性值的确定具有偶然 性。例如,仿真中广泛应用的决策逻辑中,偶然性的影响和作用是非常大的,在一 个系统的运行中,已知按照一条给定的路径运行有一定的可能性。在对系统的仿真 中需要有一种方法能够选定这条路径,从而使仿真器的长期行为更加接近于真实系 统。由于在多数情况下这个决策是不确定的,对路径的选择通常要基于概率关系。

由此或其它一些原因,几乎在所有的仿真模型中都需要有某种装备用来产生随 机数。

这种装备必须能够给出一个特定的值使得在任何给定的仿真运行中,事件看起 来像是随机发生的。在这种情况下需要有一个决策模块,它必须能够指导仿真器在 某些运行中采用一条给定的路径,而在其它的运行中采用其它的路径。

由于这些随机因素的存在,在建模过程中就可能需要用服从各种分布的随机变 量来描述系统中存在的随机和偶然性问题。随机变量的产生基础是随机数,随机数 可以通过对[0,1]区间上均匀分布抽样生成。曾经使用过的真正随机数的发生源包括:

- 一袋小球,进行可替换抽样;
- 微秒级时钟上的低阶数字;
- 一个随机电子噪声源的周期性量化输出值。

这些方法都可以产生真正的随机数。但它们有一个共同的缺点,就是应用在仿 真研究中时,用这些方法产生的随机数序列通常不能复现。而在很多情况下,系统 决策需要进行多次仿真比较才能确定,因此上述重现性是非常必要的。



2.3.1 伪随机数

大量的方法被应用于克服随机数序列所固有的不可重现性。在讨论这些方法之前,首先研究对随机数发生器的要求:

所产生的数必须服从均匀分布。至于拟合的优良程度,建议采用大样本量的 χ^2 检验。样本容量 N 可以取 1000 至 10000 之间的数。

所产生的数必须是统计独立的。随机数序列中一个数的值不能影响下一个数 的值。如果随机数序列缺少独立性就可能被拒绝,但随机数序列被接受并不能证明 它的独立性。

所产生的随机数序列必须是可以重现的,这样就允许仿真试验重复进行。

所产生的随机数序列在任何需要的长度内必须是不重复的。这在理论上可能 是做不到,但从实践目的角度讲,在很多数目之后才出现重复性循环。这项要求已 经能充分保证。随机数发生器的重复性循环出现的长度称为它的周期。

随机数产生的速度必须快,因为在仿真运行中通常需要处理大量的随机数, 如果随机数发生器速度慢,就必然大大增加仿真运行的时间和费用。

用于产生随机数的方法应当占用尽可能小的存储空间,仿真模型通常需要大 量的存储空间,而存储空间总是有限的,如此宝贵的资源在产生随机数的过程中占 用得越少越好。

一个随机数序列, R_1 , R_2 , ... R_N 必须具备两个重要的性质。即均匀性和独立性。 所谓均匀性是指,如果把随机数的采样区间分成 n 个相等长度的子区间,那么在每 一个子区间上观察到的期望数为 N/n,其中 N 为观察总次数;独立性是指一个数值 出现在一个特定子区间的概率与前面已经取到的数值无关。

在最简单的情况下,一个随机数序列中的每一个随机数都是[0,1]]区间上均匀 分布的独立采样值。该分布的概率密度函数由下式给出:

$$p(x) = \begin{cases} 1 & 0 \le x \le 1 \\ 0 & \text{IT} \end{cases}$$

这样可以得到每一个随机数的期望值为:

$$\mathsf{E}(R) = \int_{v}^{1} x \mathrm{d}x = \frac{1}{2}$$

而方差为

$$V(R) = \int_{0}^{1} x^{2} dx - [E(R)]^{2} = \frac{1}{12}$$

根据对随机数发生器的要求,就可以评价为了补偿随机数序列所缺乏的重现性 而采用的一些方法。第一种方法是利用某种手段生成一个随机数序列并将它永久保



存起来。例如存在计算机硬盘上。这种方法并不能使人满意,因为每次对随机数的 需求都必须进行一次读操作,而这种操作是非常浪费时间的,而且这种方法为了满 足随机数重复周期的要求,必须存储一个很大的随机数序列。第二种方法是利用某 种手段生成一个随机数序列并将其保留在计算机内存中。这种方法可以克服第一种 方法中的速度问题,但是要满足多个仿真研究的需要就应存储一个相当大的随机数 序列,这就要求有非常大的内存量。第三种也是最常用的一种方法,是利用某种算 法根据指定的输人值来产生随机数。这种方法克服了速度和内存容量上的问题,但 同时又带来了在独立性和可重复性方面的问题。

运用某种算法产生随机数可能会破坏随机的基本性质,因此利用算法所产生的 随机数被称为伪随机数。这些数满足一定的随机性准则,但它们的产生总是以某个 称之为种子的确定的初始值开始,并且是一个完全确定的、重复的过程。当使用伪 随机序列时必须极其小心才能保证有良好的随机度以及重复周期才能足够地长。在 仿真研究中对利用算法产生的伪随机数序列进行合理性检验也是非常重要的。

注意伪随机数中的"伪"字。所谓的"伪"并不意味着产生了一个假的随机数 或随机数序列。这里使用"伪"字是指由一个已知的方法所产生的随机机数的真正 作用是排除了真

正的随机性。如果知道了产生随机数的方法那么任何一个随机数序列便可以重复地 产生出来,这样就会认为这些数不是真正的随机数。然而,不管采用什么方法,产 生随机数的目的是使所产生的随机数序列能尽可能地接近均匀分布和独立性的理想 性质。

当然,利用这种方法产生的伪随机数会与理想随机性产生一定的偏差。例如所 产生的数可能不服从均匀分布,数的均值太高或太低,数的方差可能太高或太低, 可能存在一定的周期性变化等。正因为如此,才需要对所产生的伪随机数序列进行 均匀性和独立性检验。如果检验出的偏差超出许可范围,就不能接受相应的伪随机 数产生方法。

2.3.2 伪随机数产生方法

2.3.2.1 平方取中法

平方取中法是冯·纽曼(John van Neumann)在 40 年代中期提出的。这个方法首 先从某个初始的种子数开始,求出这个数的平方。取这个平方数的中间几位作为随 机数序列中的第 2 个数;再求出第 2 个数的平方,又取这个平方数的中间几位作为 随机数序列中的第 3 个数;不断按这个方式继续此算法,即可得到相应的伪随机序 列。

例 1:利用平方取中法产生 4 位数的随机数序列,序列的种子数取为 x_0 = 3187, 通过计算得到

$$(3187)^2 = 10156969$$
; $x_1 = 1569$

2107

| (1569) ² =02461761; | x ₂ =4617 |
|---------------------------------|----------------------|
| (4617) ² =21316689; | x ₃ =3166 |
| (3166) ² =10023556 ; | x ₄ =0235 |
| $(0235)^2 = 00055225$; | x ₅ =0552 |
| (0552) ² =00304704 ; | x ₆ =3047 |
| (3047) ² =09284209; | x ₇ =2842 |

将此过程继续下去,还可以得到 0769,5913,9635,8332,4222,8542,...,

这个方法有几个缺点。首先,利用这个方法产生的伪随机数序列的重复周期通 常较短。第二,对于较长的伪随机数序列,利用这种方法可能无法通过随机性的统 计检验。第三,当在任何时候生成之后,其后产生的数都将为 0。如果这种现象在 一个较复杂的仿真研究过程中出现,它将会使仿真分析人员误入歧途。这种现象如 下例所示。

例 2:利用平方取中法产生两位数的随机数序列,种子数取为 $x_0=44$ 。通过计算 得到:

| | x ₀ =44 |
|---------------------------|--------------------|
| (44) ² =1936 ; | x ₁ =93 |
| (93) ² =8649 ; | x ₂ =64 |
| (64) ² =4096 ; | x ₃ =09 |
| (09) ² =0081 ; | x ₄ =08 |
| (08) ² =0064 ; | x ₅ =06 |
| (06) ² =0036 ; | x ₆ =03 |
| (03) ² =0009; | x ₇ =00 |
| $(00)^2 = 0000$; | $x_8 = 00$ |

这样,随机数就无法继续产生了。

利用平方取中法的另一个问题是这个方法可能产生退化,即总是得到相同的 x_i 值。

例 3:设在产生四位数的随机数过程中,得到了一个 x;值为 4500,即

$$x_i = 4500$$

从而, $(4500)^2=20250000$; $x_{i+1}=2500$

$$(2500)^2 = 06250000$$
; $x_{i+2} = 2500$

相继产生的数值总为 2500。

由于这些原因,平方取中法已经被许多新的能提供更可靠的随机数序列的算法 所取代。

2.3.2.2 线性同余法

线性同余法在 1951 年由菜默尔(Lehmer)首先提出。目前大多数随机数发生器都 采用这种方法。在这个算法中,随机救序列中的数由如下的递推关系产生

$$x_{n+1} = (ax_n + c) \operatorname{mod} m \quad n \quad 0 \tag{1}$$



初始值 x₀称为种子,常数 a 称为乘子,常数 c 称为增量,而常数 m 称为模数。 在(1)式中,显然可以得出 0 x_n m-1 (2)

为了得到[0,1]区间上分布的随机数,可以令

$$R_n = \frac{x_n}{m} \tag{3}$$

Rn 为满足要求的随机数。

从(1)式中得到的序列 0, 1...实质上完全不是随机的,因为若设

$$k_i = (ax_{i-1} + c) \operatorname{mod} m \tag{4}$$

则有

$$x_{1} = ax_{0} + c - mk_{1}$$

$$x_{2} = a^{2}x_{0} + ac - mak_{1} + c - mk_{2}$$

$$= a^{2}x_{0} + c(1 + a) - m(ak_{1} + k_{2})$$

$$x_n = a^n x_0 + c(1 + a + \Lambda + a^{n-1})$$

$$-m(k_n+k_{n-1}a+\Lambda+k_1a^{n-1})$$

$$= [a^n x_0 + c \frac{a^n - 1}{a - 1}] \operatorname{mod} m$$

即一旦 m,a,c,x_0 确定, x_i 就完全被唯一地确定下来了。而且在[0,1]区间上得到的数值 Rn 最多只能取到 m 个值。

常数 $m_{a,c}$ 的值对所产生的随机数序列的周期长度有很大的影响。 例 4:设 a=5, c=3, m=15, 取 $x_0=7$, 利用线性同余法产生随机数序列。

v -7



| $x_9 = (5*15+3) \mod 16 = 14$ | R ₉ =0.875 |
|----------------------------------|------------------------|
| x ₁₀ =(5*14+3)mod16=9 | R ₁₀ =0.563 |
| $x_{11} = (5*9+3) \mod 16 = 0$ | $R_{11}=0.000$ |
| $x_{12} = (5*0+3) \mod 16 = 3$ | R ₁₂ =0.188 |
| $x_{13} = (5*3+3) \mod 16 = 2$ | R ₁₃ =0.125 |
| x ₁₄ =(5*2+3)mod16=13 | R ₁₄ =0.813 |
| $x_{15} = (5*13+3) \mod 16 = 4$ | R ₁₅ =0.250 |
| $x_{16} = (5*4+3) \mod 16 = 7$ | R ₁₆ =0.438 |
| x ₁₇ =(5*7+3)mod16=6 | R ₁₇ =0.375 |
| $x_{18} = (5*6+3) \mod 16 = 1$ | R ₁₈ =0.063 |
| $x_{19} = (5*1+3) \mod 16 = 8$ | $R_{19}=0.500$ |

在 n=16 时出现循环。

对于(1)式,当 c=0 时,该算法称为乘同余法;当 c 0 时,该算法称为混合同余法。从例 4 中可以看出利用这种方法产生的序列的直复性,一般来讲任何由此方法 产生的序列都存在重复性。在大多数情况下,合理地选择常数 a,c,x₀和 m,可以使重 复周期充分的长。对于常数的选择讨论如下:

m 的选择

由于重复周期的长度总是小于 m,因此需要将 m 取大的数值,更进一步,所选 用的 m 的值应能简化同余关系的解,在计算机中数字都是用二进制表达的,因此已 经证明 m 取值为(2^k-1)是很好的,其中 k 为计算机的字长。

a和 c 的选取

当且仅当下列条件满足时,一个由线性同余法产生的随机数序列的重复周期为 m。

● c与m互质,即同时能被c和m整除的正整数只有1。

● (a-1)是每个能整除 m 的质数的倍数。即如果 q 是整除 m 的质数 , 则 q 能整 除 (a-1)。

● 如果 m 能被 4 整除,则(a-1)也能被 4 整除。

这些限制使得乘子值满足形式为 $a=x^{p}+1$,其中 z 是计算机中用于表示数字的基数;模数的形式为 $m=z^{k}$, k 为计算机的字长、且有 z p < k。特别地,当选择 $a=2^{16}+5=65541$ 或 $a=2^{16}+3=65539$ 时可以得到满意的结果。至子 c 的选择,只要满足 c 与 m 互为质数的条件即可。

x0 的选取

如果随机数序列的周期为 m,因为能产生完全的序列,即在一个周期内可以取到 0 至(m-1)的所有值,因此 x_0 的选取是不重要的。但仍然要小心,例如取 $x_0=0$ 时 会产生退化的序列。

对于乘同余法,由于 c=O,无论怎样选择 m,都无法满足 c 与 m 互质的条件, 因而不可能得到满周期。若选择 m=2^k,则所产生的随机数序列的周期 p 2^{k-z} ,即在 0 至 m-1 之间的整数至多只有四分之一可能成为 x_n 的值,而且这四分之一的整数在 0



至(m-1)之间是如何分布的尚难确定。这与种子数 x_0 的选取有关。若取乘子为 a=8L + 3 或 a=8L+形式的整数。种子 x_0 取为奇数.则可以达到最长的周期 $p=2^{k-2}$ 。

2.3.2.3 加同余法

加同余法需要 n 个数的序列作为它的种子,这 n 个数的序列可以应用其它的方 法产生,应用加同余法可以使这个序列不断扩大,加同余法的算法是

$$x_{i} = (x_{i-1} + x_{i-n}) \mod m$$
 (5)

这种方法的主要好处是速度快,因为它不需要作乘法运算。它可以得到大于 m 的周期但这种方法产生随机数的过程不象混合同余法那样清楚,因此由此方法产生 的随机数序列需要经过仔细的确认。

2.3.2.4 二次平方同余法

二次平方同余法适用于 m 为 2 的幂次的情况,这种方法的递推关系式为

$$x_{n+1} = [x_n(x_n+1)] \mod m \quad n \ge 0$$
(6)

种子数 x₀必须满足条件

 $x_0 \mod 4 = 2$

2.4 WITNESS 系统标准随机分布函数

为了方便用户构建随机仿真模型,WITNESS 提供了 14 种整型或实数型的标准随 机分布函数,它们能返回一系列理论分布的随机样本值。WTNESS 选择这些分布是 因为这些理论分布已经在相当长时间内广泛的研究过了,而且也是在仿真中被认为 是最有效的。

2.4.1 伪随机数流 (PRNS: pseudo-random number stream)

WITNESS 不是储存了大量的预先定义的随机数,而是储备了 1,000 个不同的数 列或者叫伪随机数流。我们在使用一个标准分布时,必须输入一个伪随机数流和相 关的变量。

例如,一台机器组装零部件的时间服从泊松分布,均值为5分钟,我们将PRN1 用到这个分布中,即 POISSON(5,1),则在仿真过程中,系统将使用它储备的1000 个伪随机流中的第一个数列作为种子数,依次生成服从均值为5的泊松分布随机数 流。这样在同一次仿真过程中,机器每次组装零部件的时间是随机的,而在不同的 仿真运行中 WITNESS 产生的组装时间随机数流则是相同的。

由于伪随机流的存在,我们可以改变模型当中的一个参数值,保持其它参数不 变,重新运行模型,比较仿真运行的结果,就可以精确的了解该因素对仿真系统绩



效的敏感性。

WITNESS 提供了如下 14 种标准随机分布函数:

- (1) BETA 分布; (2) NORMAL 正态分布;
 (3) BINOMIAL 二项分布; (4) POISSON 泊松分布;
 (5) ERLANG 爱尔朗分布; (6) RANDOM 0-1 均匀分布;
 (7) GAMMA 分布; (8) TNORMAL 截断正态分布;
- (9) IUNIFORM 整数均匀分布; (10) TRIANGLE 三角分布;
- (11) LOGNORML 对数正态分布; (12) UNIFORM 均匀分布;
- (13) NEGEXP 负指数分布; (14) WEIBULL 威伯分布

我们将从如下五方面来介绍以上的随机分布函数:

- 函数的名称与函数的简介;
- 函数语法结构及其参数;所有函数的最后一个参数都是伪随机数流;
- 分布曲线(曲线图中,横轴表示函数的返回值,纵轴表示对应返回值的发生概率);
- 随机分布函数调用示例;
- 适用情况。

2.4.2 随机分布函数详解

(1) BETA(分布)

该函数提供服从 分布的随机样本值,返回值为实数,通常应用在实际系统的 数据有限,数据变化范围大的情况。

语法结构:

```
BETA ( shape , scale , prns )
```

参数:

shape: 形状参数, 实数;

scale:比例参数,实数;

prns:伪随机数流,整数。

分布曲线 (横轴表示可能取值 , 纵轴表示对应取值的概率):

当 shape = 1.5, scale = 5 时, 分布曲线如下:



当 shape = 5, scale = 1.5 时, 分布曲线如下:





函数调用示例:

X = BETA(1.5, 5.0, 1)

```
X = BETA(5.0, 1.5, 2)
```

适用情况:

产品的次品率;

工作的完成时间。

(2) NORMAL(正态分布)

该函数提供服从正态分布的样本值,返回值为实数。该函数是应用最为广泛的 一种分布,分布曲线关于均值对称的,经常运用于工作时间围绕均值波动的随机情 况。

语法结构:

```
NORMAL (Mean, SD, prns)
```

参数:

Mean: 分布均值, 实数;

SD:标准差,实数;

Prns:为随机数流,整数。

分布曲线:

分布曲线如下图:



函数调用示例:

X = NORMAL(10.0, 3.0, 1)

X = NORMAL(5.0,2,2)

适用情况:

机器加工时间等。

注:

从图上可以看出,有 95%的随机数据落在两倍标准差之内;如果标准差的数值 越小,则数据对均值的波动性就越小。

正态分布可能会产生负的值,所以在使用它作为时间间隔的随机变量时,要特别小心。


(3) BINOMIAL (二项分布)

该函数提供服从二项分布的样本值,返回值为整数。在给定的成功几率和试验 次数条件下,该函数返回成功的次数。例如,特定供应商提供的发动机次品率为10%, 可以使用二项分布来获得批量为5的n批发动机中每批的次品数,有时是1个,有 时是2个.....,如下面的第一个图形所示。

语法结构:

BINOMIAL (prob, trials, prns) 参数:

prob: 几率, [0, 1]之间的实数;

trials:试验次数或批量,整数;

Prns:为随机数流,整数。

分布曲线:

当 prob=0.1, trials=5 时, 分布曲线如下:



当 prob=0.5, trials=5 时, 分布曲线如下:



函数调用示例:

J = BINOMIAL(0.1, 5, 1)

J = BINOMIAL(0.1, 10, 1)

J = BINOMIAL(0.5, 5, 1)

J = BINOMIAL(0.5, 10, 1)

适用情况:

指定尺寸的一批货物当中的次品数目;

仓库中需要的货物的品种数量。

(4) POISSON (泊松分布)

该函数提供服从泊松分布的样本值,返回值为整数。通常情况下,使用该函数 来生成在给定的时间段内顾客或部件的到达数量,可以使它为负指数分布的一种补 充。



语法结构:

POISSON (MEAN, prns)

参数:

MEAN:均值,实数;

Prns:为随机数流,整数。

分布曲线:

当 MEAN=0.5、MEAN=0.5 时,分布曲线如下:





mean=0.5

mean=1

当 MEAN=2、MEAN=6 时,分布曲线如下:



mean=2



mean=6

函数调用示例:

J = POISSON(0.5,1)

```
J = POISSON(1.0,2)
```

J = POISSON(2.0,3)

J = POISSON(6.0,4)

适用情况:

零件到达的随机批量;

生产机器的单位时间产出数量。

(5) ERLANG (爱尔朗分布)

该函数提供服从 K 阶爱尔朗分布的样本值,返回值为实数。爱尔朗分布是一个 分布函数族:其分布曲线随着 K 值的不同而有所不同。

当 K=1 时,爱尔朗分布同负指数分布相似,因为爱尔朗分布是 K 个具有相同均 值的负指数分布的样本值之和;

当 K=2 时,爱尔朗分布的形状象钟铃形,极度向左倾斜,同对数正态分布的分

W

布形态相似;

当 K 取大于的值时,爱尔朗分布又同正态分布相似,但是,同对数正态分布和 正态分布又有所不同,爱尔朗分布仅同均值有关,而同标准差无关;

可以通过改变爱尔朗分布中的 K 值来进行灵敏度分析,例如检测物料短缺的影响。

语法结构:

ERLANG(M, K, prns)

参数:

M:均值,实数; K:K值,整数;

Prns:为随机数流,整数。

分布曲线:

当均值 M=1, 而 K 值分别为 1, 2, 3 时, 函数的分布曲线如下:



m=1, k=3

函数调用示例:

R = ERLANG(1.0, 1, 1)

R = ERLANG(1.0,2,2)

R = ERLANG(1.0,3,3)

适用情况:

完成一项服务所需的时间,例如:完成一名顾客的服务时间或修理好一台机器



的时间。

(6) RANDOM (0-1 均匀分布)

该函数提供服从 0-1 均匀分布的样本值,返回值为[0,1]之间的实数,返回 0于 1 之间任意小数的概率是相同的。

语法结构:

RANDOM (Prns)

参数:

Prns:为随机数流,整数。

分布曲线:

分布曲线如下图:



函数调用示例:

R = RANDOM(1)

适用情况:

使用此函数作为我们自定义随机分布函数中的随机种子数。

(7) GAMMA(分布)

该函数提供服从 分布的样本值,返回值为实数。

语法规则:

GAMMA(Shape, Scale, prns)

参数:

Shape:形状参数,实数;

Scale:比例参数,实数;

Prns:为随机数流,整数。

分布曲线:



shape=2, scale=1

函数调用示例:

R = GAMMA(2.0, 1.0, 42)

Ŵ

适用情况:

机器出故障的时间间隔。

(8) TNORMAL (截断正态分布)

该函数提供服从截断正态分布的样本值,返回值为实数。它同正态分布及其相 似,不同之处在于它指定了样本值的最大值和最小值,而正态分布的最大值和最小 值为无穷大。

语法结构:

TNORMAL(Mean, SD, Min, Max, prns)

参数:

Mean:均值,实数; SD:标准差,实数; Min:最小值,实数; Max:最大值,实数; Prns:为随机数流,整数。 分布曲线:

在制定参数 mean=0.0, SD=6.0, Min=-4.0, Max=+4.0 时,其分布曲线如下:



mean=0.0, SD=6.0, Min=-4.0, Max=+4.0

函数调用示例:

R = TNORMAL(0.0, 6.0, -4.0, 4.0, 43)

适用情况:

一些服从正态分布的随机变量,但是很明显,它又不会出现在无穷大于无穷小处, 例如服务时间不可能为负值。

(9) IUNIFORM (整数均匀分布)

该函数提供服从整数均匀分布的样本值,返回值为整数。可以用来表示从指定 范围内,等概率获取整数的情况。

语法结构:

```
IUNIFORM (min, max, prns)
参数:
```

min:最小值,整数;



max:最大值,整数;

Prns:为随机数流,整数。

分布曲线:

当 min=0, max=10 时, 分布曲线如下:



min=0, max=10

函数调用示例:

```
R = IUNIFORM(1,4,3)
```

适用情况:

当仅仅知道某一变量在两个整数之间取值,而对其他情况一无所知时,首选的分 布就是整数均匀分布函数。

(10) TRIANGLE (三角分布)

该函数提供服从三角分布的样本值,返回值为实数。

语法结构:

```
TRIANGLE(min, mode, max, prns)
```

参数:

min:最小值,实数; mode:最可能发生值,实数;

max:最大值,实数;

Prns:为随机数流,整数。

分布曲线:

如该函数的名称所说,其分布曲线成三角形,分布曲线如下:



函数调用示例:

R = TRIANGLE(1.0,1.8,4.0,1) 适用情况:



当采用统计方法不能够对数据拟合成特定的概率密度函数,而其取值范围和取值 密集点能够确定时,采用三角分布函数。

(11) LOGNORML (对数正态分布)

该函数提供服从对数正态分布的样本值,返回值为实数。如果某一变量的样本 数据的对数服从正态分布,那么该变量就是服从对数正态分布。

语法结构:

LOGNORMAL (Mean, SD, prns)

参数:

Mean: 分布均值, 实数;

SD:标准差,实数;

Prns:为随机数流,整数。

分布曲线:

当 mean=1.65, SD=2.16 时, 分布曲线如下图:



mean=1.65, SD=2.16

函数调用示例:

R = LOGNORML(1.65, 2.16, 1)

适用情况:

完成一项服务所需的时间,例如:完成一名顾客的服务时间或修理好一台机器 的时间。

注:

该函数不会产生负值,其均值也不可能等于零

在 WITNESS 钟,对数正态分布函数的参数与其种子函数正态分布的参数无关, 至于样本值有关。

(12) UNIFORM (均匀分布)

该函数提供服从均匀分布的样本值,返回值为实数。用于等概率获得指定范围 内的数值。

语法结构:

UNIFORM (min, max, prns)

参数:

min:最小值,实数; max:最大值,实数;



Prns:为随机数流,整数。

分布曲线:



函数调用示例:

```
R = UNIFORM(3.0, 8.0, 1)
```

适用情况:

对某一变量的数据知之甚少,并且希望获得特定范围内的实数值时,就采用该函数。

(13) NEGEXP(负指数分布)

该函数提供服从负指数分布的样本值,返回值为实数。可以认为它是泊松分布 的补集。

语法结构:

```
NEGEXP (mean, prns)
```

参数:

mean:均值,实数;

Prns:为随机数流,整数。

分布曲线:

当 mean=1 时,分布曲线如下:



mean=1

函数调用示例:

R = NEGEXP(1.0,1)

适用情况:

相邻事件发生的时间间隔,例如:顾客到达时间间隔,及其故障时间间隔。 (14)WEIBULL(威布尔分布)

该函数提供服从威布尔分布的样本值,返回值为实数。根据指定的形状和比例 参数返回随机样本值。



语法结构: WEIBULL (shape, scale, prns)

参数:

shape:形状参数,实数; scale:比例参数,实数; Prns:为随机数流,整数。

分布曲线:

当 shape=2, scale=1 时,分布曲线如下:



shape=2, scale=1

函数调用示例:

R = WEIBULL(2.0,1.0,43) 适用情况:

可靠性建模。



第三章 Witness 仿真系统软件综述

Witness 是由英国 lanner 公司推出的功能强大的仿真软件系统。它可以用于离散 时间系统的仿真,同时又可以用于连续流体(如液压、化工、水力)系统的仿真。 目前已被成功运用于国际 3000 多家知名企业的解决方案项目,如 Airbus 公司的机场 设施布局优化、BAA 公司的机场物流规划、BAE SYSTEMS 电气公司的流程改善、 Exxon 化学公司的供应链物流系统规划、Ford 汽车公司的工厂布局优化和发动机生 产线优化、Trebor Bassett 公司的分销物流系统规划等。Lanner 公司已经在包括澳大 利亚、巴西、法国、德国、中国、意大利、日本、韩国、南非、美国、英国等 25 个 国家和地区设立的代理,负责软件的推广和技术支持等工作。

3.1 WITNESS 的安装与启动

本节主要介绍 WITNESS2003 系统的安装,包括安装前的准备和安装过程,并 介绍 WITNESS 启动与退出的具体操作方法。

3.1.1 安装环境

在安装 WITNESS2003 之前,要了解 WITNESS2003 的硬件和软件必备环境,做 好安装前的准备工作。

WITNESS 对硬件环境的一般要求并不高,但是要想获得较快的设计和仿真运行 速度、良好的可视化效果,推荐 CPU: P4 1.5G 以上、内存 512MB 及以上、独立显 卡 64M 以上显存。

WITNESS2003 系统运行需要 Window98、Windows2000、Windows NT 以及 Windows XP 的操作系统支持。

3.1.2 安装步骤

WITNESS 软件具有单机版和网络版两种版本,安装过程有所差异。本文介绍 WITNESS2003 单机版的安装过程。

(1)将 witness2003 系统光盘插入 CD-ROM 驱动器中,启动安装程序,进入安装语言选择对话框。如图 3.1 所示



| Select Language 🔀 | | | | | | |
|---|--|--|--|--|--|--|
| Please select the language that you would like to use during the installation. | | | | | | |
| Die Installation unters贖zt verschiedene Sprachen. W鋒len Sie eine Sprache aus der unten aufgef藁rten Liste aus | | | | | | |
| Veuillez 朝 ectionner une langue pour l'installation | | | | | | |
| English | | | | | | |
| Deutsch Fran鋃脯 is | | | | | | |
| OK Cancel | | | | | | |

图 3.1 witness2003 安装语言选择对话框

选择"English", 点击 OK 按钮。进入 WITNESS 版本选择对话框, 如图 3.2 所示。

| Select version | × |
|---|-----------------------|
| Please select the WITNESS install. | i version you wish to |
| Manufacturing Service (General Terminolo | gy) |
| <u>o</u> k | <u>C</u> ancel |

图 3.2 WITNESS 版本选择对话框

选择 Manufacturing (制造),单击 OK 按钮。然后选择安装路径,选择我们购买的安装组件,一路"NEXT"下去,等待软件安装到硬盘。在安装过程中,系统弹出 "Security Select"对话框,要求选择授权方式。对话框如图 3.3 所示。

| × |
|--------|
| ing |
| |
| |
| |
| Cancel |
| |

图 3.3 WITNESS 授权方式选择对话框



选择第一项,采用加密狗进行加密授权,单击 OK 按钮继续安装,直至程序安装完毕。

3.1.3 启动 WITNESS 2003

程序安装完成之后,将加密狗安装在计算机的并口上,然后通过选择 windows 系统的"开始"->"程序"-"Witness 2003"-"Witness 2003",如图 3.4 所示,来打 开 Witness 2003 系统。

| Γ | * | Windows Update | | | | | | | |
|----------|------------|-------------------|-----|---|---------------------------|---|-----|---------------------------------|---|
| | 9 | 说定程序访问和默认值 | | | | | | | |
| | a | 打开 Office 立档 | | | | | | | |
| | • | 新建 Office 立挡 | | | | | | | |
| | 0 | 启动 RealOne Player | | | | | | | |
| B | Ē | 程序(2) | • | 9 | TTOD | ٠ | | | |
| 5 | - | | | - | 附件 | • | - | Documentar | |
| 1.22 | | 文档(12) | • | - | 华泰证券 | • | - | Optimizer | |
| ŝ | 5 | 祝董(S) | | 뵘 | Microsoft Excel | | | VR. | 1 |
| Æ | - | | | 쀵 | Pecrosoft word | | 25 | Uninitial W11NESS 2005 | |
| 8 | A | 教察(量) | • | 뉟 | Plash Catcher | 1 | | WITNESS 2003 | |
| 8 | ۲ | 帮助(出) | | | LEte Faca Witness 2003 | | 1 B | WITNESS Help WITNESS XA 1.60 | |
| - No | 20 | 运行(E) | | _ | ŧ | | Γ | | |
| ž | b) | 关机(山) | | | | | | | |
| 1 | 开蜎 | 🧔 🗳 🎑 🗿 📃 🖻 | 144 | | ອງພ ອງພ | | 13 | z 🔄 w 🤌 w (| - |

图 3.4 Witness 2003 打开步骤

第一次使用 WITNESS 时,如果加密狗正确安装在计算机的并口,则 WITNESS 将会弹出如图 3.5 所示的对话框,让用户进行许可证号的输入。

| WITNESS | | × |
|--|---|-----------|
| WITNESS 2003 | | |
| No license is availa | ble that matches the current | |
| The license administ | ration dialog allows you to enter license | codes, or |
| A license request for be e-mailed directly | rm prompts for all the required information to Lanner or printed for faxing. | n and can |
| The security device automatically added security device pres | contains a unique number which will be to the form. Licenses will be issued to ma ent when the application was started. | tch the |
| | | |
| Quit | License Administration | |



3.5 许可证管理对话框

3.2 WITNESS 2003 用户界面

当用户对 WITNESS 系统有了初步的了解,并且掌握了 WITNESS 系统的安装和 启动后,用户最关心的一定是如何使用 WITNESS 系统。为了顺利地使用 WITNESS 系统,在这一节将介绍 WITNESS 的窗口环境,并介绍 WITNESS 窗口的基本操作。

当正常启动 WITNESS 系统后,首先进入的是 WITNESS 系统的主屏幕界面。如 图 3.7 所示。



图 3.7 WITNESS 系统主屏幕界面

从图上可以看出,WITNESS 系统的界面,是由标题栏、菜单栏、工具栏、元素 选择窗口、状态栏、用户元素窗口和系统布局区组成的。下面对每一部分的功能加 以介绍。

3.2.1 标题栏

标题栏位于屏幕界面的第一行,它包括系统程序图标、主屏幕标题、最小化按钮、最大化按钮和关闭按钮 5 个对象。

(1)系统程序图标 单击 WITNESS 系统程序图标,可以打开窗口控制菜单,在窗口控制菜单下,可以移动屏幕并改变屏幕的大小。双击系统程序图标,可以关闭WITNESS 系统。

(2) 主屏幕标题 主屏幕标题由两部分组成,前一部分是系统的名称,也既是 "WITNESS",后一部分是当前打开的模型的标题,可以根据不同的模型进行修改 设置。设置方法是打开"Model"菜单,选择"Title"选项,在弹出的标题设置对话 框中进行设置。标题对话框如图 3.8 所示。

在模型标题设置对话框的"General"页框下,设置模型的名称(name)标题 (Title)作者(Author)等信息。模型名称和标题将显示在主屏幕标题栏上。 (3)最小化按钮 单击"最小化"按钮,可将系统的屏幕缩小成图标,并存放在

(3) 最小化按钮 单击 最小化 按钮,可将系统的屏幕缩小成图称,并存放在 Windows 桌面的底部的任务栏中。(4)最大化按钮 单击"最大化"按钮,可将系统 的屏幕定义为最大窗口。(5)关闭按钮 单击"关闭"按钮,可将关闭 WITNESS 系统。

| Model Title |
|---|
| General Notes |
| Model |
| <u>N</u> ame : |
| STARTUP |
| <u>T</u> itle : |
| Base Model. |
| Author |
| Lanner Group. |
| Creation Mon May 31 20:14:02 2004:WITNESS V715 3.18 |
| |

图 3.8 模型标题设置对话框

3.2.2 菜单栏

菜单栏位于屏幕的第二行,它包含:File(文件) Edit(编辑) View(显示) Model(模型) Elements(元素) Reports(报表) Run(运行) Window(窗口) Help(帮助)九个菜单选项。当单击其中一个菜单选项时,就可以打开一个对应的"下 拉式"菜单,在该"下拉式"菜单下,通常还有若干个子菜单选项,当选择其中一个 子菜单选项时,就可以执行一个操作。

3.2.3 工具栏

Witness 系统提供了不同环境下的八种常用的工具栏,它们是:Standard、Model、 Element、Views、Run、Reporting、Assistant、Display Edit。激活其中一个工具栏,



即在屏幕上显示出一行相应的工具栏,用鼠标将它拖放到合适的位置,就可以使用这个工具栏提供的相应的工具进行某些操作。

激活工具栏使用菜单 View/Toolbars, 然后选中相应的菜单即可。

3.2.4 元素选择窗口

在元素选择窗口中,有五项内容:Simulation、Designer、System、Type、System Function。其中 Simulation 中将显示当前建立的模型中的所有元素列表;Designer 中 显示当前 Designer Elements 中的所有元素列表;System 中显示系统默认的特殊地点; Type 中显示 Witness 系统中可以定义的所有元素类型;System Function 中显示 Witness 系统中可以定义的所有函数类型。

该窗口的显示和隐藏可以使用菜单 View/Element Selector,或者使用 element 工

3.2.5 状态栏

状态栏位于屏幕的最底部,用于显示某一时刻的工作状态或者鼠标光标位置的工 具栏按钮的作用。

3.2.6 用户元素窗口 (Designer Elements)

系统提供的默认用户元素窗口中提供了各种元素的可视化效果的定义,不过在 建模过程中,当这些缺省设置并不能很好的表示实际系统,用户可以在该窗口定义 自己的相关元素 的名称、可视效果等,保存以便日后的使用。定义方法可以鼠标右 击页框标题,将出现弹出式菜单,其中具有菜单项"Add New Designer Group"、

"Rename Designer Group"、"Delete Designer Group"、"Load Designer Group",可以 进行添加新页框、重命名本页框、删除本页框、加载原有设计元素组。向页框中添 加自定义元素的步骤一般也分为 Define、Display、Detail 三步。页框的背景色设置 同系统布局窗口背景色的设置。自定义元素设定完毕之后,需要保存成*.des 文件, 通过菜单 File/Save As,然后选定文件类型为 Designer Element Files(*.des),输入文件 名即可。

3.2.7 系统布局区

系统布局区也叫系统布局窗口,在布局窗口中,设置实际系统构成元素的可视 化效果以及它们的二维相对位置,可以清楚的显示实际系统的平面布局图。Witness 一共提供了八个窗口,可以通过这些窗口,使得仿真项目以不同的角度显示其可视 化效果。



对系统布局窗口的设置主要有三项内容:添加元素、设置窗口名称以及窗口背 景色。如何添加元素将在后面本章最后一部分内容;设置窗口名称以及窗口背景色 可以通过选择菜单项 Window/Control...,将弹出如图 3.9 所示的窗口。

| 🥪 Window Control | | × |
|--|-----------------------|------------------|
| <u>N</u> ame: | | |
| Window 2 | | OK |
| Background Color: | <u>Z</u> oom: 55% | Cancel |
| | Reveal All | <u>H</u> elp |
| | Movable Display | |
| Included | <u>E</u> xcluded | |
| Layer 9 Layer 8 Layer 7 Layer 6 Layer 5 Layer 4 | | |
| $E_{\underline{x}}$ clude $>>$ | << | In <u>c</u> lude |

图 3.9 窗口控制对话框

在 Name 下的文本框中输入窗口的名称;点击 Background Color 下的颜色按钮, 在弹出的调色板中选定背景颜色;选择 Zoom 中的比例可以放大或缩小布局窗口中 元素的尺寸。

3.3 WITNESS 建模元素

现实的商务或事物系统总是有一系列相互关联的部分组成的,比如制造系统中 的原材料、机器设备、仓库、运输工具、人员、加工路线或运输路线等,服务系统 中的顾客、服务台、服务路线等。Witness 软件使用与现实系统相同的事物组成相应 的模型,通过运行一定的时间来模拟系统的绩效。模型中的每个部件被称之为"元 素(Element)"。该仿真软件主要通过如下五类元素来构建现实系统的仿真模型:离 散型元素、连续型元素、运输逻辑型元素、逻辑型元素、图形元素。

3.3.1 离散型元素

表示所要研究的现实系统中可以看得见的、可以计量个数的物体,一般用来构 建制造系统和服务系统等。主要包括:

零部件或实体 (Part or Entitie); 机器 (Machine); 输送链 (Conveyor); 缓冲区 或仓库 (Buffer); 车辆 (Vehicle); 轨道 (Track); 劳动者 (Labor); 路径 (Path); 模块 (Module)



3.3.2 连续型元素

同离散型元素相对应,这种类型的元素用来表示加工或服务对象是流体的系统, 比如化工、饮料等。主要包括:

流体 (Fluid); 管道 (Pipe); 处理器 (Processor); 容器 (Tank)

3.3.3 运输逻辑型元素

用于构建物料运输系统。主要包括:

运输网络(Network);单件运输小车(Carriers);路线集(Section);车辆站点(Station)

3.3.4 逻辑元素

用来处理数据、定制报表、建立复杂逻辑结构的元素,通过这些元素可以提高 模型的质量和实现对具有复杂结构的系统的建模。主要包括:

属性 (Attribute); 变量 (Variable); 分布 (Distribution); 函数 (Function); 文件 (File); 零部件文件 (Part file); 班次 (Shift);

3.3.5 图形元素

图形元素可以将模型的运行绩效指标在仿真窗口形象的表现出来。主要包括: 时间序列图 (Timeseries); 饼状图 (Pie chart); 直方图 (Histogram)。

3.4 WITNESS 建模与仿真过程

使用 witness 软件包进行物流与供应链系统的建模与仿真,同样要遵循建模与仿 真的一般步骤。在适用它进行计算机模型的建立时,有其特定的步骤:

(1)定义系统元素 可以通过在布置窗口中点鼠标右键,选定快捷菜单中的"define"菜单项,来定义模型基本元素的名称、类型、数量;

(2)显示系统元素 witness 软件是一套优秀的可视化建模与仿真工具,它可 以将被仿真系统的可视实体以二维或三维的图形显示出来;在仿真运行时,它可以 显示原材料、零部件、人员、运输车辆在系统中的运动状况。所以在定义了元素的 基础上,要定义元素在各种状态下的现实图形。本步骤可以通过右击要定义显示特 征的元素,通过选定弹出式菜单中的"display"菜单项,来进行设定。各种元素的 平面布置可以在 witness 的布置窗口中设定,也可以通过导入被仿真系统设施布置图 的.dwg 文件来设定。

(3) 详细定义 本步骤详细定义模型基本元素工作参数以及各元素之间的逻辑



关系,如系统结构、被加工对象在各台机器上的加工时间分布、加工对象的工艺路 线、以及其他规则等。可以双击鼠标左键,通过弹出的 " detail " 对话框来设定。

(4)运行 通过试运行和修改模型,重复前三步得到正确的计算机仿真模型 之后,对系统进行一定时间范围的运行,并在屏幕上动画显示系统运行的过程,运 行方式可以是单步的、连续的和设定时间的。本步骤通过 witness 提供的 "run"工 具栏来进行操作。

(5)报告 系统运行一段时间后,显示系统中各元素的运行状态统计报告。 通过该报告,可以分析系统中可能存在的各种问题;或通过某项指标,来比较可选 方案的优缺点。如机器的利用率、产品的通过时间、在制品库存等。该操作通过使用"reporting"工具栏来实现。

(6) 归档 witness 还提供了归档 "documentor"模块,可以让我们提取计算机 模型的各种信息,生成 word 文档或直接打印出来。主要是生产报告模块没有包含的 有关元素的说明型文字、规则、活动、中断和基本信息。

(7)优化 witness 还提供了系统优化 "optimizer"模块。如果一个系统的绩效将因为其构成元素的配置不同,而得到不同的结果,并不需要建立多种配置的计算机模型。我们可以直接使用同一个计算机模型,然后通过 "optimizer"模块来设定每一元素的可变属性值的取值范围,得到一个取值范围集合,并设定表示绩效的目标函数是取最大值还是最小值,进行优化仿真运行,就可以得到前 n 个最优绩效的系统配置(n 可自行设定)。

3.5 WITNESS 建模与仿真过程应用举例

在实例模型中,小部件(widget)要经过称重、冲洗、加工和检测等操作。执 行完每一步操作后小部件通过充当运输工具和缓存器的传送带(conveyer)传送至下 一个操作单元。小部件在经过最后一道工序"检测"以后,脱离本模型系统。仿真 一周,试求该系统的产出量和各设备的利用率。

本模型系统的流程视图如图 3.10 所示。





模型元素说明:widget 为加工的小部件名称;weigh、wash、produce、inspect 为四种加工机器,每种机器只有一台;C1、C2、C3为三条输送链;ship 是系统提供的特殊区域,表示本仿真系统之外的某个地方;



该模型设计的具体操作步骤与方法请参看第六章,下面简单介绍该模型的建立 步骤。

(1) 定义元素:通过 define 菜单,分别定义各个元素名称、类型、数量,如表 3.1 所示;

表 3.1 元素名称、类型、数量信息表

| 名称 | Widget | Weigh | Wash | Produce | Inspect | C1 | C2 | C3 |
|----|--------|---------|---------|---------|---------|----------|----------|----------|
| 类型 | Part | Machine | Machine | Machine | Machine | Conveyor | Conveyor | conveyor |
| 数量 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(2)显示元素:由于是动画仿真系统,因此对系统的每一个对象要进行可视化定义。 系统提供了图形库和颜色集,用户可以根据自己的想象选择合适的图形和颜色;同时,用户还可以自己绘制系统中各个对象的图形。本步骤需要选定各个对象,然后 可视化对象并进行定位。

(3) 详细定义:本步骤要分别定义每个元素 detail 对话框中的参数。

机器的详细定义:本模型中对机器主要定义它们的机器类型、加工周期、进入 规则和送出规则。

Witness 提供了的机器类型有:single(单流程) batch(批处理) assembly(组装) production(生产) general(通用) multicycle(多周期) multistation(多站 点)这七种。通过它们,可以很容易的对实际使用的机器建立模型。本例中四种机器类型都是单流程的。

加工周期 (Cycle Time): 加工周期可以是固定的时间值,也可以是随机分布函数或表达式。本例中四种机器的加工周期都采用固定时间值。

进入规则(Input Rule):进入规则有主动 (push) 和被动(pull)两种方式。对零件的进入采用拉进规则。

送出规则(Output Rule):也有主动和被动两种方式。当零件加工完后,采取主动送出规则。

各机器的加工周期、进入规则、送出规则如表 3.2 所示。

表 3.2 各机器的加工周期、进入规则、送出规则表

| 机器名 项目 | W | Veigh | | | Wash | | Prod | uce | Insp | ect |
|-----------|------------|--------|-----|----|---------|----|------|-------|------|------|
| 加工周期 min | | 4 | | 3 | | 3 | | | | |
| 进入规则 | PULL from | Widget | out | of | PULL | | PULL | from | PULL | from |
| | WORLD | | | | from C1 | | C2 | | C3 | |
| 送出规则 | PUSH to C1 | | | | PUSH | to | PUSH | to C3 | PUSH | to |
| | | | | | C2 | | | | SHIP | |

(4) 仿真运行:运行一周(5天*8小时*60分钟=2400分钟),得到统计结果如表3.3和表 3.4 所示:

表 3.3 机器统计信息报表

机器名 空闲 率忙 率已完成零件数



| | % | % | |
|---------|-------|-------|-----|
| Weigh | 0 | 100 | 480 |
| Wash | 20.33 | 79.67 | 478 |
| Produce | 40.46 | 59.54 | 476 |
| Inspect | 40.62 | 59.38 | 475 |

表 3.4 零部件统计信息报表

| | | | 平 均 | |
|------|------|-------|------|--------|
| 进入数量 | 送出数量 | 在制品库存 | WIP | 平均通过时间 |
| 481 | 475 | 6 | 5.92 | 29.54 |

(5)仿真分析:通过仿真运行的结果可以看出,称重机器满负荷运转,是本系统的瓶颈。小部件的产出量为475个。每个部件通过时间为29.54分钟,由于实际被加工时间是:5+4+3+3=15分钟,所以它的输送与排队时间占了较大的比率(49%)。

第四章 元素——模型的组成部分

现实的商务或事物系统总是有一系列相互关联的部分组成的,比如制造系统中 的原材料、机器设备、仓库、运输工具、人员、加工路线或运输路线等,服务系统 中的顾客、服务台、服务路线等。Witness 软件使用与现实系统相同的事物组成相应 的模型,通过运行一定的时间来模拟系统的绩效。模型中的每个部件被称之为"元 素(Element)"。该仿真软件主要通过如下五类元素来构建现实系统的仿真模型:离 散型元素、连续型元素、运输逻辑型元素、逻辑型元素、图形元素。

4.1 离散型元素

离散型元素是为了表示所要研究的现实系统中可以看得见的、可以计量个数的物体,一般用来构建制造系统和服务系统等。主要包括:零部件(Part)、机器(Machine)、输送链(Conveyor)、缓冲区(Buffer)、车辆(Vehicle)、轨道(Track)、劳动者(Labor)、路径(Path)、模块(Module)。

4.1.1 零部件 (Part)

零部件是一种最基本的离散性元素,它可以代表在其他离散性元素间移动的任 何事物。如产品、大公司全程处理的项目、电话交流中一个的请求、微型电子元件、 超市中川流不息的人、医院中的病人、机场上的行李等等。

在模型中,零部件的使用方法有很多种。我们可以单独使用零部件,可以将多 个零部件组装成一个零部件,也可以将一个零部件分成许多零部件。零部件可以被 同批处理,可在同一时间被批量或单个创建,在模型的处理过程中还可以转变为另 一些零部件。

零部件进入模型主要有两种方式。第一种方式是被动式的,只要有需要,零部 件可以无限量进入模型。如在生产性企业中,一些零部件堆放在仓库中,当生产需 要时,可以随时把它取出来供应生产。第二种方式是主动式的,零部件可以间隔固 定的一段时间(例如,每隔10分钟)进入模型;可以按照一定的随机分布进入模型, 如顾客到达商店的时间间隔服从均匀分布;也可以以不规则的独特的时间间隔(例 如,10分钟,20分钟,30分钟)到达模型中;还可以以重复的不规则的方式进入模 型,例如,一个餐馆,有50位"顾客"(零部件)在上午8点到达那里,10位顾客 在上午8:01至11:59到达那里,50位在中午12点到达,50位在12点半到达等等, 每星期都如此。在该方式中,我们可以对零部件达到模型的时间、时间间隔、到达 最大数量等选项进行设置。

4.1.2 机器 (Machine)

机器是获取、处理零部件并将其送往目的地的离散元素。不同的机器代表不同



类型的处理过程。一台机器可建立不同的模型,它可以代表有装载、旋转、卸载、 空闲和保养这五个状态的一台车床,也可以代表有空闲、工作、关闭三个状态的一 个机场登记服务台 (将旅客与他们的行李分开 , 并发放登机卡) , 还可以代表有焊接 , 空闲和保养三个状态的一个机器人焊接工等等。

Witness 提供了七类机器来建立不同类型处理过程的模型:(1)单处理机 (single)。单处理机只能一次处理一个部件,单输入单输出。(2)批处理机(batch)。 批处理机一次能处理多个部件,n个部件输入n个部件输出。(3)装配机(assembly)。 装配机可将输入的多个零部件组装成一个组件输出,n个部件输入1个部件输出。 (4)生产机(production)。一个原部件输入到生产机中能输出许多部件。1个部件输出 入n个部件输出。如单片钢板的切割,会得到一些成品和边角料。我们要注意的是, 生产机不仅输出原部件,而且输出带有规定生产数目的部件,就如相片的加洗,在 复制的最后,我们得到了需要数目的复制品再加上原件。(5)通用机(general)。在 通用机器中输入一批部件,输出的是相同数目或不同数目的一批部件,这个处理过 程可能存在单个循环或多重循环。(6)多周期处理机(multiple cycle)。多周期处理 机是一台特殊的通用机器,它模拟机器实行的是经过许多独立的处理周期来完成一 次操作。可以为每个周期指定不同的输入、加工时间、输出数量。(7)多工作站机 (multiple station)。一台多工作站机工作起来就像许多台联结在一起的机器。它有 多个不同的部件加工位置,每个部件将依次通过每一个工作站,完成一系列的工序。

4.1.3 输送链 (Conveyor)

输送链是一种可以实现带传送和滚轴传送的离散性元素。如机场里运送行李的 传送带,将卡车车体沿生产线移动的传送装置,将空纸盒送往包装操作的滚轴传送 装置等等都可以称为输送链。

Witness 提供了两种输送链:(1)固定式(Fixed)。这是一种保持部件间距不变的输送链。假如该输送链停了,它上面的部件间的距离仍保持不变。(2)队列式 (Queuing)。这种输送链允许部件的累积。假如该输送链上的部件被阻塞,部件仍 不断的滑向一起,直到这个输送链被塞满。

输送链通常把部件从一个固定点移到另一个固定点。部件从输送链后段进入, 并向前移动。我们能确定部件在输送链上的特定位置,并可以将部件装载或卸载到 特定的位置。不管是固定式输送链还是队列式输送链都可能发生故障,需要工人来 修理。在设计输送链时,我们可以对它的长度、最大容量、部件移动每单位长度所 需的时间等项进行设定。

4.1.4 缓冲区 (Buffer)

缓冲区是存放部件的离散元素。例如存放即将循环焊接的电路板,即将用于旅 行的真空包装食品的储藏区,位于加工区的盛放产品部件的漏斗形容器等等物体都 称为缓冲区。

缓冲区是一种被动型元素,既不能像机器元素一样主动获取部件,也不能主动



将自身存放的部件运送给其他元素;它的部件存取依靠系统中其他元素主动的推或 拉。我们可利用缓冲区规则,使用另一个元素把部件送进缓冲区或者从缓冲区中取 出来。部件在缓冲区内还按一定的顺序整齐排列(例如,先进先出,后进先出)。

我们可以将缓冲直接与机器相结合,在一台机器中,设置一个输入缓冲和一个 输出缓冲,这种缓冲区我们把它称之为专用缓冲区。专用缓冲区不是一种独立的元 素,我们可以在设置机器元素的输入和输出规则时,设置它的输入缓冲和输出缓冲。

4.1.5 车辆 (Vehicle)

车辆是一种离散元素,我们用它来建立的装置模型可以将一个或多个部件从一 个地点运载到另一个地点。(例如,卡车,起重机,铲车)。

车辆沿着轨道(track)运动。虽然车辆实际上自身在移动,但却是轨道定义了 物理布置图,并包含了使系统运行所需的逻辑理论。

在 Witness 里建立运输系统应按照如下两个步骤:

(1) 设计轨道布置图和运载路线。这需要创建所需的轨道和车辆,并且详细说明车辆在轨道之间移动所需的细节。在这一步不要考虑部件怎样装上车辆或怎样从上面卸载下来。只有将第一步设计好了,我们才可以进入下一步的工作。

(2) 详细说明我们所定义的车辆怎样来满足运输的需要。它有两种方式,可能是被动式的,也可能是主动式的。

4.1.6 轨道 (Track)

轨道是一种代表车辆运输部件时所遵循的路径的离散元素。它们也定义了车辆 装载,卸载或停靠的地点。

车辆所走的路径是由一系列轨道组成的。每条轨道都是单向的;假如你需要一 条双向的轨道,只需定义两条沿相同线路但方向相反的轨道就可以了。车辆在"尾 部"(rear)进入轨道并向"前部"(front)运动。一旦到达前部,该车辆可以进行装 载,卸载或其它的操作。然后它将移动到下一条路线的尾部并开始向那条路线的前 面运动。

根据一个车辆的类型来规定它的路径是可行的,所用方法与根据部件类型来规 定它们在元素间的路径一样。在仿真运行的开始,所有的车辆沿一条特定轨道进入; 这与部件的初次进入类似。

4.1.7 劳动者 (Labor)

劳动者是代表资源(例如,工具或操作工人)的离散元素,它一般负责对其它 元素进行处理、装配、修理或清洁。如从事精密工作的机器人、一台选矿机或一个 固定装置都是劳动者。

我们通过对特定的劳动者的属性设置来建立同一劳动单位在不同技术水平下的 模型。我们可以对各种类型的劳动者设置不同的班次,可以在模型中加入班次的构 成,并且用不同的工作方式,休息和加班时期来进行试验。假如另外一个元素要完



成更重要的任务,我们可以从元素中撤离劳动者到该元素中去。这就是劳动者使用 的优先权。

4.1.8 路径 (Path)

路径是设定部件和劳动者(或者其它资源)从一个元素到达另一个元素的移动 路程的离散元素。我们在模型中可以用它来代表现实系统中行程的长度和实际路线。

不论何时,当运动时间对于两项操作非常重要时,路径对于提高模型的精确性 是特别有用的。在一个制造单元的模型里,一个操作者要控制数台机器的操作,在 各台机器之间的走动时间是完成整个任务总时间的重要组成部分。此时,路径就起 到了作用。路径还有许多其它的用途,例如,选定仓库贮存的应用模型,详细的零 售规划的模型,机场或医院的规划等等。

只有在必要时我们才使用路径。假如模型中的元素有很长的周期时间而且它们 间的行程很短,那就不必去增加模型的复杂性了。路径的使用应基于建模对象(例 如,在某些情况下用轨道或车辆代替可能会更合适)。

4.1.9 模块 (Module)

模块是表示其他一些元素集合的离散元素。有了模块,在模型内部,我们就可 以建立具有自处理功能的模型。例如,一家工厂的油漆店可能由许多 Witness 的元素 构成。我们可以定义一个包括所有这些元素的"PSHOP"模块。然后对这家油漆店 以外的其它元素定义一些规则,比如说推入"PSHOP"。

使用模块有几种方式:(1)详述模块里的一个处理过程并且重点检查在这个特 殊处理过程中的各个元素;(2)详述模块里的一个处理过程但随后将这个模块拆毁 以便于能着重于顶层模型的设计。元素从模型到达模块的一个输入点(或元素),经 过模块内适当的元素,然后经由这个模块的一个输出点(或元素)返回模型。(3) 为模块输入一个近似的周期时间以便能运行整个模型,并在随后填充模块里的元素。 假如模块使用一个周期时间,我们也能生成这个模块的报告(4)详述模块中的一个 处理过程,将它保存到一个模块文件中(*.mdl),这个文件包含所有关于这个模块 的信息,元素和图标等等。然后把这个文件载入另一个 Witness 模型中。(5)在一个 模块中可以创建另一个模块。这就是阶层模型。(6)可以利用一个特殊的模块结构 存储对话框所需元素的数据来建立自己的对话框。(7)我们还可以用密码来保护模 块。

4.2 连续型元素

同离散型元素相对应,连续型元素用来表示加工或服务对象是流体的系统,比如化工、饮料等。主要包括:流体(Fluid);管道(Pipe);处理器(Processor);容器(Tank)。



4.3 运输逻辑型元素

运输逻辑型元素用于建立物料运输系统。主要包括:运输网络(Network);单件运输小车(Carriers);路线集(Section);车辆站点(Station)

4.3.1 运输网络 (Network)

运输网络把一系列的路线集,工作站和单件运输小车组合在一起。我们必须把 每一个提供能量的单体元素分配到网络中去。网络的建立方式影响着其内部提供能 量的单体元素的行为。

运输网络可以分为自动提供能量和路线集提供能量两种类型。如果该网络是自动提供能量型的,则单件运输小车是主动的并推动自身向被动的路线集运动。例如 一个"ROBOT"单件运输小车在"LOAD_TUBE"工作站装载了一个"TESTTUBE" 部件,沿着一条叫做"SECTION1"的固定路径移动,并且在"UNLOAD_TUBE" 工作站把该部件卸下。如果该网络是路线集提供能量型的,路线集的行动类似于附 带有铁钩的带传送装置。路线集上的铁钩钩起非活动性的单件运输小车并且把它们 带往下一个元素,然后放下这些单件运输小车。最后空钩子绕回路线集的起始处, 准备钩起另一个单件运输小车。例如一个"SCOOP"单件运输小车装载了一个 "APPLE"部件,在一个叫做"BELT1"的路线集上把"SCOOP"单件运输小车用 铁钩钩起,将它们移动到"BELT1"路线集的尾部,然后把"SCOOP"单件运输小 车从铁钩上放下,空铁钩则沿着路线集返回起点。

使用运输网络应注意以下两点:

- 在同一个网络中,只能使用路线集,工作站和单件运输小车;
- 网络所应用的类型和班次也被应用于所有配置在该网络中的路线集,单件运输 小车和工作站。

4.3.2 单件运输小车 (Carriers)

单件运输小车沿着路线集或工作站来运输部件。它的运输方式取决于网络的类型。它可以在两个网络之间移动。

使用单件运输小车应注意以下七点:

- 每个单件运输小车的最大搬运量是一个部件;
- 单件运输小车可以从一个网络移动到另一个网络;
- 可以在每个网络中使用多个类型的单件运输小车;
- 单件运输小车只有在路线集体提供能量的网络中才能跨越式运动;
- 一个单件运输小车的入口规则支持 "PUSH, PERCENT 和 SEQUENCE"输出 规则;
- 可以把单件运输小车从一个模块推到另一个模块;
- 当定义一个单件运输小车的时候,必须把它配置到网络中去。然而,Witness 只



有在运行模型的时候才会去检查该搬运工具是否配置到有效的网络中去了。

4.3.3 路线集 (Section)

路线集是一种代表单件运输小车所走路径的提供动力的单体要素。在模型中, 路线集是网络的组成部分。

使用路线集应注意以下三点:

- 只有在运行模型时,Witness 才会去检查这个路线集是否配置到有效的网络中去 了;
- 可视规则编辑器不支持路线集连接规则;
- 路线集连接规则支持"PUSH, PERCENT, SEQUENCE"输出规则。

4.3.4 工作站 (Station)

工作站是代表一个点的提供动力的单体元素,该点在路线集的起始或末尾,在 这个点上,我们能对单件运输小车或者其里面的部件实施操作。共有四种类型的工 作站:(1)基站(Basic)。当单件运输小车(或单件运输小车上面的部件)进入、 离开或在工作站内时,可以对它们进行操作。(2)装载站(Loading)。可将部件装 入单件运输小车,指派劳动者去协助装载作业,并可以在单件运输小车装载部件的 时候实施操作。(3)卸载站(Unloading)。可以从一个单件运输小车里卸载部件, 指派劳动者去协助卸载作业,并可以在单件运输小车卸载部件的时候实施操作。(4) 停泊站(Parking)。工作与缓冲十分相象,它是一个不引起路线集堵塞的可供单件运 输小车等待的空间。

使用工作站应注意以下五点:

- 只有运行模型的时候,Witness 才会去检查工作站是否已配置在有效的网络中。
- 可视规则编辑器不支持工作站连接规则。但我们能利用可视的推、拉规则(比如"SEQUENCE"和"PERCENT")去将部件推进或拉出合适的工作站。
- 所有工作站类型都支持自由处理法(在进行处理时,单件运输小车与传送装置 分离),装载站和卸载站也支持由动力推动的处理方法(单件运输小车在处理的 操作中始终与传送装置机构相连)。
- 不建议使用"系列"动力工作站,因为装载/卸载操作可能在进入后一个工作站 之前没有完成,而且还可能因此产生意想不到的后果。
- 工作站连接规则支持"PUSH, PERCENT和 SEQUENCE"输出规则。

4.4 逻辑元素

逻辑元素是用来处理数据、定制报表、建立复杂逻辑结构的元素,通过这些元 素可以提高模型的质量和实现对具有复杂结构的系统的建模。主要包括:属性 (Attribute);变量(Variable);分布(Distribution);函数(Function);文件(File); 零部件文件(Part file);班次(Shift);



4.4.1 属性 (Attribute)

属性是反映单个部件,劳动者,机器或单件运输小车特性的元素。例如,我们 可以用属性来形容颜色,大小,技能,成本,密度,电压或数列等。

我们可以在仿真的过程中改变属性的值。例如,一个部件的"颜色"属性的值 开始是"灰",在部件通过了一台"着色"机器之后变成了红色。可以用活动"action" 来设置、检查或改变任何属性的值。Witness提供了许多能用于部件,劳动者,车辆, 机器或者单件运输小车的系统属性,另外我们也可以自己定义用于部件,劳动者, 车辆,机器或单件运输小车的属性。

Witness 本身已经包含了一些能使用的属性,这些属性就叫做系统属性。每个部件,单件运输小车,车辆,机器和劳动者都带有"PEN,ICON,DESC and TYPE" 属性;"CONTENTS and FLUID"属性用于盛放液体的部件;"STAGE,NSTAGE, R SETUP and R CYCLE"属性则是用于部件走的路线。

当创建部件属性时,可以将它分配给十一个组(0组--10组)中的任何一组,然 后在部件详细的属性设置页上,将该组分配给该部件。部件的属性值可以是变量或 者是常数。

当创建劳动者,单件运输小车,机器或车辆的属性时,必须把这些属性分配给 0组。劳动者,单件运输小车,机器和车辆的属性永远是变量。

4.4.2 变量 (Variable)

变量包含了一个值(或一系列的值,假如这个变量的数量大于1)。当定义一个 变量时,我们必须还要选定它的数据类型,这个数据类型说明了变量所含有的数据 类型(整型,实数型,名型和字符型)。

Witness 共有三种类型的变量:

(1)系统变量。这些变量是系统已经创建好了(I,M,N,TIME,VTYPE和ELEMENT) 的,并且具有特殊意义的变量,它们存储仿真中常用的数据,例如,TIME表示现在 的仿真时钟。

(2)全局变量。全局变量是我们自己利用"Define, Display 和 Detail"过程创建的 作为 Witness 元素的变量。

与局部变量比较起来,用全局变量的好处在于:

- 柔性。我们可以从模型的任何地方检查或更新一个全局变量的值。例如,变量 "TOTAL_SHIPPED"能被模型中用于将部件送出模型的所有的元素更新。同 样,任何函数,行为规则等等都可以读取"TOTAL_SHIPPED"变量中包含的 值。
- 我们能生成全局变量的报告,但不能生成局部变量的。
- 我们能在模型中显示全局变量和它们的值。
- 全局变量可以被设定为数组。我们能通过给一个全局变量 1 以上的下标来创建 数组(行,列和数据表格),我们最多能创建 15 维的数组。



 我们能创建一个整型或实数型的变量作为动态变量,这意味着它能容纳多个值。
 例如,一个动态变量能包含每个部件离开模型的仿真时间。在仿真的开始这个 变量里没有值,然后当第一个部件离开时有了1个值,当第二个部件离开时有 2个值,以此类推。

(3)局部变量。局部变量是一个我们能自己在使用它的活动或函数中创建的变量。 局部变量只能是一个数 , 而不能是带有下标的数组。

局部变量的定义方式如下:

DIM 变量名 {AS 数据类型} {!注释}

如果省略了数据类型的定义,系统赋予变量默认的数据类型为整型 integer。

用局部变量而不用全局变量的好处在于:a. 安全。局部变量只有在一个行为 (action)或函数执行的时候才存在,所以不可能在另一个行为(action)或函数中使用 或修改它。例如,变量"TOTAL_SHIPPED"已在一个机器的"action"中被定义了, 直到结束它都只能被那一系列行为更新或读取,而不能被这台机器的其它行为或模 型中的其它元素更新或读取。b. 快速。当行为和函数使用局部变量而不是全局变量 时,它们能被更快地执行。c. 方便。局部变量在使用它们的行为中被定义,不必像 全局变量那样先定义它们。

4.4.3 分布 (Distribution)

分布是一个逻辑性元素,我们从"现实世界"搜集数据,并用分布代表模型中 具有规律性的变化。例如,假设观察证明某一种特定部件的打磨操作需要 5 至 10 分 钟,但大部分部件通常是 8.2 分钟完成,我们就可以用分布把这些信息引入模型中。

Witness 提供了一些标准分布。其中有一些是将一系列理论分布返回到随机样本的分布。Witness 包含的理论分布曾在很长一段时间内被广泛研究并且被认为在仿真 中是最有用的。还有一些是一系列整数和实数的分布。当使用一个标准分布时,必 须为其输入一个伪随机数流和参数。

假如没有标准分布适用的情况,或者我们收集的现实生活中的数据是在未研究 领域中的,我们可能需要在 Witness 中建立自己的分布并从中采样。我们能创建整型, 实数型和名称型的分布,并且它们可以是离散(从分布中选择实际值)的或是连续 的(从一串连续值中选择一个值)。

总的来说,假如我们有详尽的现实生活的数据,那就创建自己的分布。如果没有,那么就选择 Witness 提供的最适当的标准分布。

4.4.4 函数 (Function)

函数元素是能返回有关模型状态的信息或者使得模型显得更具有真实性的一组 命令集合。Witness 提供了大量能直接使用的函数,同时我们也能创建自己的函数。

如下两种情况创建自己的函数是特别有用的:

- 对许多元素使用相同的操作。
- 这些行为模块包含了很多说明。



例如,假设在计算一台机器的周期时间时要考虑多种因素,而我们在周期时间 表达式中的输入又不能超过一行,在这种情况下,我们就可以自己创建一个函数, 想写多少行就写多少行,然后只要把这个函数的名称输入这台机器的周期时间表达 式区域就行了。

4.4.5 文件 (File)

文件是可以使我们从仿真模型外部将数值输入模型(从一个"READ"型文件) 或从模型中输出值(到一个"WRITE"型文件)的一个元素。例如,我们能从其他 软件生成的文件读入如周期时间这样的值,或者生成适当的报告。

使用文件时我们应注意以下几点:

- 可以用文字处理工具或文本编辑工具(或其它能生成简单 ASCII 文本文件的程序)来创建"READ"文件。在这样的文件中以"!"符号开头的行被略去不读。
- 不要在仿真运行时对同一个文件进行读和写的操作。
- 假如有两个模型在仿真运行,应该保证它们不对同一个文件进行写入操作,但 从同一个文件中读出是可行的。
- 假如要在运行中检查"WRITE"文件,应该在检查前先把它关掉,这样才能检 查到一个完全更新了的文件。

4.4.6 零部件文件 (Part file)

"READ"型零部件文件是从外部数据文件读入零部件清单到模型中去的一个逻辑元素。"WRITE"型零部件文件是将零部件清单写入外部文件的逻辑元素。

零部件文件可用于从一个模型中生成输出,然后将其用于另一个模型中。零部 件文件对于追溯零部件离开仿真的确切时间和零部件在那时的属性值也是很有用 的。使用零部件文件应注意以下两点:

- 不要在一个仿真运行时对同一个文件进行读和写的操作。
- 假如有两个模型在仿真运行,应该保证它们不对同一个文件进行写入操作,但
 是从同一个文件中读出是可行的。

4.4.7 班次 (Shift)

班次是一个能用来创建一个班次模式或一系列班次模式的逻辑元素,它作用于 一连串的工作和非工作时期。其它元素仿真班次工作时可以引用班次模式。我们可 以将班次应用于下列元素:

| 缓冲 | 运输网络 |
|------|------|
| 传送装置 | 饼状图 |
| 流体 | 管道 |
| 劳动者 | 槽 |
| 机器 | 时间序列 |
| 零部件 | 车辆 |



零部件文件

我们可以以详细的方式输入包含有班次数据的 ".sft"文本文件。注意,班次数据不能涉及到不存在的次级班次(sub shift)。

4.5 图形元素

图形元素可以将模型的运行绩效指标在仿真窗口动态的表现出来。主要包括: 时间序列图 (Timeseries); 饼状图 (Pie chart); 直方图 (Histogram)。

4.5.1 时间序列图 (Timeseries)

时间序列图是以图形方式来画出仿真随时间变化的值,从而表现仿真结果的图 形元素。垂直的 Y 轴代表值,水平的 X 轴代表时间。可以选择以下的一种方式来表 示 X 轴:

- 仿真时间。当一个点在 X 轴上被标注时一个仿真的时间就被记录下来了。
- 表达式。不论何时,只要表达式被求值,一个标注点就被确定下来了,而且标 注该点时的仿真时间被记录在 X 轴上。
- 24小时制。X 轴以 24小时制列出小时数。
- 12小时制。X 轴根据 12小时制列出小时数。
- 8小时制。X 轴根据 8 小时制列出小时数。
- 小时制。X 轴以 1, 2, 3 等等列出小时数。

时间序列在预测模型的趋势和周期方面是非常有用的,因为它们提供了给定值 的历史数据以及静态的平均值和标准差。

时间序列图类似于一个 "pen plotter ": 它在仿真时标注点。Witness 在给定的时间间隔从模型中 "读取",并且在一个图上 "标注点",在一段时间内建立一系列的值。一旦屏幕上分配给这个时间序列图的空间用完了,这个图形会"翻页"以使新的点可以被标注。虽然 Witness 时间序列的标注点以一条连续的线条显示,但这条线条是将各个在仿真时间点收集的值点连接起来的标注点连线。这条连接标注点的线条仅仅说明了值的变化方向。我们可用 7 种不同的颜色来标注 7 个值。

4.5.2 饼状图 (Pie chart)

饼状图用来在仿真窗口表示仿真结果,显示如何使用一个或一组元素的图形元 素。例如,我们可以用一个饼状图来分块表示一个给定时段的空闲时间,装配时间 和工作时间。

4.5.3 直方图 (Histogram)

直方图是一种在仿真窗口用竖条式的图形来表示仿真结果的图形元素。在模型 中适当的地方我们可以用"record"、"drawbar"、"addbar"行为在直方图中记下值。





第五章 规 则

我们一旦在模型中创建了元素,就必须说明零部件,流体,车辆和单件运输小 车在它们之间是怎样流动以及劳动者是怎样分配的,这就要用到规则。

Witness 有几类不同的规则:

- 输入规则(这类规则包括装载和填入规则)。 输入规则控制输入元素的零部件或 者流体的流量。
- 输出规则(这类规则包括连接,卸载,空闲,单件运输小车进入,车辆进入和 缓冲区退场管理)。输出规则控制从元素中输出的零部件,流体,车辆或者单 件运输小车的流量。
- 劳动者规则。劳动者规则可用来详细说明劳动者的类型和机器,输送链,管道, 处理器,容器,路线集或者工作站为了完成一项任务而需要的劳动者的数量。
 我们可以利用可视化规则对话框输入简单的规则,并且在模型窗口中显示流动

方向;或者可以通过使用规则编辑器输入更复杂的规则。

5.1 输入规则 (Input Rule)

输入规则控制零部件或者流体进入在系统中的流动过程。例如: 一台空闲机器 要启动的话,会按照输入规则输入零部件直到有足够的零部件启动它;一台尾部有 空间的输送链在每向前移动一个位置时,按照输入规则输入零部件。

可以通过以下几种方法输入零部件或者流体:

- 具有相同名称的一组元素。
- 一组元素中的一种特殊的元素(需要指定那种元素的下标)。
- 在模型外的一个特定的位置(WORLD)中得到零部件或流体元素。
 设定输入规则的方法主要有两种:
- 通过元素细节(detail)对话框中的"FROM"按钮

首先选中对象,然后双击鼠标左键,在弹出式 detail 对话框中的 general 页中, 点击该按钮就显示出输入规则编辑器。

● 使用可视化输入规则按钮

| 🥪 Input Rule for Machine001 | | × |
|--|-----------------------------|-------|
| Rule: Operation PULL Oycle | World Ship Sgrap Route Asmb | OK |
| n a pull | <u>E</u> dit | Close |

图 5.1 输入规则对话框

首先选中对象,然后点击 Element 工具栏上的"Visual Input Rules" 🛃 图标,

W

北京威特尼斯科技中心

将会显示如图 5.1 的输入规则对话框,然后进行输入设定。

通过这两种方法设定了元素的输入规则后,都会使得元素 detail 对话框 general 页框中的"FROM"按钮下方,显示出元素当前的输入规则的名称。当创建一个元素的时候,Witness 会自动的给它一个默认的规则——WAIT,表示它不能接收零部件或者流体。为了规定零部件和流体通过模型时的路线,我们必须用一个其他的规则来代替 WAIT 规则。

要注意的是,在设定输入规则之前要先弄清楚零部件,流体,车辆和单件运输 小车的在模型中的路径。我们不能把它们弄反了,举例来说,我们不能在元素 A 向 元素 B 输入零部件的同时,元素 B 向元素 A 也输入零部件。另外,还可以考虑使用 零部件路线(ROUTE)来控制它们通过模型的路线。

Witness 提供的可以在输入规则使用的命令有:

| BUFFER | FLOW | LEAST | MATCH |
|--------|----------|-------|--------|
| MOST | PERCENT | PULL | RECIPE |
| SELECT | SEQUENCE | WAIT | |

5.2 输出规则 (Output Rules)

输出规则控制着当前元素中的零部件,流体,车辆和单件运输小车输出的目的 地和数量等。例如:一台机器在完成对零部件的加工后按照一个输出规则将零部件 输出到另一台机器上。要是它出了什么故障不能这样做,那将会出现堵塞向现象; 当一个零部件到达一个有输出规则的输送链前方时,输送链将把零部件输出,如果 输送装置由于故障不能将零部件输出的话,这里将会出现堵塞(固定输送链)或者 排长队(队列式输送链);车辆到达有输出规则的轨道前方的时候,轨道把车辆输送 到另外一个轨道上面,要是轨道输送失败,路线将会变的堵塞;一台有输出规则的 处理器完成对流体的处理后,把流体输出;一单件运输小车到达一个有输出规则的 路线集的时候,路线集输出它到下一路线集。

可以输出零部件或者流体到:(1)具有相同名称的一组元素;(2)一组元素中的一种特殊的元素(指定该元素的下标);(3)模型外的一个特定的位置(SHIP, SCRAP, ASSEMBLE, WASTE, CHANGED, ROUTE 或者 NONE)。

设定输出规则的方法主要有两种:

● 通过元素细节(detail)对话框中的"TO"按钮

首先选中对象,然后双击鼠标左键,在弹出式 detail 对话框中的 general 页中, 点击该按钮就显示出输出规则编辑器。

使用可视化输出规则按钮

首先选中对象,然后点击 Element 工具栏上的"Visual Output Rules" 🗞 图标, 将会显示如图 5.2 的输出规则对话框,然后进行输入设定。



| 🥪 Output Rule for Machine001 | | × |
|------------------------------|------------------------------------|------|
| Rule: Operation PUSH VCle | <u>World</u> Ship Scrap Route Asmb | OK |
| ro Carlense | <u>E</u> dit C | lose |

图 5.2 输出规则对话框

元素 detail 对话框 general 页框中的"TO"按钮的下方,Witness 会显示元素当前 的输入规则的名称。当我们创建一个元素的时候,Witness 会自动的给它一个默认的 规则——WAIT,表示它没有传送零部件或者流体到其他元素的规则。为了规定零部 件和流体通过模型时的路线,我们必须用一个其他的规则来代替 WAIT 规则。

输出规则的注意点跟输入规则相类似,设置输出规则时,同样应先搞清楚零部件, 流体,车辆和单件运输小车的流动路线,也可考虑使用零部件路线记录来控制它们 通过模型的路线。

Witness 提供的可以在输出规则中使用的命令有:

| BUFFER | CONNECT | DESTINATION | FLOW |
|--------|---------|-------------|------|
| LEAST | MOST | PERCENT | PUSH |
| RECIPE | SELECT | SEQUENCE | WAIT |

5.3 劳动者规则 (Labor Rules)

5.3.1 劳动者规则概述

机器,输送链,管道,处理器,容器,路线集和工作台都需要劳动者才能完成 任务。劳动者规则可以让我们详细说明实体元素为完成任务所需要的劳动者类型和 数量。我们可以通过创建劳动者规则来完成的任务有:调整机器,并为它设定时间 周期或修理它;修理输送链;帮助流体通过管道,并且做好清洁、清洗和修理的工 作;帮助处理器处理流体,并且做好填入、清空、清洁和修理工作;帮助流体通过 管道,或做修理工作;修理各种类型的工作站,在行为站做好进入、处理、退出动 作,在装载(卸载)站做好装载(卸载)工作,在停靠站做好停靠工作;修理路线 集。

可以使用元素细节对话框进入劳动者规则。如果一个元素需要劳动者,点击元 素细节对话中的对应按钮。一个细节对话框可能包含几种劳动者规则按钮,例如, 一台机器就有装配,循环,修理几种劳动者规则。劳动者规则按钮旁边如果有打钩 的标记,则表明我们已经为这项工作建立了劳动者规则;如果有打叉的标记,则表 示我们没有为这个工作建立劳动者规则。点击劳动者规则按钮后,弹出规则编辑器, 在这里我们可以输入劳动者规则。

可以使用"Visual Labor Rules" 👗 按钮,来输入劳动者规则,但要注意在我们



使用元素的劳动者规则之前,我们必须建立 Labor 元素。输入劳动者规则最简单的 方法是对元素的 Labor 规则编辑框中输入需要的劳动者元素的名称,例如,如果一 台机器需要一个操作者处理零部件,只需要输入 OPERATOR 作为劳动者规则就可以 了,当然先要定义一个 Labor 元素,其名称叫 OPERATOR。

5.3.2 三种劳动者规则

我们也可以使用如下 3 种劳动者规则,当然有时也可将这三种劳动者规则结合 起来使用。

5.3.2.1 NONE 规则

在某种情况下,当元素不需要劳动者时,我们可以在劳动者规则中使用 NONE 规则。例如,当一个元素完成某项任务不需要劳动者时,我们不需要输入任何规则。 该规则经常用在 IF 条件语句中,如

IF JOB_TYPE = RIBBON

KNOTTER

ELSE

NONE

ENDIF

在这个例子中,包装机包装两种不同类型的巧克力箱:一种是有带子的,一种 是没有带子的。机器包装没有带子的巧克力箱时,不需要劳动者;包装有带子的时候(JOB_TYPE= RIBBON),需要进行打结的劳动者 KNOTTER。劳动者规则指示机 器当遇到有带子的巧克力箱时,获取劳动者;当遇到无带子的巧克力箱时,不获取 劳动者。在这里如果我们使用"WAIT"规则而不是"NONE"规则时,只要无带巧 克力箱一进入,机器就会阻塞。

5.3.2.2 MATCH 规则

在 WITNESS 中,既可以用 MATCH 规则作为输入规则来输入一系列相匹配的 零部件或者劳动力单位到机器,也可以用它来作为劳动者规则,设定某一元素,匹 配完成某项工作所需的劳动力单位数。

(1) MATCH 规则的语法如下:

MATCH / qualifier location1 $\{\#(qty)\}$ {[AND | OR] location2 $\{\#(qty)\}\}$ {...}

关于 qualifier 的设置在下面我们将会详述, location 取决于我们是将 MATCH 规则作为机器的输入规则还是作为劳动者规则的, #(qty)是指 MATCH 规则从 location 中所选的元素的数量。它是一个整型表达式。如果不具体说明匹配的数量,则 Witness 默认数量为 1。

作为机器的输入规则的语法如下:

{part_ name {from}} location _name {(index _exp)} {at position_ exp}
{with labor_ name {#labor_ qty}} {using PATH}

Part _name 是指模型中零部件的名称; location _name 是指元素输入或输出零部



件的地方,它可能是个名字表达式或是个模块的名称;如果发送零件的元素数量大于1,我们可以详细说明一组元素中的哪一个用来发送该零件,Index_exp是一个含有括号的整数表达式,例如,如果 location_name 是 PACKING,则 PACKING(3)表示第三台包装机,如果我们不使用 Index_exp,规则从一组元素中的任何一个元素输入零部件;如果发送或者接收的零部件是输送链,我们可以从特殊的位置输入或者输出零部件,Position_exp是一个包含有输送链位置号的整型表达式。如果我们不使用该表达式,Witness 从输送链的后方输入零部件(或输出零部件到后方);With labor_name 是模型中要求与机器项目匹配的劳动单位的名称;#labor_qty 是指被详细说明的与机器项目相匹配的劳动者数量。using PATH 指定相匹配的项目从前一地 点输入机器时使用一条路径。

作为劳动者规则的语法为: {labor_name} {(index_exp)} {using PATH}

其中 Labor _ name 是指模型中劳动者的名称。

限定词是如下几种之一

·ANY 元素选择列出地点的任何一个。

·ATTRIBUTE attribute_ name {(attribute _index _exp)}

ATTRIBUTE 指元素具有特殊属性值的地方(例如,SIZE = 10); attribute_name 是指系统属性的名字或是自己创建的属性的名字; attribute_index_exp 是一个供选择 的属性目录,它由一个整数表达式组成。如果使用目录,规则使用一组同类属性中 的一种特殊的属性;如果不使用目录,规则使用一组同类属性中的任何一种属性。

·CONDITION condition CONDITION 是指元素的一般属性符合某种条件的地方; condition 是一个条件, 如 SIZE>10。

(2) 下面是有关 MATCH 作为机器的输入规则的例子:

例 1. MATCH/ANY CLOCK STORE(1) #(2) AND BOX STORE(2) #(1)

在这个例子中, MATCH 规则从第一个 STORE 元素中取出任意两个钟表零部件, 再从第二个 STORE 元素中取出一个盒子零部件,并把它们一起输入到机器中去。 例 2.MATCH/ATTRIBUTE COLOR CHASSIS BUFF # (1) AND DOOR BUFF # (4)

在这个例子中,MATCH 规则将从 CHASSIS_BUFF 缓冲区中取出 1 个零部件, 再从 DOOR_BUFF 缓冲区中取出 4 个与之颜色属性相同零部件,并把它们输入到机 器中去。

例 3. MATCH/CONDITION SIZE>10 STOREA #(7) OR STOREB #(7)

在这个例子中, MATCH 规则从 STORE_A 缓冲区中取出 7 个 SIZE 属性值大于 10 的零部件或者从 STORE_B 中取出 7 个 SIZE 属性值大于 10 的零部件,并将它们 输入机器。

例 4. MATCH/ANY HELP_DESK_CALL #(1) AND SUPPORT_ENGINEER #(1)

在这个例子中, MATCH 规则使用一个 SUPPORT_ENGINEER 劳动者从 HELP_DESK_CALL 缓冲区取出 1 个零部件。

例 5. MATCH/ANY CLOCK STORE(1) with WORKER(1) using path #(2) AND BOX STORE(2) #(1)
在这个例子中,MATCH 规则从第二个 STORE 元素中取出一个盒子零部件,并 且使用劳动者元素 WORKER(1)从第一个 STORE 元素中取出任意两个钟表零部件沿 着适当的路径移动,并把它们一起输入到机器中去。

(3)下面是 MATCH 作为劳动者规则的例子(假设是机器加工需要人员辅助): 例 1. MATCH/ANY (JOE #(2) AND FRED #(1)) OR (BILL #(2) AND TOM #(1))

在这个例子中, MATCH 规则设定如果有两个 JOE 劳动者与 1 个 FRED 劳动者 同时可以使用,或者 2 个 BILL 劳动者与 1 个 TOM 劳动者同时可以使用,就使用一 组进行辅助操作。

例 2. MATCH ATTRIBUTE TEAM_NO DOCTOR#(1) AND NURSE#(2)

在这个例子中,如果有 TEAM_NO 属性相同的 1 个 DOCTOR 劳动者与 2 个 NURSE 劳动者同时可用,就使用他们 3 个进行辅助操作。

例 3. MATCH/CONDITION SKILL>5 FITTER#(1) OR FOREMAN#(2)

在这个例子中,任意1个 SKILL 属性值大于5的 FITTER 劳动者可以辅助机器 操作,或者任意2个 SKILL 属性值大于5的 FOREMAN 劳动者可以辅助机器进行操 作。

例 4. MATCH/ANY FITTER#(5) AND FOREMAN#(1) using PATH

在这个例子中, MATCH 规则通过特定的路径匹配任意 5 个 FITTER 劳动者与任意 1 个 FOREMAN 劳动者来辅助机器进行操作。

5.3.2.3 WAIT 规则

(1) WAIT 规则定义一个元素如何等待。每一个元素的输入输出规则在默认的情况下,都设为 WAIT 规则。为了指明模型中零部件的走向,我们必须用其他规则来替代它。

在 IF 语句中,劳动者规则非常有用。例如:

IF NPARTS(WIDGET_BUFFER)< 50

PUSH WIDGET to WIDGET_BUFFER

ELSE

4

WAIT

ENDIF

在机器的输入规则这个例子中,如果缓冲区 WIDGET_BUFFER 中的零部件少于 50 个的话,机器把 WIDGET 零部件推入缓冲区内;如果 WIDGET_BUFFER 中的零 部件不少于 50 的话,机器等待,直到缓冲区内的零部件数少于 50。

(2) WAIT 规则的语法如下,它没有其它的参数。

WAIT

WAIT 的输入规则

WAIT 的输入规则指的是元素等待直到元素输入零部件或者流体到其中。下列元素可以用 WAIT 作为输入规则:

| ・机器 | ・输送链 | ・轨道 |
|------|------|-------|
| · 管道 | · 容器 | · 处理器 |



・工作站

· 工作站

WAIT 的输出规则

WAIT 的输出规则指的是元素等待直到其中的零部件或流体被取出。如果没有其他元素从中将它的零部件取出,元素将会永远被阻塞。以下元素可以用 WAIT 作为输出规则:

| ·零部件 | · 单件运输小车 | · 缓冲区 |
|------|----------|-------|
| ·机器 | · 输送链 | ·运载工具 |
| • 轨道 | ·零部件文件 | · 流体 |
| · 管道 | · 容器 | · 管道 |
| ·处理器 | ·工作站 | ·路线集 |
| | | |

·路线集

WAIT 的劳动者规则

WAIT 的劳动者规则指的是元素等待直到劳动者有效。以下元素可用 WAIT 作为劳动 者规则:

| ・机器 | · 输送链 | ・管道 |
|-------|-------|-------|
| ・容器 | ·处理器 | · 停靠站 |
| • 行为站 | · 装载站 | · 卸载站 |

5.3.3 劳动者规则的例子

例1

TWO _TON#1 OR ONE_TON#2

这个例子展示了劳动者作为工具或者资源来使用的情况。元素需要 1 个 TWO_TON 工具或者 2 个 ONE_TON 工具来完成任务,它将按照列表中的次序来选 择劳动者,所以元素将会首先选择前者。

例 2

OPERATOR OR NONE

元素需要一个操作者去执行任务,要是没有操作者可提供,Witness 允许在没有操 作者的情况下继续执行任务。

例 3

NOVICE AND AUTOMATIC OR EXPERT AND MANUAL

这个例子把操作者作为一种资源或者一个工具来使用。元素需要一个初学者和 一个自动工具或者一个专家和一个手动工具一起才能完成任务,它将按照列表中的 次序来选择劳动者,所以元素将会首先选择前者。

例 4

IF NPARTS(PACKING(1)) < 3

NONE

```
ELSEIF NPARTS(PACKING(1)) < 9
```

OPERATOR AND PACK_TOOL



ELSE

OPERATOR#2 AND PACK_TOOL#2 OR ROBOT

ENDIF

ENDIF

如果第一个包装机的零部件的数目小于 3,元素不需要劳动者。如果第一个包 装机的零部件的数目少于 9,但是大于 3,元素需要一个单位的 OPERATOR 劳动者 和一个 PACK_TOOL 工具去执行任务。如果零部件的数目等于 9 或者更多,元素就 需要 2 个单位的 OPERATOR 劳动者和 2 个单位的 PACK_TOOL 工具,要是劳动者和 工具不能被提供,那么用 ROBOT 来代替。



第六章 Witness 程序设计基础

6.1 变量类型

witness 提供了四种类型的变量,用来进行数据处理。它们是整型、实型、名型、 字符型。

6.1.1 整型 (integer)

整型变量用来存储不包含小数点部分的数字。在 witness 中,可以是-2147483648 到+2147483647之间的整数。

使用整数变量能够比较精确的存储数据,并且处理速度比比实数要快。但是由 于整数的"循环"性,可能会使得它们过大或过小。例如:

2 147 483 647 + 1 = -2 147 483 648

-2147483647 - 2 = 2 147 483 647

6.1.2 实型 (real)

实型变量可以存储由数字(0~9)、小数点和正负号组成的数据。范围为(3.4E-38,3.4E38);

6.1.3 名型 (name)

名型(name)变量用来存储 witness 仿真系统组成元素的名称。例如:

widget

miller (3)

注:函数、数值型变量、数值型属性不能够存储为名型数据。

6.1.4 字符型 (string)

字符型变量用来存储不具有计算能力的字符型数据。字符型数据是由汉字和 ASCII 字符集中可打印字符(英文字符、数字字符、空格以及其他专用字符)组成, 长度范围是 0~4095 个字符。

6.1.4.1 字符运算符

- = 比较前后两个字符串是否相同;
- + 连接两个字符串
- = 对字符型数据赋值

如果连接操作得出的字符型数据长度超出长度范围,witness显示出错信息。

6.1.4.2 特殊用途字符串



字符型数据可以存储任何键盘上的字符。反斜线字符(\)却是一个特殊的字符。

\" 向字符串中引入一个引号(")。引号标识字符的结束。

\\ 向字符串中引入一个反斜线 (\)。

\n 向字符串中引入换行符。

\r 向字符串中引入回车。

\t 向字符串中引入 8 个空格 (TAB) 字符。

\f 向字符串中引入走纸字符。如果是打印(PRINT)操作,交互窗口被清空;如果是写(write)操作,将另起一页进行写入。

6.2 运算符及表达式

6.2.1 算术运算符

通过算术运算符可以构成算术表达式,进行数值型数据的处理。算术运算符和 表达式实例见表 6.1。

| | ₹ 6.1 昇 | 本运算符及头例- | ⁻觉表 |
|-----|---------|-------------|--------|
| 运算符 | 功能 | 表达式 | 表达式值 |
| ** | 乘方 | 2**8 , 5**2 | 256,25 |
| *,/ | 乘 , 除 | 36*4/9 | 16 |
| +,- | 加,减 | 5+6 - 7 | 4 |

在进行算术表达式计算时,要遵循以下优先顺序:先括号,在同一括号内,按 照先乘方,在乘除,后加减。

6.2.2 关系运算符

关系运算符用来构成关系表达式,关系运算是运算符两边同类元素的比较,关系成立结果为真(T);反之,结果为假(F),参看表 6.2。

| | | NXLEIVA | 71 2010 |
|------------|------|---------|---------|
| 运算符 | 功能 | 表达式 | 表达式值 |
| \diamond | 不等于 | 15<>20 | Т |
| > | 大于 | 5>8 | F |
| < | 小于 | 8+4>10 | Т |
| <= | 小于等于 | 12<=3* | Т |
| | | 4 | |
| >= | 大于等于 | 16>=20 | F |

表 6.2 关系运算符及表达式实例一览表

6.2.3 逻辑运算符

逻辑运算符用来构成逻辑表达式。逻辑表达式可与关系表达式一起组成满足 IF, WHILE 语句的判断条件,参看表 6.3。



北京威特尼斯科技中心

| | 表 6.3 | 逻辑运算符及表达式实例 | 一览表 |
|-----|-------|----------------|------|
| 运算符 | 功能 | 表达式 | 表达式值 |
| NOT | 逻辑非 | NOT 3+5>6 | F |
| AND | 逻辑与 | 3+5>6 AND | Т |
| | | 4*5=20 | |
| OR | 逻辑或 | 6*9<45 OR 7<>8 | Т |

6.2.4 转换运算符

转换运算符主要有两种:

(1) & 将整数转换为英文字母。

如果变量值为 1,则返回 "A";如果变量值为 2,则返回 "B";如此类推,当 变量值为 26时,则返回 "Z"。如果变量值大于 26,将进行循环,如:变量值为 28, 则返回 "B"。

如果变量值为 - 1,则返回 " a ";如果变量值为 - 2,则返回 " b ";如此类推, 当变量值为 - 26 时,则返回 " z "。如果变量值小于 - 26,将进行循环,如:变量值 为 - 28,则返回 " b "。

(2) @ 将数值型数据转换为字符型数据。

例如:DESC="P"+@VAR1(此时数值型变量 VAR1=56),则 DESC=P56。

6.3 程序三种基本结构

6.3.1 顺序结构

顺序结构是在程序执行时,根据程序中语句的书写顺序依次执行的命令序列。 WITNESS 系统中的大多数命令都可以作为顺序结构中的语句。

6.3.2 分支结构

分支结构是在程序执行时,根据不同的条件,选择执行不同的程序语句,用来 解决有选择、有转移的诸多问题。

分支结构有单向分支和多向分支语法结构,分别如下:

(1) 单向分支,也叫简单分支结构

IF <条件表达式>

<命令行序列>

ENDIF

该语句首先计算<条件表达式>的值,当<条件表达式>的值为真(T)时,执行< 命令行序列>;否则,则执行 ENDIF 后面的命令。

(2) 多向分支,也叫复杂分支结构

IF <条件表达式 1>

<命令行序列 1>



ELSE

<命令行序列 2>

ENDIF

该语句首先计算 条件表达式 1 的值,当 条件表达式 1 的值为真时,执行 命令行序列 1 中的命令;否则,执行 命令行序列 2 中的命令;执行完 命令 行序列 1 或 命令行序列 2 后都将执行 ENDIF 后面的第一条命令。

- (3) 使用分支语句应注意的几点:
- IF..... ENDIF 必须配对使用;
- 条件表达式 可以是各种表达式或函数的组合,其值必须是逻辑值;
- 命令行序列 可以由一个或多个命令组成,也可以是条件控制语句组成的嵌 套结构。

(4)示例

IF (water_level ≥ 0) AND (water_level ≤ 5)

PRINT "The level in the water tank is low"

ELSEIF (water_level > 5) AND (water_level <= 10)

PRINT "The level in the water tank is normal"

ELSE

PRINT "The level in the water tank is high"

ENDIF

这段程序是用来检测容器中的水位的。变量 water_level 记录容器的水位,当水 位在[0,5]之间时,在交互窗口(interact box window)中打印出"The level in the water tank is low"语句,提醒水位较低;如果第一个条件不满足,判断第二个条件表达式, 看水位是否在(5,10]之间,若表达式为真,在交互窗口中打印出"The level in the water tank is normal"语句,提醒水位正常;当条件二也不满足时,唯一的一种情况就是 水位高于 10了,此时在交互窗口中打印出"The level in the water tank is high"语句, 提醒水位超高了。

6.3.3 循环结构

允许有限次重复执行某一特定的程序。

```
(1) 计数型循环
```

基本语法如下:

FOR 循环变量 = 循环变量初值 TO 循环变量终值 [STEP 循环变量步 长]

命令行序列

NEXT

参数说明:

循环变量:一般情况下为整型变量,并且其数量为一,即不是数组;

循环变量初值、循环变量终值:数值型常量、变量或表达式;



循环变量步长:每次步进的长度,可以为整数,也可以为负数;缺省时为1。

语句功能:

该语句用 循环变量 来控制 命令行序列 的执行次数。执行语句时,首先 将 循环变量初值 赋给 循环变量 ,然后判断 循环变量 是否大于或小于 循 环变量终值 ,若结果为"真",则结束循环,执行 ENDFOR 后面的第一条命令;否 则,执行 命令行序列 ,循环变量 自动按 循环变量步长 增加或减少,再重 新判断 循环变量 当前的值是否大于或小于 循环变量终值 ,直到其结果为真。 示例

NUMBER FOUND = 0

FOR BUFFER INDEX = 1 to NPARTS(STORE(1))

IF STORE(1) at BUFFER INDEX:color=red

NUMBER FOUND = NUMBER FOUND + 1

ENDIF

NEXT

这段程序用来统计缓冲区 store(1)中,颜色为红色的部件的数量。变量 NUMBER_FOUND 用来统计红色部件的数量,开始时被置零;BUFFER_INDEX 是 循环变量;NPARTS(STORE(1))函数计算出缓冲区 STORE(1)中的部件数量;IF 分支 用来判断 STORE(1)中排于第 BUFFER_INDEX 位置的部件的属性 color 是否等于 "red",如果为真,统计变量 NUMBER_FOUND 自动加1,如果为假,则执行 ENDIF 后面的语句;执行完成一次 IF......ENDIF 语句后,则循环变量自动增加1,然后判 断循环变量是否大于循环变量终值,如果为假,再执行 IF 语句,如果为真,结束循 环,执行 NEXT 后面的程序,循环结束。

(2)"当"型循环

语法结构:

WHILE <条件表达式>

<命令行序列>

ENDWHILE

注: ENDWHILE 可以缩写为 END。

语句功能:

当<条件表达式>为真时,一直执行<命令行序列>,直到<条件表达式>为假时, 循环结束。

示例:

WHILE NPARTS(STOCK) > 90

PRINT "Warning! The STOCK buffer is nearly full."

ENDWHILE

该段程序实现当缓冲区 STOCK 中的部件数量大于 90 时,在交互窗口发出提醒 信息 "Warning! The STOCK buffer is nearly full."。



- (3)使用循环语句时应注意的几点
- WHILE 和 ENDWHILE、FOR 和 NEXT 必须配对使用。
- 命令行序列 可以是任何 WITNESS 的命令或语句,也可以是循环语句,即 可以嵌套为多重循环。



第七章 可视化仿真项目的设计及运行示例

7.1 流水线仿真系统

下面描述如何通过 WITNESS 系统提供的 Designer Elements 模板,快速的建立 WITNESS 模型。这个过程仅仅展示了采用 WITNESS 建模的思想,它并不代表真正 的工业系统。

通过本节的学习,要能够掌握:

part、machine、conveyor、labor 实体元素、variable 逻辑元素的使用; 掌握可视化输入、输出关系的建立;

掌握 report 工具栏的使用和分析,并根据分析,进行系统优化设计。

7.1.1 引言 (Introduction)

模型的建立采用一种循序渐进的方法,这种建模方法可以在确保本阶段正确无 误的基础上继续进行下一阶段的建模,而且能够清楚地看到在做任何改变产生的效 果。

第一阶段(stage1.mod)和第二阶段(stage2.mod),可以建立一个包含机器和传送装置的简单模型,可以从这一模型计算出此模型系统的输出和性能。

第三阶段 (stage3.mod) 和第四阶段 (stage4.mod), 可以加入更多的功能元素 (例 如机器故障、劳动者、设备调整等) 使模型更符合实际。即使对模型增加一点点复杂性,人工计算生产量和设备利用率也将变的很困难,而使用 WITNESS 的仿真技术可以很轻松的解决这一问题。

第五阶段(stage5.mod)和第六阶段(stage6.mod)为了增加产量和更有效的利用资源可以调整模型的运行方式和参数。

如果并不想建立模型而只是想了解逐步建模的方法,那么只需运行 WITNESS 安装路径下 Demo\Tutorial 中的 stage1.mod~stage6.mod 六个模型文件;如果希望自行建立模型,那么建立的模型文件将会与 stage1.mod~stage6.mod 模型相对应。

7.1.2 模型概述

在模型中,零部件(widget)要经过称重(weigh)冲洗(wash)加工(produce) 和检测(inspect)四个工序的操作。执行完每一步操作后零部件通过充当运输器和 缓存器的输送链传送至下一步操作;经过检测以后零部件脱离模型;同时需要一个 操作人员控制加工机器的各种加工活动。最后完成的模型如图7.1所示:





流水线系统仿真系统布局图

7.1.3 构建第一阶段 (Stage1.mod) 模型

7.1.3.1 定义元素

打开 WITNESS 安装路径下 Demo 文件夹中的建模文档 startup.mod,或者本书附 带光盘文件夹 models 中的的建模文档 startup.mod。

WITNESS 的用户元素窗口(designer elements)允许快速而简易的添加 WITNESS 默认的元素到所要建立的模型中。

下面建立被称为 Widgets 的零部件、一台称重机器(Weigh)和一条输送链(C1)。 点击 designer elements 窗口的机器 machine 图标,使其变成可选项,将光标移 向系统布局窗口 window 1 的位置,然后单击左键,在系统布局窗口出现机器 (Machine001)图标,同时在元素选择窗口 element selector 中的 simulation 页下出现 Machine001:1图标;当鼠标在 window 1 中选中 Machine001,鼠标光标成十字形时, 可以在屏幕范围内拖动元素到所适当的位置,此时 display edit 工具栏必须打开,同 时关于 window1 的 window control 中的 movable display 选项必须选中。现在 Machine001是所要建立的模型的一部分了。

通过点击 designer elements 窗口的输送链 conveyor 图标可在模型中加入输送链。 选中图标将光标移到仿真窗口再次点击创建输送链,然后将其拖到想要的位置。

通过点击 designer elements 窗口的输送链 part 图标可在模型中加入小零件 widget。选中图标将光标移到仿真窗口再次点击创建小零件,然后将其拖到想要的



位置。

现在第一阶段所需的三个仿真元素已经加入模型中了,由于 designer elements 窗 口中的建模元素都设定好了默认的显示 display 属性,在本例中将不加以修改,则下 一步是对元素进行详细设计。

7.1.3.2 建模元素详细设计 (Detailing an element)

简单方法:

改变元素细节最简单方法是在屏幕中的元素图标上双击(即机器图标和输送链图标)。

其他方法如下:

- 点击建模元素图标,然后点击标准工具栏的 detail elements 图标 ;
- element selector 窗口中找到所需建模元素点击鼠标右键,选择弹出菜单中的 detail...。
- (1) 零部件明细 (Part detail information)
- 双击 PART001 得到元素细节设计对话框。
- 输入新的元素名 Widget 覆盖掉系统默认的名字。
- 点击 OK 键确认。
- (2) 机器明细 (Machine detail information)
- 双击 Machine001 图标, 输入以下信息:

名字 name: Weigh

加工时间 cycle time: 5

点击对话框中的 OK 键确认。

- (3) 输送带明细 (Conveyor detail information)
- 双击 Conveyor001 图标输入:

| 🥪 Window 1 | | | |
|------------|---------|---------|----------|
| | | | _ |
| | | | |
| | | | |
| Widget | Weigh | C1 | |
| - | - 🔶 🗖 👘 | | |
| | | | |
| | | | - |
| • | | | • // |

图 7.2 stage1 模型布局图

名字:C1

4

点击 OK 键确认。

设计完毕,窗口显示如图 7.2。

此阶段需要做的是输入机器加工时间、改变元素的名字,其实建模元素的所有 活动都能通过这种方式定义的。

7.1.3.3 建立元素之间的逻辑规则

接下来定义各个元素之间链接的逻辑规则,规则输入可以通过以下两种方法: 一是通过工具栏和鼠标,一是通过元素细节对话框。

常用规则包含推(输出规则) 拉(输入规则) 百分率和顺序规则;非常用规则通常都是根据匹配的属性和环境而定义的复杂控制规则,比如:" if " 命令来实现 有复杂的选择规则。

(1) 机器规则明细 (Machine detail rule information)

- 点击选中 Weigh 图标,然后单击 element 工具栏中的 visual input rule 图标
 出现 input rule for weigh 对话框,如图 7.3 所示;
- 规则文本框的缺省值为 pull
- 在规则文本框中输入"PULL Widget out of WORLD", 定义了机器 Weigh 加工 完成一个 Widget 之后,从本系统模型的外部 WORLD 处拉进一个 Widget 进行 加工。规则定义结果显示如图 7.3 所示;

| 😅 Input Rule for | Weigh | | | | × |
|------------------|---------------------|---|------------------------|------|-------|
| Bule: FULL V | Operation Cycle | • | Yorld Ship Sgrap Rogte | Arrb | OK |
| 10 CP PULL | Widget out of WORLD | | | Edit | Close |

图 7.3 机器规则定义对话框

- 单击 OK 键确认。
- 然后单击 visual output rule 图标 😤;
- 点击 C1 图标, 点击 OK 键确认。
- (2) 输送带明细 (Conveyor detail rule information)
- 点击输送带 C1 的图标,选中 C1;
- 单击 element 工具栏中的 visual output rule 图标 ^术;
- 点击 output rule for C1 对话框中的 SHIP 按钮,为输送链 C1 创建输出规则 PUSH SHIP,将 Widget 发运出去,即输送带将 Widget 送出本系统之外。
- 点击 rules action bar 菜单条中的 OK 按钮确认

7.1.3.4 运行模型(Running the model)



首先介绍运行工具栏中按钮及其作用,运行工具栏如图7.4所示。



图 7.4 运行工具栏示意图

运行工具栏中的第一个按钮 reset 进行仿真的复位操作,点击该按钮,系统仿真 时钟和逻辑型元素(变量、属性、函数)的值将置零;step 按钮控制模型以步进的 方式运行,同时在 interact box 窗口中显示仿真时刻所发生的事件,便于理解和调试 模型;run 按钮控制模型的连续运行,如果没有设定运行时间,模型将一直运行下去, 直到按 stop 按钮,如果设定了运行时间,模型连续运行到终止时刻;stop run at 包括 一个按钮和一个输入框,用来设定仿真运行时间,按钮决定仿真是否受到输入框中 的输入时间点控制,输入框输入时间点;walk on/off 包括一个按钮和一个滑动条, 用来设定仿真连续运行时,仿真运行的速度。

本例中,在运行工具栏 run toolbar 中按下 stop run at 按钮,在输入框中输入模型运行终止时间 100。然后点击 run 按钮开始运行模型。

WITNESS 内置安全系统,在运行模型时,如果缺少重要数据,将弹出提示和数据输入对话框,从而保护模型。如果没有输入输送带的移动速度 index time(即输送带将小零件向前移动一个零件的长度所需要的时间),当仿真开始运行时,会出现提示信息要求输入移动速度,输入:

- Index time : 0.5
- 点击 OK 确认。

7.1.3.5 结果预测(Anticipated results)

已知输送带的长度等于在输送带上的 10 个零部件的长度,也就是输送带可以连续排列 10 零部件,所以每个零件在输送链上经历的时间为 0.5*10mins=5mins.

- 在整个过程中处理1个零件需10mins:
- 10mins=5mins 称重时间(=weigh 加工时间)+5mins(在 C1 上的时间)
 每 5mins 有一个零件到达(由 Cycle Time of weigh 控制),因此,可以推算出如

Ŵ

北京威特尼斯科技中心

果模型运行 100 分钟会有 19widgets 被加工完成。

如果在模型运行完 100mins 后检查 widget 的记录会发现与预测结果一致。

7.1.3.6 修正模型显示(Modifying the model display)

可能你想修正模型中建模元素的图标显示使其看起来更加真实。如果不想改变 所建模型的显示可以略过这一部分内容,直接进入第二阶段,但是这部分内容介绍 了显示的另一个层面,有助于理解 WITNESS 在这一领域所能提供的强大功能。

用鼠标在屏幕内选择并拖动元素图标,而且还可以用鼠标同时框住几个元素, 只要拖动框就可以同时拖动被选的几个元素。

WITNESS 允许锁定窗口或显示层,也允许链接一个元素的其他显示条目。锁定 图标通过以下操作:点 view/layers 或 window/control 选项。为元素链接显示条目需 用 display 的下拉菜单(详见 Quick Reference Book 或在线求助)



7.5 元素显示工具栏

元素显示工具栏 (如图 7.5)的调用可用以下方法:

- 鼠标右键点击元素图标显示元素菜单项,然后选择 Display (见图 7.6);
- 在 element selector 窗口中的仿真元素上单击鼠标右键也可打开元件菜单,然后 选择 Display;
- 选中元素,然后点击 element 工具栏中的 display 按钮。



图 7.6 元素右键弹出式菜单

图 7.5 的元素显示工具栏中一共有九个对象,下面从左到右,分别介绍这九个



对象如下:

- 显示设计模式下拉列表框:有两个选项 draw 和 update,第一次设计建模元素的 某一显示属性时,选择 Draw 项;对已经设计了的显示属性进行修改设计时,选择 update 项。
- 属性下拉列表框:选择用于 draw 或 update 的元素的属性项,该列表框中的属 性项目随着元素类型的不同而有所不同。可以通过更改属性项的"label"来更 改属性项的名称,当我们多次使用同一类型的显示属性项时,通过改变它们的 名称以便于识别。例如一个元素可能有多个 icon 显示属性项,可以通过改变其 名称来加以辨认。
- Draw 按钮:用于激活显示设计对话框,来进行显示属性的绘制和更改;
- Erase 按钮:用于激活删除显示属性项对话框,来进行属性项的删除;
- □□ Layers 按钮:用于激活层设计对话框,来进行层的可移动性和可视性设计;
- Lock 按钮:用于设定元素显示属性项的锁定状态,该按钮将在三种状态之间进行切换(Lock、Unlock、Superlock)。

DUnlock:在此状态下,元素的可视显示属性项可以在系统布局区单独移动;

Lock:在此状态下,元素的可视显示属性项将作为一个整体在系统布局区 移动;只有显示属性项在同一层时,才能够锁定;

Superlock:当设计的元素类型为模块 Module 时,在此状态下,该模块内的
所有的元素的显示属性项将作为一个整体在系统布局区移动;

- Grid 按钮:用于设定显示对象的位置是否捕捉屏幕上的网格,以便进行精确的定位;
- **『**Help 按钮:激活 WITNESS 帮助文件;
- OK 按钮:当对一个元素的显示属性设置完毕,点击该按钮关闭显示工具栏。
 在进行元素显示属性设计的时候,一般步骤如下:
- 从显示设计模式下拉列表框中选择 Draw 或 Update;
- 在属性下拉列表框中选择所要设计的属性项,如机器的 " name "、 " icon "、 " part queue " 等;
- 点击 / 按钮进入显示项目的详细设定对话框,进行设定即可。



元素图标的移动和删除可以使用鼠标来完成。用鼠标左键选中要删除的图标, 在图标上点击右键进入弹出式菜单,选择 Delete Graphics选项,图标被删除。

元素图标尺寸大小的改变通过鼠标 + CTRL 键来完成。用左键选中图标,按住 CTRL 键,拖动图标周围的可改变大小的小方框,小方框只出现在可以改变大小的 图标上,也可以用 View/Graphical Editing 菜单项编辑图标。

7.1.4 构建第二阶段 (Stage2) 模型

现在,已经建立包含一台机器和一条输送带的仿真系统模型,而且运行并测试 了模型的运行情况,在 stage2.mod 中将在此基础上,添加更多的机器和输送链,构 建一个比较复杂的仿真系统。

使用逐步建模方法具有很重要的意义。它能够避免错误的发生和潜在问题的出现,能够使建模者更加清楚地知道发生了什么事情,还可以在任何时间改变模型而 不需要重新建模。

本阶段需要添加的机器为清洗(wash)加工(produce)检测(inspect),添加的输送带为C1、C2、C3,同时添加了一个逻辑元素 变量 output,用于动态显示 模型中加工完成的小零件的数量。

在本阶段模型构建过程中,对每个元素的定义、显示、明细设计三个步骤融入 到一起,不再分开来编写。

7.1.4.1 元素设计步骤

- 在 designer elements 窗口点击 machine 项在 C1 的末尾加入另一台机器。双击机 器图标,输入:
 - ◆ 名字 name: Wash
 - ◆ 加工时间 cycle time:4
- 在 Wash 后加入另一条输送链。双击其图标,输入:
 - ◆ 名字 name : C2
 - ◆ 移动速度 index time: 0.5
- 在 C2 的末尾添加一台机器。双击机器图标,输入:
 - ◆ 名字 name: Produce
 - ◆ 加工时间 cycle time: 3
- 加入另一条输送链。双击其图标,输入:
 - ◆ 名字 name : C3
 - ◆ 移动速度 index time: 0.5
- 加入最后一台机器输入:
 - ◆ 名字 name : Inspect
 - ◆ 加工时间 cycle time:3
- 在 designer elements 窗口点击 Vinteger(整数变量)图标,创建一个变量用来记录 和显示 Inspect 机器的产量。



- ▶ 双击变量图标将名字改为 Output,点击 OK 确认。
- ◆ 改变变量字体颜色:鼠标右键点击 Output 选择弹出式菜单中的 display 菜 单项,从 display 对话框的下拉菜单中选择 name 属性项,然后点击 pencil

按钮,激活 name 显示设计对话框,点击字体前景色颜色按钮 → (按钮 在标签栏下方),从调色板中选择任意想要的颜色,点 OK 键确认。

◆ 改变变量字体:在改变字体颜色之后,在通过 name 显示设计对话框中的

字体按钮,可以选择所需要的字体。

- ◆ 点击更新 update 按钮更新显示的字体和颜色。
- 设计变量细节 (Variable detail)

变量 output 用来计算从 Inspect 中输出的 Widgets 的量,将计数结果显示在屏幕上。可以通过设计 Inspect 机器的 detail 来实现,具体步骤如下:

- ◆ 选中 Inspect 机器,双击其图标;
- ◆ 点击细节对话框中 actions on finish 按钮;
- ◆ 在规则编辑框中输入语句:output=output+1;
- ◆ 点击 OK 确认

这样,每当 inspect 机器加工完成一个 widget,将触发完成活动 actions on finish 中的代码,变量 output 的值将累加1,并在模型布局窗口显示出来。

● 规则(Rules)设定

为了使模型顺利运行,最后一步所做的工作与第一阶段中所做的工作相同,即 键入控制零件流的输入和输出规则:

- ▲ 点击 C1 图标选中;
- ◆ 点击 visual output rule ^え按钮;
- ◆ 在编辑栏中删除 ship 项,然后点击 Wash 机器;
- ◆ 点击 OK 确认

然后设定新建机器的规则。

首先是 Wash:

◆ 利用 visual output rule ★ 按钮创建规则: push C2; 其次是 PRODUCE

◆ 利用 visual output rule [▲] 按钮创建规则: push C3; 最后是 INSPECT:



- ◆ 利用 visual input rule 📝 按钮创建规则: pull C3;
- ◆ 利用 visual output rule [▲] 按钮创建规则: push ship

如果有必要可重新安排元素在屏幕中的位置,结果如图7.7所示。



图 7.7 stage2 模型布局图

● 模型的标志键图标 (Displaying status keys for the model)

通常为了用不同颜色显示元素的不同状态,在模型布局的边缘区域设置标志键, 可以起到提示作用。设置步骤如下:

- ◆ 选择 view/keys 选项;
- ◆ 选择 machine 图标,改变背景颜色使之与窗口颜色相匹配,然后点击 draw 按钮,当光标变成十字尖头的形状,将光标移动到屏幕左上角要设置键 (key)位置处;
- 点鼠标左键创建出所需要的标志键。

7.1.4.2 结果预测(Anticipated results)

已知输送链的长度等于在输送链上的 10 个零件的间距,所以每个零件在输送链上经历的时间为: 0.5*10=5mins。

一个零件在整个运行过程中需要的时间为 30mins

- =5mins 在 weigh 上 (=循环时间)
- +5mins在C1上时间



+4mins 冲洗机上的时间(=冲洗时间)

+5mins 在 C2 上时间(输送链长*循环时间)

+3mins 在加工机上的时间(加工循环时间)

+5mins在C3上时间(输送链长*循环时间)

+3mins 在检测机上时间(=检测循环时间)

模型每 5mins 加工一个 widget (weigh 每 5mins 循环一次"瓶颈",决定了整个 系统的节拍)。

通过预测可知,如果模型运行 100mins 可以加工出 15 个 widgets。

点击 begin 按钮清零,使模型运行 100mins,检查 widget 数目,可以在 Number shipped 中读出零件数量为 15,表明模型在 100mins 内成功加工了 15widgets,与之 相应的变量 output 显示为 15。统计 widget 可以看出平均在制品库存 Ave W.I.P 为 5.21, 平均通过时间 Ave Time 为 24.83。

以上就是 stage2 模型的建立。

7.1.5 构建第三阶段 (Stage3) 模型

为了使上述模型更有现实意义需要对其赋予更多的特性和功能。在本阶段中, 将假设 Produce 机器每加工完五个零部件就需要进行一次刀具的调整,调整时需要 人员来参与,调整时间为 12 分钟。

构建本阶段模型需要在 stage2 的基础上,向模型中添加 Labor 元素,设置 Produce 机器的调整属性。

7.1.5.1 机器 setup 页框说明

通过双击机器图标,得到机器的详细设计对话框,选择 setup 页框。对话框如 图 7.8 所示,可以细分为调整项目、调整模式、调整间隔、调整时间四块,分别介绍 如下:

- 调整项目:可以进行调整项目的添加和删除Add/Remove...、调整项目的选择 setup 下拉列表框、一台机器的调整项目汇总 Summarize...;
- 调整模式:witness 对机器划分了三种调整模式,(1) No. of operations,在经过 了指定次数的操作后,机器要进行一次调整。(2) Part change,如果机器更换加 工零件的种类,则需要进行一定的调整。例如不同的零件需要不同的刀具。(3) Value change,当某一指定的变量值发生变化时,机器需要进行调整。
- 调整间隔:在设定机器的调整模式为 No. of operations 之后,决定调整间隔的数据有两项,一项是 Number of operations,指定两次调整之间的作业次数;一项是 Ops to first setup,指定第一次调整发生的操作。设定机器的调整模式为 Value change 之后,决定调整间隔的数据是 expression,需要输入一个变量名。
- 调整时间:用来指定机器的调整需要花费多长的时间以及需要怎样的劳动者协助。时间值直接输入到 Setup time 下的文本框,劳动者设定需要通过点击"Labor"按钮,在弹出式 Labor rules 编辑框中进行编写。



| C | Detail Machine - Produce | × |
|---|---|---|
| : | General Setup Breakdowns Fluid Bales Shift Actions Reporting Notes Setup: Setup Humber 1 Setup Humber 1 Setup Mode: Setup Mode: Setup Interval Sol of Setup Interval Sol of Setup Interval Bo. of Das to Eirst Bo. of Operations H盤问篇 | × |
| | 現泊 帮助 | |

图 7.8 机器 setup 页框示意图

7.1.5.2 构建模型

- 添加和设计 labor 型元素
 - ◆ 从 designer elements 窗口中找到 labor 元素将其加入模型;
 - ◆ 双击 labor001 图标得到元素明细对话框;
 - ◆ 将其名字改为 Operator,即 labor Operator。
- 加工机器调整设置(Machine Produce Setup detail)
 - ◆ 双击 Produce 图标得到对话框;
 - ◆ 从对话框中选择 setup 页框;
 - ◆ 点击 add/remove...按钮进行调整的详细信息设置,本例中添加一个调整描述 setup Description: Setup Number 1;
 - ◆ 点击 OK 确认,返回 setup 页框;
 - ◆ 设置 setup Number 1 如下:

调整模式 setup mode: no.of operations;

调整间隔次数 No.of:5;

调整时间 setup time: 12.0;

labor 设定过程为:选择 labor rule 按钮,在编辑框中输入规则。默认值为 NONE, 输入" operator "即可,点击 OK 确认。这样每次需要调整时,机器将需要名叫 operator 的操作者。

经过以上设置 Produce Machine 每 5 个循环操作就需要调整一次,也就是说,在



5widgets 加工完成以后 operator 需要花 12mins 来调整机器,以便机器能够正常运行下去。

7.1.5.3 结果预测

将模型再次运行 100mins (运行前先要复位),查看所有元素的运行情况统计报 表。在布局窗口中框选所有的模型元素,点击 reporting 工具栏中的 statistics report 按 钮,将弹出安装元素类型分类的统计报表,要想查看其他类别元素的统计情况,使 用右边的"》"或"《"键进行转换。

注意由于重新设置了 Produce 机器的调整时间,使得 Produce 机器成为了"瓶颈",加工完成的 widgets 的数量将会下降到 12。修改机器的调整时间值,结果可能为 13widgets。

Labor 元素 Operator 空闲状态占 76%, 共进行了两次辅助操作。

统计 widget 可以看出 Ave W.I.P 为 6.12, Ave Time 为 29.14。分别比 stage2 增加 了 17.46%和 17.35%。

通过表 7.1 输送链统计报表可以看出,三条输送链可能会出现空载或阻塞。 表 7.1 输送链统计报表

| Nam e | % Empty | % Move | % Blocked | Now On | Total On | Avg Size | Avg Time |
|----------|------------|-----------|-----------|-----------|----------|----------|----------|
| C1 | 5 | 84.5 | 10.5 | 2 | 20 | 1.13 | 5.63 |
| C2 | 15.5 | 56 | 28.5 | 2 | 17 | 1.47 | 8.65 |
| C3 | 44.5 | 55.5 | 0 | 1 | 14 | 0.68 | 4.82 |

在这样的系统流程下,通过方程是难以计算系统的运行效果的,但是通过仿真 就很容易的得出复杂系统的运行状态和结果。

以上为第三阶段的建模操作过程。

7.1.6 构建第四阶段 (Stage4) 模型

在本阶段中,考虑将随机分布函数应用到机器的故障发生时间间隔和劳动者维 修机器的故障所需要的维修时间中。假设 Produce 机器在工作一定的时间后,可能 会发生意外的抛锚,通过以往机器两次抛锚时间间隔的统计发现,其时间间隔服从 均值为 60 分钟的负指数分布;每当机器抛锚时,都需要人员对它进行维修,维修过 程所持续的时间受到故障诊断时间、故障排除的难易程序、维修人员的生理和心理 状态的影响,呈现随机波动性,统计数据表明维修时间服从均值为 10 分钟、标准差 为 2 分钟的对数正态分布。

构建本阶段模型需要在 stage3 的基础上,设置 Produce 机器的故障 breakdown 属性。

7.1.6.1 机器 breakdown 页框说明

通过双击机器图标,得到机器的详细设计对话框,选择 breakdown 页框。对话



框界面如图 7.9 所示,可以细分为抛锚项目、抛锚模式、抛锚间隔、维修时间四块分 别介绍如下:

| Detail Machine - Produce | × |
|--|---|
| Detail Machine - Produce Detail Machine - Produce General Setup Breakdowns Fluid Rules Shift Actions Reporting Notes Breakdown: Breakdown Number 1 With项目列表 Summarize Breakdown Momber 1 Image: Check Only At Start Of Cycl Breakdown Mode: Breakdown Duration Busy Time Image: Default Action Section Image: Default Action Section | |
| 他错模式 Breakdown Interval Time Between NEGEXF (60.0.1) 地锚间隔时间 | |
| 确定 取消 帮助 | |

图 7.9 机器 breakdown 页框示意图

- 抛锚项目:可以进行抛锚项目的添加和删除 Add/Remove...、抛锚项目的选择 breakdown下拉列表框、一台机器的抛锚项目汇总 Summarize...;以及 Check only at start of cycle选项,如果选择了该选项,WITNESS 在机器的每次加工周期开 始时进行抛锚时间检查,假设在仿真钟为 70 时,机器输入了一个新的零部件准 备加工,但是发现在 71 分钟这一时间点上,有一个抛锚事件,即发现在该加工 周期内会有抛锚事件发生,则强迫抛锚事件在加工周期开始前发生并进行相应 的维修活动,维修结束后,机器才开始对零件进行 3 分钟加工;如果不选择该 选项,抛锚会发生在加工周期内的任意时间点上,则同样的情况下,机器将先 对零件加工 1 分钟,然后被抛锚中断,在抛锚事件结束后,继续对零件加工 2 分钟。
- 抛锚模式:witness 对机器划分了三种抛锚模式,(1)Available time,在这种模式下,WITNESS 累计机器在系统中的所有时间,既包括加工时间也包括等待零部件的时间,既包括当班时间也包括下班时间,一旦达到指定的抛锚时间间隔,则不论是否被使用,机器都将发生抛锚事件。(2)Busy time,在这种模式下,WITNESS 只累计机器的加工时间,也就是机器的 busy state 时间。一旦指定的抛锚时间逝去,将发生抛锚事件。使用这一模式,机器仅仅会在其加工状态或即将要进行加工时,才会发生抛锚。(3)Operations,在这种模式下,WITNESS 累计机器从刚刚发生的抛锚事件结束到现在已经完成的加工零件数量,如果数



量达到指定的抛锚间隔数量,则机器将发生抛锚事件。在这种模式下,抛锚将 发生在加工结束的时间点上。

- 抛锚间隔:输入时间值或表达式,来设定机器前后两次抛锚的时间间隔。模式 为 Available time 或 Busy time 时,决定抛锚间隔的数据为 Time between failures, 用来指定两次失效的时间间隔;模式为 Operations 时,决定抛锚间隔的数据为 Number of operations,用来指定两次失效的产成品的数量间隔。注:WITNESS 可以接受确定的数值、表达式以及随机分布作为抛锚的间隔。
- 维修时间:用来指定机器的抛锚需要花费多长的时间来进行维修以及需要怎样 的劳动者协助。时间值直接输入到 Repair time 下的文本框,劳动者设定需要通 过点击"Labor"按钮,在弹出式Labor rules 编辑框中进行编写。

注:起始抛锚应该发生在什么时间点?在仿真开始初始化运行时,还没有发生抛锚 事件,WITNESS 既不知道究竟有没有发生了抛锚事件,也不知道其发生的时间,这 样 WITNESS 将第一个抛锚间隔减半,由于仿真初始时钟为0,所以将在抛锚时间间 隔数值的 1/2 时点处发生抛锚事件。

7.1.6.2 构建模型

在本阶段示例中,为了设定模型的随机性特征,在第三阶段模型的基础上,进行 produce 机器故障细节的设计,步骤如下:

- 双击 Produce 机器图标,选择 Breakdown 页框;
- 点击 add/remove 按钮,用 add 项添加故障项目,缺省值 breakdown number 1;
- 将 breakdown mode 改变成 busy time;
- 点击 labor rule 按钮输入需要的规则;
- 删除默认值输入 operator;
- 点击 edit labor rule 对话框中的 OK 键确认;

将鼠标移到 breakdown interval 窗口的 time between failures 字段,现在可以使用 assistant 工具栏,点击 view/toolbars 菜单将其激活,然后点击 assistant。

- 点击 assistant 工具栏中的 distributions 按钮
- 选择 NEGEXP 分布,点击 prompt 按钮
- 输入如下参数: Mean=60, PRN stream=1
- 点击 OK 确认
- 点击 repair time
- 点击 assistant 工具栏中的 distributions 按钮
- 点击 Lognorml 分布, 然后点击 prompt
- 输入以下参数: Mean=10, Standard Deviation=2, PRN stream=2
- 点击 OK 确认
- 点击 OK 确认关闭对话框 以上操作意味着为 Produce 机器设定了如下抛锚细节:
- time between breakdowns 以 60 为均值, 1 为伪随机流的负指数分布;



▶ repair time 为以 10 为均值, 2 为标准差, 2 为伪随机流的对数正态分布。

现在已经了解了故障的产生方式和维修方式,为了确认几个故障的排除应该使 模型运行一段较长的时间,例如 500mins,这样才能够很清楚地看出在模型运行的整 个过程中操作者和建模元素之间的相互影响。

7.1.6.3 仿真运行及统计分析

按照 stage3 中一样的步骤运行模型:使用鼠标按下 Run 工具栏上的 stop run at 时钟按钮,在与其相邻的文本框内输入 500,点击 reset 按钮(指向左边的双尖头), 点击 run 按钮(指向右边的单尖头)运行模型。

仿真结束后,点选元素然后点击 report 工具栏中的 statistics report 按钮,检查每 个元素的报表统计项。可以从统计记录中发现现在系统的主要问题是当 Produce 机 器发生故障而停机时,导致了前面的流程阻塞,从而影响了工序的正常运行,以及 影响了 inspect 机器的检测速度,而且 C3 上的容量也只有正常运行时的一半。有关 机器的统计数据参看表 7.2。

| Name | % Idle | % Busy | % Blocked | % Setup | % Broken Down | No. Of Operations |
|---------|--------|--------|-----------|---------|---------------|-------------------|
| Weigh | 0 | 87.23 | 12.77 | 0 | 0 | 87 |
| Wash | 3.3 | 65.23 | 31.47 | 0 | 0 | 81 |
| Produce | 4.5 | 45 | 0 | 33.63 | 16.87 | 75 |
| Inspect | 56.57 | 43.43 | 0 | 0 | 0 | 72 |

表 7.2 机器运行统计报表

此外,还会看到如下情况:

- Operator 大约有一半的闲置时间,一共完成了 23 次设备调整和维修;
- 只生产了 72widgets;
- 统计 widget 可以看出 Ave W.I.P 为 12.06, Ave Time 为 68.51, 分别比 stage3 增加了 97.05%和 135.1%。
 此即为第 4 步建立的模型。

7.1.7 构建第五阶段(Stage5)模型

通过比较分析可以看出机器 produce 是生产线的"瓶颈",严重制约了系统的产出率,大大增加了 widget 的系统通过时间,为了改进这种状态,考虑添加一台 produce 机器和一条 C2 输送带,这样生产线上将有两台 Produce 机器和两条 C2 输送链同时工作,应该能够提高系统的产量。

注:可以在模型运行的任何仿真时间点实施改善,而不仅仅是在仿真运行的初始时 间点,但是在进行改善设计之前必须先停止模型的运行。

在添加元素之前,建议重新布局屏幕中建模元素,目的是给将要添加的第二台 Produce 和 C2 留出足够的空间,以免显得拥挤,为了节约空间以及符合实际情况, 可以选择将新元素拖到原来建模元素的下方。



7.1.7.1 添加新元素并进行相应的设计

添加这两种元素,操作非常简单。

- 双击 C2 图标显示 C2 明细对话框,
 - ◆ 输入数量 quantity: 2;
 - ◆ 点击 OK 确认。
- 双击 Produce 图标,显示明细对话框
 - ◆ 输入数量 quantity: 2;
 - ◆ 点击 OK 确认。

经过以上设定,模型系统中将有两条输送链 C2(1)和 C2(2),它们的名称、类型 等明细是相同的,只是下标不同,分别为 1 和 2;以及两台机器 Produce(1)和 Produce(2)。系统的输入输出关系设定为 Produce(1)仅仅向 C2(1)"拉"零件来加工, Produce(2)仅仅向 C2(2)"拉"零件来加工。给每台 Produce 机器与相对应的输送链 C2 设定输入输出规则,需要使用到系统变量 N。

系统变量 N:保存当前元素下标的整型变量。例如,如果模型系统中有一个数量为3的机器 MACH,当 MACH(1)恰好完成了一个加工周期(或即将开始一个加工周期时),则 N 被赋值为1;如果是 MACH(2),则 N 被赋值为2;如果是MACH(3),则 N 被赋值为3。

为了实现 Produce(1)仅仅向 C2(1)" 拉 "零件来加工, Produce(2)仅仅向 C2(2)" 拉 " 零件来加工, 需要进行下面的步骤:

| 🥪 Edit Input Rule For Machine Produce | × |
|---------------------------------------|----------------|
| S <u>e</u> lect <u>S</u> earch Editor | |
| PULL from C2(N) at Front | <u> </u> |
| | - |
| <u> </u> | Þ |
| No other elements feed Produce | Line: 1 Col: 1 |
| <u>V</u> alidate <u>P</u> rompt | Help |
| Messages: | |
| | OK |
| | Cancel |

图 7.10 输入规则编辑框

- ◆ 双击 Produce 机器图标显示 general detail 对话框;
- ◆ 点击对话框中的 From...按钮, 弹出机器的输入规则编辑框如图 7.10 所示;
- ◆ 输入规则 "PULL from C2(N) at Front ";
- ◆ 点击 OK 确认 。

同时 WASH 机器上零件清洗完毕之后,将输出到 C2 两条链上队列较短的输送



链上。规则设计操作如下:

- ◆ 点击 output 窗口的 To 按钮;
- ◆ 删除窗口顶部的默认规则, 输入: Least PARTS C1(1), C2(2);
- ◆ 点击 OK 键确认;
- ◆ 点击 OK 确认以上操作。

7.1.7.2 模型运行与分析

所设置的规则使得 widget 从最闲置的输送链向整个生产线上展开。下面使模型 运行 500 时间单位 (运行前要复位),观察改进后的产量,比较为增加产量的所需的 花费是否值得?从统计报表中可以看出:

- 生产了 94 个 widgets,比 stage4 增长了 30.5%;
- Operator 只有 38%的闲置时间, 工作效率提高了 3.5 个百分点;
- 统计 widget 可以看出 Ave W.I.P 为 7.39, Ave Time 为 36.58, 分别是 stage4 的 61.3
 %和 53.4%。

如果有相关机器和输送链的成本、产品的利润,就可以比较改善是否具有经济 性了。

7.1.8 构建第六阶段 (Stage6) 模型

通过 stage5,添加瓶颈设备,可以有效地提高系统的产量,但是 widget 的在制 品库存与平均通过时间仍然比 stage2 要高很多,我们试图在 stage6 中加以改善。

同时在建模过程中得到准确的调整时间和其他的关键信息是很困难的,这样了 解所建立的模型对特定的输入数据的敏感性将是很有意义的。

下面尝试增加 produce 机器抛锚的维修时间 Repair time, 观察模型维修时间值的 改变对产量的的敏感性。

- 双击 Produce 图标显示明细对话框;
- 选择 Breakdown 页框,将 Repair time 的均值由原来的 10 增加到 20,如下:
 - ◆ LOGNORML(20,2,2)

再在 batch 模式下运行模型 500 时间单位(运行前复位), 然后检查输出结果统 计报表:

- 一共生产了 93 个 widgets;
- Operator 有 17%的闲置时间;

由结果可知维修时间均值从 10 增加为 20,只对产量产生很小的影响。下面考虑 继续提高维修时间均值。

- 双击 Produce 图标显示 general 细节对话框;
- 选择 repair time 将平均时间由 20 改为 30, 如下:
 - ◆ LOGNORMOL (30, 2, 2)

在 batch 模式下运行模型 500 时间单位,检查输出记录:



- 共生产了 83 个 widgets ;
- Operator 闲置时间为 8%;

从结果中可以看出 Produce 机器的 repair time 在 20mins 以内变化时,产量相应 变化不敏感;当超过 20mins 时, repair time 的变化将引起产量的较大变动,所以 repair time 范围应该尽量控制在 20mins 以下。

此即为第6步所建立的模型。

在本节中,模型的构建是通过 WITNESS 系统提供的 Designer Elements 模板以 及缺省的布局窗口来构建的,但是,为了形象的代表实际系统中元素的可视化特征, 就需要自定义布局窗口的背景色、建模元素的图标了。下面的示例模型都将使用到 自定义可视化内容。

7.2 装配模型

7.2.1 模型流程概述



7.11 装配模型的可视化界面

一台装配机器把三种零部件单元组装起来,形成一个组装元件。对应示例模型 程序随书光盘的 models/assemblemachine.mod,在建模过程中请参照该模型,模型运 行一段时间后的可视化界面如图 7.11 所示。

7.2.2 定义元素

● 通过菜单项 Window/Control...修改布局窗口的背景色为蓝色, 名称 name 为

W

Assembly Machine.

- 建立本模型,通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过 元素定义窗口定义下列元素:
 - part : Top, Bottom, Screw, Assembled
 - buffer : Store (3)
 - machine : Assembly_machine

在定义元素过程中,需要注意在元素定义窗口中的两个选择问题:

(1) 父对象 (parent) 列表中选定 Simulation;这样定义的元素将是仿真项目的 子对象;如果选择的是 Designer,则定义的元素将是 Designer Elements 窗口的子对 象。

(2) 定义另一种元素时,注意变换类型 Element Type。

通过上述设定之后,在元素最左边的 Element Selector 中的 Simulation 列表下, 将出现六个对象:Top、Bottom、Screw、Assembled、 Store:3、Assembly_machine, 但是在 Assembly Machine 系统布局窗口中,没有出现任何可视化的图标或对象。为 了使得定义的对象以可视化效果呈现在系统布局窗口中,需要进行下一步操作 Display Elements,可视化元素。

7.2.3 可视化元素

7.2.3.1 可视化 part

这里有四个零部件元素:红色的 Top、白色的 Bottom、绿色的 Screw、以及中间为白色,边框为红色的 Assembled。

通过菜单项 View/Picture Gallery...,查看系统图片库,其中没有该模型需要的图标文件,需要自定义图标。

自定义图标方法有下面两种:

- 选中 Picture Gallery...中的一个位置,点击 Picture Gallery...菜单项 Edit/Editor..., 打开图形编辑窗口,进行图标的绘制。
- 选中 Picture Gallery...中的一个位置,点击 Picture Gallery...菜单项 File/Import, 打开随书光盘 pictures 目录,找到 top.bmp、 bottom.bmp、 screw.bmp、 assembled.bmp、 assembly_machine.bmp,将它们逐个导入到当前图库当中,以 便使用。

假设我们定义图标采用的是第二种方法,将这五个图片分别导入在当前图库的 1,2,3,4,5五个位置上。

在元素最左边的 Element Selector 中的 Simulation 列表下,鼠标右击各个元素, 选择弹出式菜单中的 display 菜单项,将弹出可视化对话框,分别在 draw 模式下, 设定它们的 name、icon 属性。现以 top 为例,说明零部件的可视化设计步骤。

- 右击 top;
- 选择 display 菜单项,将弹出 display part 对话框,如图 7.12 所示;



- 选择可视化对话框的 draw 模式;
- 选择第二个列表框中的 name 属性项;
- 点击 pencil 按钮,将弹出 display name 对话框;

| 🥪 Display Part - Top | | × |
|----------------------|---|---|
| Draw Name | - <u>, </u> | |
| | $\overline{\mathcal{L}}$ | |
| (1) 2 | 3 | হ |

7.12 display part 对话框

- 在 display name 对话框中可以设定本次设置的标题 label,和字体颜色,采用缺 省值,点击 draw 按钮,在系统布局窗口的适当位置按下鼠标,将出现 top 三个 字符。
- 在 display part 对话框的第二个列表中,选择 Icon 属性项;
- 点击 pencil 按钮,将弹出 display icon 对话框;
- 点击 display icon 对话框中可以设定本次设定的图标的标题、图标、与原始图标的倍数关系等,在本例中不改变标题和 Size 值的 icon 列表的滑动条,或者双击列表中的图标,选择1号红色图标,点击 draw 按钮,在系统布局窗口的适当位置按下鼠标即可。

其它三种零部件可视化设计步骤与 top 雷同,只是在选择图标的时候,分别选择 2、3、4 号图标即可。

上述步骤设计了 part 元素的静态可视化效果,但是在仿真运行的动态过程中, 要想使得各个 part 元素按照静态的图标在系统布局区内移动,还需要进行动态图标 设置,动态图标设置有下面两种方法:

- 设置 part 元素的 display style 属性项,步骤为:
 - ◆ 点选对象、右击出现弹出式菜单;
 - ◆ 选择 display 菜单项,激活显示设计对话框;
 - ◆ 选择 style 属性项, 弹出 display style 对话框;
 - ◆ 将默认的 style = description, 改为 style = icon;
 - ◆ 选定与静态相同的图标;
 - ◆ 点击 draw 按钮即可。
- 指定动态显示图标的图库位置号,步骤为:
 - ◆ 双击对应元素,弹出 detail 对话框;
 - ◆ 选择 general 页框;
 - ◆ 在其 actions on create 中设置活动代码:icon = 静态图标号,比如,icon = 1, 则改元素在仿真运行中,将以图库中1号位置的图标出现。
 - ◆ 点击"确定"按钮即可。

在本示例模型中,选择第二种方法,分别赋予四种 part 的 actions on create 代码



如下:

- top: icon=1 ;
- bottome: icon=2 ;
- screw: icon=3;
- assembled: icon=4 ;
- 7.2.3.2 可视化 buffer

本例中设计了一个数量为 3 的缓冲区 store, 三个库位分别用于存放零部件 top、 bottom、screw, 并采用一个矩形线框使它更像一个仓库的整体。对它的可视化步骤 如下:

- 右击 store;
- 选择 display 菜单项;
- 选择 draw 模式;
- 选择 name 属性项;
- 点击 draw 按钮;
- 在系统布局区绘制元素名称 store;
- 选择 Rectangle 属性项绘制空心矩形线;
- 选择 Text 属性项绘制提示字符 "Top Bottom Screw";
- 选择 Part Queue 属性项绘制零部件在仓库中的队列位置。

7.2.3.3 可视化 machi ne

本例中设计了一个带有专用缓冲区的组装型机器 assembly_Machine,原材料(top、bottom、screw)队列放于机器图标的左边,成品放于其专用缓冲区内,在它的右边。对其可视化步骤如下:

- 选中并右击 assembly_Machine;
- 选择菜单项 display;
- 选择 draw 模式;
- 选择 name 属性项绘制机器的名称;
- 选择 icon 属性项绘制机器的图标 icon = 5;
- 选择 Part Queue 属性项确定零部件排队的位置;

注:此时并不能够确定产品放入专用缓冲区的队列位置和排队方式,在进行了机器的输出规则定义为 buffer (10) 之后,才能进行下面的操作。

● 选择 output buffer 属性项确定产品在缓冲区的队列位置;

7.2.4 详细定义元素

7.2.4.1 对 part 元素 top 的详细定义

- 属性定义:
 - ◆ top.Arrival Type = Active ! top 的到达类型为主动型;



- top.Maximum=10 !top 零件源有限,最多到达 10个;
- ◆ top.inter Arrival= UNIFORM (3.0,4.0,1)
 ! 相邻 top 到达时间间隔服从均 匀分布;
- ◆ top.Lot=1 ! top 的批量为 1;
- 规则定义:

۲

top's output Rules :

- ◆ PUSH to Store(1) ! top 到达将排队于 store 的第一个位置;
- 7.2.4.2 对 part 元素 bottom 的详细定义
- 属性定义:
 - ◆ bottom.Arrival Type = Active ! bottom 的到达类型为主动型;
 - ◆ bottom.Maximum=10 ! bottom 零件源有限,最多到达10个;
 - ◆ bottom.inter Arrival= UNIFORM (4.0,6.0,2) ! 相邻 bottom 到达时间间隔 服从均匀分布;
 - ◆ bottom.Lot=1 !bottom的批量为1;
- 规则定义:

bottom's output Rules :

- ◆ PUSH to Store(2) ! bottom 到达将排队于 store 的第二个位置;
- 7.2.4.3 对 part 元素 screw 的详细定义
- 属性定义:
 - screw.Arrival Type = Active
 - ◆ screw.Maximum=20 ! screw 零件源有限,最多到达 20 个;
 - ◆ screw.First Arrival=20.0 ! screw 第一批在仿真钟 20 处到达;
 - screw.inter Arrival= UNIFORM (20.0,30.0,3)
 - ◆ screw.Lot=5 ! screw 的批量为 5;
- 规则定义:

screw's output Rules :

- ◆ PUSH to Store(3) ! screw 到达将排队于 store 的第三个位置;
- 7.2.4.4 对 machi ne 元素 assembl y_Machi ne 的详细定义
- 属性定义:
 - ◆ assembly_Machine.Type=Assembly ! 机器 assembly_Machine 为组装 型机器;
 - ◆ assembly_Machine.Input Quantity=4 ! assembly_Machine 的输入零部件数 量为 4 个;
 - ◆ assembly_Machine.Cycle Time=4.0 ! assembly_Machine 的组装周期时间 为 4 个时间单位:
- 规则定义:



assembly_Machine.Input Rules (From...) :

- ◆ SEQUENCE / Wait Store(1)#(1), Store(2)#(1), Store(3)#(2)
 - ! assembly_Machine 机器空闲时,将从依次从 store (1)中取1个零件、
 从 store (2) 中取1个零件、从 store (3) 中取2个零件;

assembly Machine.Input Rules (To...) :

- ♦ BUFFER (10) ! assembly_Machine 机器组装完毕之后,成品输 出到自身的缓冲区,缓冲区的大小为10;
- 活动定义:

assembly_Machine's actions on finish ! 设定机器完成一个周期工序时发生的 活动;

 ◆ CHANGE ALL to Assembled ! 组装完毕之后,将所有的输入零部 件转换为 Assembled 零部件;

7.2.5 仿真运行

通过以上的设计,该模型设置完毕,可以仿真运行了。

按 Run 工具栏上的连续可视运行按钮,仿真钟运行到 110.63,系统停止运行, 并弹出对话框"Unable to run model further (No events scheduled)"提示没有计划的 运行事件,不能够继续运行模型。这是因为计划中的装配 10 件产品的任务已经完毕 了。

7.2.6 小结

通过本节内容的学习,应当能够:

- 了解主动、被动 Part 元素的区别;
- 分类分区摆放零件的缓冲区的建模;
- 带有专用缓冲区的机器的建模;
- 组装型机器的建模;
- 自定义元素的可视化效果;
- 使用 style 与 icon 两种方式设定零部件的动态可视化效果;

7.3 属性模型

7.3.1 模型概述

有两个种类的部件:标准型(Standard)和豪华型(Deluxe),主动到达系统,存放 于同一缓冲区 Store 中, store 设置了动态显示当前存储的两种部件的数量的属性,排 队规则为先入先出 FIFO;有名称叫做 Machine1 的机器从缓冲区 store 中获取部件进



行处理,加工时间根据加工的部件类型不同而不同,标准型部件的加工时间为 9 分钟,豪华型部件的加工时间需要 11 分钟。模型的可视化效果如图 7.13,参看随书光 盘的 models 目录下的 Attribute.mod。

| Store | Machine1 | Parts Key Standard Deluxe HACHINE STATES HACHINE STATES Off Shift Waiting Parts Busy Blocked Setup Broken Down Wait Cycle Labor Wait Repair Labor |
|-------|----------|---|
|-------|----------|---|

7.13 属性模型的可视化界面

7.3.2 元素定义

- 通过菜单项 Window/Control...修改布局窗口的背景色为蓝色,名称为 Showing a Use of Attributes。
- 通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过元素定义窗口
 定义下列元素:
 - part : Standard, Deluxe
 - buffer : Store
 - machine : machine1
 - ◆ attribute:process_time !用于存储两种部件的加工时间的属性;
- 通过系统菜单 View/Keys/Machine..., 弹出 Display Key 对话框,进行机器的标志键设置,按 backgroud 按钮,设置标志键的背景色为蓝色,按 OK 按钮,完成 MACHINE STATES 的设定。

7.3.3 可视化元素

选中 Picture Gallery...中的一个位置,点击 Picture Gallery...菜单项 File/Import, 打开随书光盘 pictures 目录,找到 standard.bmp、deluxe.bmp、attribute_machine.bmp、 attribute_store.bmp,将它们逐个导入到当前图库当中,以便使用。假设将这四个图 片分别导入在当前图库的 37,38,39,40 四个位置上。

通过 7.2 节的设置元素可视化属性的步骤,进行如下设定:

◆ 设定 standard 和 deluxe 两种 part 的可视化属性 name、icon;



- · 设定 Machine1 的可视化属性 name、icon、part queue;
- ◆ 设定 store 的可视化属性 name、icon、rectangle、text、 expression、text、 expression,两个表达式 expression 通过返回函数的值,分别显示当前缓冲 区中 standard 和 deluxe 部件的数量,两个文本 text 分别显示一段字符,起 到提示作用;对第一个 text 属性项进行设定:在 display text 对话框中,对 label下的文本框中输入 standard text,在 text下的文本框中输入 Number of Standard Items;对第二个 text 属性项进行设定:在 display text 对话框中, 对 label下的文本框中输入 deluxe text,在 text下的文本框中输入 Number of Deluxe Items;对第一个 expression 属性项进行设定:在 display expression 对话框中,对 label下的文本框中输入 standard, expression下的文本框中 输入 NPARTS2 (Store,Standard,0), type选择实型 integer, update选择 1.00, 每个仿真时刻点刷新表达式一次;对第二个 expression 属性项进行设定: 在 display expression 对话框中,对 label下的文本框中输入 deluxe expression 下的文本框中输入 NPARTS2 (Store,Deluxe,0), type选择实型 integer, update 选择 1.00,每个仿真时刻点刷新表达式一次。

注:NPARTS2 (element_name, part_name, mode) 函数返回指定元素中所包含的 指定部件的数量。

7.3.4 详细定义

7.3.4.1 对 part 元素 standard 的详细定义:

- 属性定义:
 - standard.Arrival Type = Active
 - ◆ standard.inter Arrival= NEGEXP (10.0,1) ! 相邻 standard 到达时间间隔 服从负指数分布;
 - ♦ standard.Lot=1
- 规则定义:

standard's output Rules :

- PUSH to Store
- 活动定义:

standard's actions on create :

- ◆ ICON = 37 !指定动态图标;
- ◆ Process_Time = 9.0 ! 指定该部件的属性 process_time 的值;
- 7.3.4.2 对 part 元素 del uxe 的详细定义:
- 属性定义:
 - deluxe.Arrival Type = Active
 - deluxe.inter Arrival= NEGEXP (10.0,2)



- ♦ standard.Lot=1
- 规则定义:

deluxe's output Rules :

- PUSH to Store
- 活动定义:

standard's actions on create :

- ♦ ICON = 38
 !指定动态图标;
- ◆ Process_Time = 11.0 !指定该部件的属性 process_time 的值;
- 7.3.4.3 对 machi ne 元素 machi ne1 的详细定义:
- 属性定义:
 - ◆ machine1.Type=single ! 机器 machine1 为单处理机器;
 - ◆ machine1.Cycle Time=process_time ! machine1 的处理周期为当前处理 对象的 process_time 属性值;
- 规则定义:

machine1.Input Rules (From...) :

 ◆ PULL from Store ! machine1 机器空闲时,将从 store 队列最前面的1个 零件;

machine1.Input Rules (To...) :

◆ PUSH to SHIP ! 完成品输出模型系统之外;

7.3.5 运行模型

如果运行模型至仿真时间 1000,从报告中可以看出 machine1 的运行效率为 100%。但是 machine1 并不是非常高效的,因为许多的部件在缓冲区中排队等待了很 长的一段时间(在仿真钟为 1000 时, machine1 处理了 100 个部件,但是还有 84 个 仍然在缓冲区里,所有部件在缓冲区中的平均等待时间为 265 分钟)。模型运行的时 间越长,情况越差:在仿真钟为 10000 时,部件在缓冲区的平均等待时间达到 2378 分钟。

7.3.6 小结

通过本节内容的学习,应当能够:

- 使用 attribute 元素;
- 可视化 expression 属性项;
- 了解加工时间可以是常数、属性名称以及表达式。


7.4 输送链模型

7.4.1 模型概述

这个示例模型解释了怎样模拟一个固定式 Fixed 输送链。

部件 part1 每 8 分钟达到一个,输出到输送链 conveyor1,输送链将部件输出到 机器 Machine1,机器对部件进行一定的加工后,送出模型系统。机械偶尔会出现故 障,仿真结果会显示故障对机器的影响。模型的可视化效果如图 7.14 所示,参看随 书光盘的 models 目录下的 Conveyor.mod。



图 7.14 输送链模型的可视化界面

7.4.2 元素定义

- 通过菜单项 Window/Control...修改布局窗口的背景色为蓝色,名称改为 Conveyor。
- 通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过元素定义窗口
 定义下列元素:
 - part : part1
 - conveyor : conveyor1
 - machine : machine1
- 通过系统菜单 View/Keys/Machine..., 弹出 Display Key 对话框,进行机器的标志键设置,按 backgroud 按钮,设置标志键的背景色为蓝色,按 OK 按钮,完成 MACHINE STATES 的设定。

7.4.3 可视化元素

选中 Picture Gallery...中的一个位置,点击 Picture Gallery...菜单项 File/Import,



打开随书光盘 pictures 目录,找到 conveyor_part.bmp、 conveyor_machine.bmp,将它 们逐个导入到当前图库当中,以便使用。假设将这四个图片分别导入在当前图库的 191,192 号位置上。

Part1 不需要进行可视化设置;

设置 Conveyor1 的 name 和 parts on;说明一下 parts on 属性项,该属性项用来指定 part 在 conveyor 上是以队列方式还是计数方式进行显示,如果是队列方式,需要 设定队列的方向等。本例中通过点击 display conveyor 对话框,选择 parts on 属性项,点击 pencil 按钮,弹出 display parts on 对话框,设定 queue Type 为 queue 队列式,队列方向为倾斜 diagonal,点击 draw,在系统布局区绘制即可。

设置 machine1 的 name 和 icon 属性项, icon = 192。

7.4.4 Conveyor 详细对话框介绍

| Detail Conveyor - Conveyor1 | × |
|---|---|
| General Breakdowns Shift Actions R | eporting Notes |
| Name: Quantity | Priority Type: |
| Conveyor1 1 | Lowest Fixed |
| <u>L</u> ength in Parts: | Ma <u>x</u> imum Capacity: |
| 10 | Same as length |
| Input Movement Index Time: 2.0 Restart Undefined Erom ctions On Join Wait X | Output :tions On Front. X To Push |
| | |

7.15 Conveyor 详细对话框的 general 界面

通过双击 Element Selector 窗口中的 conveyor1 图标,将弹出 detail conveyor 窗口, 如图 7.15,现在对它的 general 页框中的属性项目进行说明。

- name:显示和修改输送链的名称;
- quantity:同名输送链的条数,可以为1-999条;
- Type:设定输送链是固定式还是队列式;
- Length in Parts:输送链的长度,以能够连续排列多少个 parts 来计量;
- Maximum Capacity:输送链上可以排列的 parts 的数量,默认为" Same as length",



当输送链有承重约束时,特别需要设定好该数值;

- From:设定输入规则;
- To:设定输出规则;
- Index Time:设置输送链的运动速度(1part长度/Index Time),多长时间输送链 向前移动一个部件的长度,Index Time 越小,输送链速度越快;
- Restart:当输送链被阻塞后重启,发动机需要多长时间达到正常速度,可以输入常数值,也可以输入表达式。

7.4.5 详细定义

- 7.4.5.1 对 part 元素 part1 的详细定义:
- 属性定义:
 - part1.Arrival Type = Active
 - ◆ part1.inter Arrival= 8.0 相邻 part1 等间隔达到;
 - ♦ part1.Lot=1
- 规则定义:

part1's output Rules :

- ◆ PUSH to Conveyor1 at Rear 推至 Conveyor1 的尾部;
- 活动定义:

part1's actions on create :

- ◆ **ICON** = 191 指定动态图标;
- 7.4.5.2 对 conveyor 元素 conveyor1 的详细定义:
- 属性定义:
 - ◆ conveyor1.Type = Fixed 输送链为固定式;
 - ◆ conveyor1.Length in Parts=10 输送链长度为 10 个 part1 的长度;
 - ◆ conveyor1.Index Time=2.0 输送链每 2 个时间单位移动一个 part1 的 长度;
- 规则定义:

conveyor1's output Rules :

- ◆ PUSH to Machine1 将部件送出到 Machine1;
- 7.4.5.3 对 machi ne 元素 machi ne1 的详细定义:
- 属性定义:
 - ◆ machine1.Type = Single 机器类型为单处理机;
 - ◆ machine1.Cycle Time = 1.0 机器加工零部件的周期时间为1个时间单位;
- 规则定义:
 machine1's output Rules:
 - ◆ PUSH to SHIP 加工完毕将部件送出模型系统;



- 由于机器 machine1 偶尔会发生故障,需要设定它的 breakdown 属性页,将 detail 对话框由 general 页框转至 breakdown 页框,进行如下的设置。
 - ◆ machine1.breakdown = Machine Breakdown1 设置一个故障的名称;
 - ◆ machine1. Machine Breakdown1. Check only at start of cycle = .T. 设置在每 一加工循环开始时检测故障;
 - ♦ machine1. Machine Breakdown1. Breakdown mode = Busy Time 设置
 故障模式为 Busy Time ;
 - ◆ machine1. Machine Breakdown1.Time Between Failure = 20.0 相邻两次 故障间隔中,机器加工了 20 个时间单位;
 - ◆ machine1. Machine Breakdown1.Repair Time = 50
 机器故障维修需要 50 个时间单位;

7.4.6 运行模型及分析

运行模型(假设运行到仿真钟 1000),因为部件每 8 个时间到达一次,在输送链 上移动一个位置要 2 个时间,你将看到输送链上每四个位置有一个部件。

在 100 时间的时候,机械出现故障,部件不能传送。这时,位置 4 和位置 8 分 别有一个部件,在以后的 6 分钟内,部件不能移动到位置 1 和 5。但是在 106 个时间 后,在前部的部件将无处可去,输送链就被堵塞了。

但是,如果有足够的空间,部件依然能够进入输送链。在 104 时间第三个部件 进入,在 112 时间第四个部件进入了最后一个位置,从时间 100 到时间 150,机械处 于维修状态,完全被阻塞,任何到达的部件都被拒绝,直到机械被修好。

7.4.7 输送链类型不同的差异

上面设定的输送链为固定式的,一旦输送链的前边部件不能够移动,整个输送 链就停止运行了。

而队列式 queue 输送链允许部件在输送链上累积,即使输送链被阻塞,它上面的 parts 还是会滑动到连续的队列,直到输送链上排满了 parts。

双击模型中的传送机,显示详细对话框,将 conveyor type 变为 queuing,把模型 重置为 0,重新运行,将会看到队列式输送链的不同。

在时间 106,队列式模型和固定式模型完全一样(机械出现故障,位置 1,5 和 9 上各有一个部件)。

在时间 108,在位置1上的部件已经无处可去,但是位置5和9上的部件可以向 前移动一个位置。部件每两分钟向前移动一次,堆在位置1的后面。在时间150,位 置1到8上都有部件,共有8各 Parts 在输送链上,此时机器被修好,部件从传送机 上移动下来。

如果把这两种类型的输送链模型都运行 1000 个单位时间,在表 7.3 的统计报告



中,可以比较出它们的不同:

| Report | Fixed conveyor | Queuing conveyor | 说明 |
|-------------------|----------------|------------------|----|
| Conveyor1 | | | |
| % Move | 78.5 | 65.2 | |
| % Blocked | 21.5 | 0 | |
| % Queue | 0 | 34.8 | |
| Now On | 3 | 3 | |
| Total On | 106 | 126 | |
| Avg Size | 2.87 | 3.59 | |
| Avg Time | 27.04 | 28.48 | |
| Machine1 | | | |
| % Idle | 64.7 | 57.7 | |
| % Busy | 10.3 | 12.3 | |
| % Broken Down | 25 | 30 | |
| No. Of Operations | 103 | 123 | |
| Part1 | | | |
| No. Entered | 106 | 126 | |
| No. Shipped | 103 | 123 | |
| No. Rejected | 20 | 0 | |
| W.I.P. | 3 | 3 | |
| Avg W.I.P. | 3.22 | 4.01 | |
| Avg Time | 30.37 | 31.83 | |

表 7.3 使用不同类型输送链的统计报表

数据显示:固定式输送链比队列式输送链的移动时间要长,阻塞时间长,不过 队列式输送链多送了 20 个部件,这是因为它在机器不能处理的时候依然能容纳部 件,而固定传送机则拒绝了 20 个部件。

7.4.8 小结

通过本节内容的学习,应当能够:

- 使用 conveyor 元素的建模;
- 两种 conveyor 元素的异同;
- 机器 busy time 抛锚模式的设定。

7.5 函数模型

7.5.1 模型概述

部件 part1 到达模型,在 Store 缓冲区中储存。机器 Machine1 从缓冲区中取出部



件进行加工,然后送出模型。

部件有三个特性,每一个特性由一个属性代表:Length, Height 和 Weight。机器 对每个部件的处理时间是通过一个函数计算表达式(Length* Height)+Weight 得到 的,即每个部件的处理时间与它的属性值有关系。模型的可视化效果如图 7.16 所示, 参看随书光盘的 models 目录下的 function.mod。



7.16 函数模型的可视化界面

7.5.2 定义元素

- 通过菜单项 Window/Control...修改布局窗口的背景色为蓝色,名称改为 Showing a Use of a Function。
- 通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过元素定义窗口
 定义下列元素:
 - part : part1
 - buffer : store
 - machine : machine1
 - attribute : Length , Height , Weight
 - Function : Process_time

7.5.3 可视化元素

可视化 machine 时, icon 选择随书光盘中 pictures 目录的 function_machine.bmp; 可视化 buffer 时,其 expression 属性项的 expression 属性设为 NPARTS (Store);



7.5.4 详细定义

- 7.5.4.1 对 part 元素 part1 的详细定义:
- 属性定义:
 - part1.Arrival Type = Active
 - ◆ part1.inter Arrival=NEGEXP (8.0,1)
 - ♦ part1.Lot=1
- 规则定义:

part1's output Rules :

- ♦ **PUSH** to Store
- 活动定义:

part1's actions on create :

- ICON = 37
- ◆ Length = UNIFORM (1.6,2.3,2) 对属性 Length 随机赋值;
- ◆ Weight = NORMAL (2.3,0.1,3) 对属性 Weight 随机赋值;
- ◆ Height = TRIANGLE (1.1,4.5,5.0,4) 对属性 Height 随机赋值;
- 7.5.4.2 对 machi ne 元素 machi ne1 的详细定义:
- 属性定义:
 - machine1.Type = Single
 - ◆ machine1.Cycle Time = Process Time () 机器加工零部件的周期时间由函 数 Process Time ()决定;
- 规则定义:

machine1's output Rules :

- **PUSH** to SHIP 加工完毕将部件送出模型系统;
- 7.5.4.3 对 Function 元素 Process Time 的详细定义:
- 通过双击 Element Selector 窗口的 Process Time 图标,对函数进行详细定义。
 - Process Time.Type = Real

Process Time.Function body 函数体语句如下:

• **RETURN** Length * Weight + Height Weight 和 Length 相乘然后与 Height 相加得到的值为 Process Time 函数的结果。也是 Machinel 机器的 加工周期时间。每当机器要处理一个 part 时,要先决定加工时间,这时将 当前 part 的属性值带入函数 Process Time 中计算。

7.5.5 运行模型

仿真一段时间之后,可以看出 Machinel 大部分时间都是运行的,只有缓冲区中



没有部件的时候它才空闲下来。模型运行的时间越长,缓冲区中的部件越多,机器 空闲比率就越小。

7.5.6 小结

通过本节内容的学习,应当能够:

- 掌握系统标准随机分布函数 UNIFORM、NORMAL、TRIANGLE 的使用,具体 用法参看第二章;
- 掌握函数的定义方法,关键字 RETURN 的使用;

7.6 随机分布模型

7.6.1 模型概述

Part1 部件以间隔时间为 NEGEXP (8.0,1)的随机分布到达模型,进入 Store 缓冲 区排队,等待 Machine1 机器从 store 中取出来进行加工 加工时间由分布 Process_Time 来确定。Machine1 会出现故障,故障发生间隔时间和修理所需的时间分别用 WITNESS 的两个标准分布来定义的。

本节通过修改 7.5 节的模型,来得到所需要的仿真效果。模型的可视化效果同 7.5 节的图形完全一样,参看随书光盘的 models 目录下的 distribution.mod。

7.6.2 修改并添加元素 process_time 分布的步骤

- 打开随书光盘的 models 目录下的 function.mod;
- 使用系统菜单 File/Save As..., 另存为 distribution.mod;
- 删除模型元素 process_time 函数、Weight、Length、Height 属性;在删除这些元素之前,删除它们的被使用设置,本例中先删除机器 machine1 的细节对话框的 Cycle Time 文本框中的 Process_Time(),再删除 part1 的 actions on create 中的下列三条语句:
 - Length = UNIFORM (1.6, 2.3, 2)
 - Weight = NORMAL (2.3, 0.1, 3)
 - Height = **TRIANGLE** (1.1, 4.5, 5.0, 4)

这样,需要删除的几个元素就没有被其它元素使用的设定了,此时,在元素选择列表窗口选中这四个元素前面的复选框,点击右键,选择弹出式菜单中的 Delete Elments 菜单项,删除即可。

● 添加 distribution 元素:

- name = process_time ;
- type = real ;



♦ distribution = continue₀

7.6.3 详细定义

- 7.6.3.1 对 part 元素 part1 的详细定义:
- 属性定义:
 - part1.Arrival Type = Active
 - part1.inter Arrival=1.0
- 规则定义:

part1's output Rules :

- PUSH to Store
- 活动定义:

part1's actions on create :

• ICON = 37

7.6.3.2 分布 process_time 的详细定义

双击分布元素 process_time 图标,进行分布的详细定义,双击后出现设置对话框,填写 Cells 右下角的 Value 值和 Weight 值之后,按 Add 按钮添加,设置完毕后的情况如图 7.17 所示。

| Detail Distribution | - process_tir | ne | × |
|---------------------|---------------|-----------------|-----------------|
| General Action | ns Notes | | |
| <u>N</u> ame: | | Type: | |
| process_time | | Continuous Real | |
| | | : | Emport |
| Cells: | laight: | | |
| | 0 | | Add |
| 1 | 20 | | <u>R</u> emove |
| 2 | 20 | | <u>U</u> pdate |
| 4 | 30 45 | Valu | 1e ⁻ |
| 5 | 25 | 3 | |
| 6 | 15 | <u> </u> | ght: |
| | | 15 | |
| | | | |
| | 确定 | | 帮助 |

图 7.17 分布 Process_Time 详细设计图

从上图可以看出, Process_Time 是一个连续型实数分布 (Continuous Real), 取



值包括六个区间:0-1,1-2,2-3,3-4,4-5,5-6。调用 Process_Time 时, WITNESS 是从一个数据区间中选取值 values,选取的值不包括上限。

Weight 是 WITNESS 选取值的相对频率权重, process time 分布的含义如下:

以 20/155 的概率, 返回在 0.0 到 1.0 之间的实数 以 20/155 的概率, 返回在 1.0 到 2.0 之间的实数 以 30/155 的概率, 返回在 2.0 到 3.0 之间的实数 以 45/155 的概率, 返回在 3.0 到 4.0 之间的实数 以 25/155 的概率, 返回在 4.0 到 5.0 之间的实数 以 15/155 的概率, 返回在 5.0 到 6.0 之间的实数

注:

WITNESS 使用特定的伪随机流来确定使用哪一个区间,权重越大的区间,被选择到的概率越大;

一旦 WITNESS 选用了某个区间,它将按照均匀随即分布地方式,随机地从区间 中选取一个实数值返回。例如,如果它选取了第4区间,它可能选取大于等于 3.0 小 于 4.0 (例如 3.4, 3.978 或 3.0025)的任何实数。每个值的可能性是相等的。

7.6.3.3 对 machi ne 元素 machi ne1 的详细定义:

- 属性定义:
 - ◆ machine1.Type = Single ! 机器类型为单处理机;
 - ◆ machine1.Cycle Time = process_time(2) ! 机器加工零部件的周期时间
 由用户定义分布 process time(2)所决定;
- 规则定义:

machine1's output Rules :

- PUSH to SHIP
- 由于机器 machinel 偶尔会发生故障,需要设定它的 breakdown 属性页。
 - ◆ 将 detail 对话框由 general 页框转至 breakdown 页框;
 - ◆ machine1.breakdown = Machine Breakdown1 设置一个故障的名称;
 - ◆ machine1. Machine Breakdown1. Check only at start of cycle = .T. 设置在每 一加工循环开始时检测故障;
 - ♦ machine1. Machine Breakdown1. Breakdown mode = Busy Time 设置 故障模式为 Busy Time;
 - ◆ machine1. Machine Breakdown1.Time Between Failure = NEGEXP (100.0,3)
 相邻两次故障间隔中,机器加工了 NEGEXP (100.0,3)个时间单位;
 - ◆ machine1. Machine Breakdown1.Repair Time = ERLANG (5.0,2,4)
 税 器故障维修时间服从 ERLANG (5.0,2,4)分布;

7.6.4 运行模型

运行模型至 1000 仿真时间点,检查部件,缓冲区和机械的统计报告,发现机器



的 busy 状态不到 30%,可以存放 1000 个零部件的缓冲区的最大队列长度仅仅为 2, 所以原料采购为"瓶颈",制约了系统的产出率。改变零部件的到达速度,即时间间 隔由原来的 NEGEXP (8.0,1)改为常数 1.0,再运行模型 1000 个时间单位,得到两种方 案的统计结果比较如表 7.4:

| | Part1 | Part1 |
|-------------------|---------------------|---------|
| Name | inter=NEGEXP(8.0,1) | inter=1 |
| Machine1 | | |
| % Idle | 66.78 | 0 |
| % Busy | 29.07 | 90.84 |
| % Broken Down | 4.16 | 9.16 |
| No. Of Operations | 94 | 303 |
| Store | | |
| Total In | 95 | 1001 |
| Total Out | 95 | 304 |
| Now In | 0 | 697 |
| Max | 2 | 697 |
| Avg Size | 0.1 | 351.27 |
| Avg Time | 1.01 | 350.91 |
| Part1 | | |
| No. Entered | 95 | 1001 |
| No. Shipped | 94 | 303 |
| W.I.P. | 1 | 698 |
| Avg W.I.P. | 0.43 | 352.27 |
| Avg Time | 4.51 | 351.91 |

表 7.4 在不同部件到达时间间隔情况下的统计报表

通过比较可以看出,在改变了部件到达的时间间隔后,传送的部件更多,机械 的利用率为 90%。机械的故障次数也增多,因为故障的次数依赖于运行的时间,而 运行的时间变长了。

在此也显示了大缓冲区的好处,有时多于 2/3 的缓冲被部件占有。但是部件在 缓冲区的时间太长了(351.91 仿真时间,而以前是 4.51 个仿真时间)。如果模拟的产 品十分易损(比如冰激凌),那么就需要进一步的探索,找更满意的答案。

7.7 路径模型

7.7.1 模型概述

这个模型展示了如何使用路径 path 元素。



图 7.18 路径模型可视化效果图

路径是一个单元,部件和劳动力(或其他资源)能沿着它从一个单元到另一个 单元。在模型中它用来表示真实世界中路径的长度和物理性质。当两个操作的时间 间隔相当重要时,路径在增加模型的准确度上是十分有用的。

在这个模型中,椅子由靠背 back、坐垫 seat、椅腿 leg 组装完成之后,沿着一个路径被送到喷漆部门,喷成红色、绿色或黄色,然后送去检查,有部分由于喷漆不 合格,被送回重新喷漆,其它的其他的被送去包装,相同颜色的 4 把椅子打成一包, 然后被运走。模型的可视化效果如图 7.18 所示,也可以参看随书光盘的 models 目录 下的 paths.mod。

7.7.2 定义元素

- 通过菜单项 Window/Control...修改布局窗口的名称为 paths。
- 通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过元素定义窗口
 定义下列元素:
 - part : Back, Seat, Legs ;
 - buffer : B1、B2、B3、Paint_Q、Inspection_Q、Packing_Q;
 - path : path1、 path2、 path3、 path4、 path5 ;
 - machine : Assembly, Painting, Inspection, Packing ;
 - ♦ labor : Inspector ;
 - variable : x ;
 - ♦ attribute : c ;



7.7.3 可视化元素

7.7.3.1 绘制成品椅子图标

- 选择系统菜单 View/Picture Gallery...;
- 选中一个没有图标的位置,假设为153;
- 右击 153,选择弹出菜单中的"Editor"菜单项,将弹出 Icon Editor 窗口,如图 7.19。

选中对话框中三支笔的第一只,当选中该笔时,它将灰化;点击颜色按钮的第 一个按钮,设置成白色;鼠标左键点击右侧绘图区,绘制成模型布局图中椅子的形 状;选定单色 Monochrome,这样该图标才能够改变颜色;单击 OK 按钮确认。



图 7.19 图标编辑窗口

7.7.3.2 设置 part 和 buffer 的可视化效果

在元素选择窗口分别选择 Back、Seat、Legs 元素,设置它们的 name、icon 和 style 属性项;

分别选择 B1、B2、B3、Paint_Q、Inspection_Q、Packing_Q 元素,设置它们的 name、icon 和 part queue 属性项,在 display part queue 对话框(如图 7.20)中设置 part queue 属性项的 queue type 选择为 Count, display Size 设置为 3 位,则这些缓冲区将 采用计数的方式动态显示缓冲区内的 part 数量。



| 🌌 Display Part Q | ueue - B2 | | × |
|-------------------------------|-----------------------|----------------------|--------------------|
| La <u>b</u> el: Part Queue | _ | | Draw |
| Queue Type | | | <u>U</u> pdate |
| Count | | | Cancel |
| C Queue | | | Rese <u>t</u> |
| A Abc | | | Help |
| Foreground | | <u>Г</u> <u>О</u> ре | aqueText |
| S <u>e</u> paration: 0 + | Display <u>S</u> ize: | Displa 32000 | y <u>Maximum</u> : |

图 7.20 display part queue 对话框

7.7.3.3 设置 machi ne 的可视化效果

分别选择 Assembly、Painting、Packing,设置它们的 name、icon(机器图标), icon(可随状态改变颜色的图标),放置于适当的位置即可。

选择机器 Inspection,设置它的 name、icon(机器图标) icon(可随状态改变 颜色的图标), part queue 和 labor queue 属性项。

7.7.3.4 设置 path 的可视化效果

分别选择 path1、path2、path3、path4、path5,设置它们的 name 属性项和 path 属性项, display path 对话框如图 7.21:

| 🥶 Display Path - Path1 | × |
|--------------------------|---------------|
| Label: | Draw |
| Path | |
| | <u>Update</u> |
| V Show Line | Reverse |
| | |
| Display Size: | Cancel |
| 2 Width: Inner Width: | Help |
| 8 + 5 + | |
| V Hollow V Center | |
| Layer: | |
| Simulation Layer 🗨 | |



在选定 Hollow (中间填充)和 Center (居中)复选框后,上图 Label 文本框下



面的两个按钮作用分别是:前一个按钮设置路径的边框颜色,后一个按钮设置路径的填充颜色;width 设置路径的宽度,Inner Width 设置内部路径的宽度。设置如上图后,点击 Draw 按钮,在系统布局区绘制路径。

首次绘制的路径在可视化长度上以及方向上可能不满足要求,通过按住 CTRL 键的同时,在鼠标光标变成一字型后,拖拉鼠标即可改变路径的形状。

7.7.3.5 设置 Labor 的可视化效果

选择 labor 元素 Inspector,进行 name 和 Idle 的设置如图 7.18。

7.7.4 详细定义

7.7.4.1 对 part 元素的详细定义:

- 属性定义:
 - seat.Arrival Type = Active
 - ♦ seat.inter Arrival=2.0
 - back.Arrival Type = Active
 - back.inter Arrival=2.0
 - legs.Arrival Type = Active
 - legs.inter Arrival=2.0
- 规则定义:

seat's output Rules :

• **PUSH** to B1

back's output Rules :

• **PUSH** to B2

legs' output Rules :

- **PUSH** to B3
- 7.7.4.2 对 machi ne 元素 Assembly 的详细定义:
- 属性定义:
 - Assembly.Type=Assembly
 - Assembly.Cycle Time=6.0
 - Assembly. Input Quantity=3;
 1 机器 Assembly 的输入零部件数量为 3个;
- 规则定义:

Assembly.Input Rules (From) :

 ◆ MATCH/ANY B1 #(1)B2 #(1)B3 #(1) ! 匹配缓冲区 B1、B2、B3 中 的任意类型的 part 各一个;

Assembly.Output Rules (To) :

◆ PUSH to Paint_Q Using Path ! 通过路径将成品送至缓冲区 Paint_Q;



活动定义:

Assembly.actions on finish:

- ♦ ICON = 143
 ! 通过变换图标,表示 seat、back、legs
 组装成了一把白色椅子;
- 7.7.4.3 对 path 元素 path1 的详细定义:

● 属性定义:

双击 path1 图标,显示 detail path 对话框,在它的 general 中设置属性如下:

- ◆ path1. Path traverse time=15.0 ! 设定路径 path1 输送一个 part 通过所需 要的时间;
- ◆ path1. Path update interval = 0.01
 ! 设定 path1 的图形刷新率,数字越小, 刷新越频繁,图形显示越连续;
- ◆ path1.Source element = Assembly ! 设定 path1 的起始元素为机器 Assembly;
- ◆ path1. Destination element = Paint_Q ! 设定 path1 的目的地元素为缓冲
 区 Paint_Q ;
- 7.7.4.4 对 machine 元素 painting 的详细定义:
- 属性定义:
 - painting.Type=Single
 - painting.Cycle Time=10.0
- 规则定义:

painting.Input Rules (From) :

```
◆ PULL from Paint_Q ! 从缓冲区 Paint_Q 中提取零件加工;
```

painting.Output Rules (To) :

- ◆ PUSH to Inspection_Q Using Path ! 通过路径将成品送至缓冲区 Inspection_Q;
- 活动定义:

painting.actions on finish:

```
    !下面这段程序实现等概率将椅子油漆成红、绿、黄三种颜色,并根据颜色对椅子
    设定属性 C = "red"、"green"、"yellow"。
    x = IUNIFORM (1,3,356)
    !随机对变量 x 赋值;
```

```
ICON = 143
```

```
IF x = 1
```

```
PEN = 1
```

C = "red"

```
ELSEIF x = 2
```

```
PEN = 2
```

C = "green"



ELSE

PEN = 3

C = "yellow"

ENDIF

程序说明:

PEN为系统属性,可以表示当前元素的颜色,针对本模型,如果PEN=1,则椅子为红色;如果PEN=2,则椅子为绿色;如果PEN=3,则椅子为黄色色;完整说明参看WINTESS帮助文件。

第三到第十二行是一个多向选择分支结构。如果变量x = 1,则给PEN赋值1,C 赋值"red",即当前椅子的颜色将呈现红色;如果变量x = 2,则给PEN赋值2,C赋 值"green",即当前椅子的颜色将呈现绿色;如果变量x = 3,则给PEN赋值3,C赋 值"yellow",即当前椅子的颜色将呈现黄色。

7.7.4.5 对 path 元素 path3 的详细定义:

属性定义:

双击 path3 图标,显示 detail path 对话框,在它的 general 中设置属性如下:

- ◆ path3. Path traverse time=10.0 ! 设定路径 path3 输送一个 part 通过所需 要的时间;
- ◆ path3. Path update interval = 0.01 ! 设定 path3 的图形刷新率,数字越小, 刷新越频繁,图形显示越连续;
- ◆ path3.Source element = Painting ! 设定 path3 的起始元素为机器 Painting;
- ◆ path3. Destination element = Inspection_Q ! 设定 path3 的目的地元素为 缓冲区 Inspection_Q;
- 7.7.4.6 对 machine 元素 inspection 的详细定义:
- 属性定义:
 - inspection.Type=Single
 - inspection.Cycle Time=3.0
 - inspection.labor = Inspector
- 规则定义:

inspection.Input Rules (From) :

◆ PULL from Inspection_Q ! 从缓冲区 Inspection_Q 中提取零件加工; inspection.Output Rules (To):

• PERCENT /189 Packing_Q Using Path 90.00,

Paint_Q With Inspector Using Path 10.00 ! 产生随机概率, 以 90%的概率通过检测,使用路径移向 Packing_Q,进行打包;以 10%的 概率检测出油漆有质量问题,需要人工搬运,通过路径送回缓冲区 Paint_Q, 排队重新油漆。PERCENT 命令详细说明参考 WITNESS 帮助文件。



- 7.7.4.7 对 path 元素 path4 的详细定义:
- 属性定义:

双击 path4 图标,显示 detail path 对话框,在它的 general 中设置属性如下:

- ◆ path4. Path traverse time=10.0 ! 设定路径 path4 输送一个 part 通过所需
 要的时间;
- ◆ path4. Path update interval = 0.01 ! 设定 path4 的图形刷新率,数字越小, 刷新越频繁,图形显示越连续;
- ◆ path4.Source element = Inspection ! 设定 path4 的起始元素为机器 Inspection;
- ◆ path4. Destination element = Paint_Q ! 设定 path4 的目的地元素为缓冲
 区 Paint_Q ;
- 7.7.4.8 对 path 元素 path2 的详细定义:
- 属性定义:
 - 双击 path2 图标,显示 detail path 对话框,在它的 general 中设置属性如下:
 - ◆ path2. Path traverse time=15.0 ! 设定路径 path2 输送一个 part 通过所需 要的时间;
 - ◆ path2. Path update interval = 0.01 ! 设定 path2 的图形刷新率,数字越小, 刷新越频繁,图形显示越连续;
 - ◆ path2.Source element = Inspection ! 设定 path2 的起始元素为机器 Inspection;
 - ◆ path2. Destination element = Packing_Q ! 设定 path2 的目的地元素为缓 冲区 Packing_Q;
- 7.7.4.9 对 machi ne 元素 packi ng 的详细定义:
- 属性定义:
 - packing.Type=Assemly
 - packing.Cycle Time=4.0
 - packing. Input Quantity=4
- 规则定义:

packing.Input Rules (From) :

 ◆ MATCH/ATTRIBUTE C Packing_Q #(4) ! 从缓冲区 Packing_Q 中提 取 4 各属性 C 相匹配的 part, MATCH 命令的用法参看 WITNESS 帮助文 件;

packing.Output Rules (To) :

- ◆ PUSH to SHIP Using Path ! 使用路径将打包好的 part 送出模型;
- 活动定义:

pacting.actions on finish:



! 指定一种代表四把同颜色椅子打包后的图标; ICON = 1IF C = "red"PEN = 1ELSEIF C = "green" PEN = 2ELSE

PEN = 3

ENDIF

第二行到第八行是一个多向分支结构,根据输入零部件的颜色属性来指定输出零 部件的颜色。

- 7.7.4.10 对 path 元素 path5 的详细定义:
- 属性定义:

双击 path5 图标,显示 detail path 对话框,在它的 general 中设置属性如下:

- path5. Path traverse time=5.0 ! 设定路径 path5 输送一个 part 通过所需要 ٠ 的时间;
- ◆ path5. Path update interval = 0.01 !设定 path5 的图形刷新率,数字越小, 刷新越频繁,图形显示越连续;
- ◆ path5.Source element = Packing ! 设定 path5 的起始元素为机器 Packing;
- path5. Destination element = Ship ! 设定 path5 的目的地元素为缓冲区 ٠ Ship;

7.7.5 运行模型

仿真运行该模型 1000 时间单位,得到如下的统计报表:

| Name | % Idle | % Busy | No. Of Operations |
|------------|--------|--------|----------------------|
| Assembly | 0 | 100 | 166 |
| Painting | 2.1 | 97.9 | 97 |
| Inspection | 68.2 | 28.8 | 96 |
| Packing | 92 | 8 | 20 |

表 7.5 机器工作状态统计表

表 7.6 路径工作状态统计表

| Name | Parts In | Parts Out | Labor In | Labor Out | % Busy | % Idle |
|-------|----------|-----------|----------|-----------|-----------|--------|
| Path1 | 166 | 164 | 0 | 0 | 99.4 | 0.6 |
| Path2 | 86 | 85 | 0 | 0 | 88.5 | 11.5 |
| Path3 | 97 | 96 | 0 | 0 | 96.9 | 3.1 |
| Path4 | 10 | 10 | 10 | 10 | 10 | 90 |



北京威特尼斯科技中心

| Path5 | 20 | 0 | 0 | 0 | 10 | 90 |
|-------|----|---|---|---|----|----|

表 7.7 劳动者工作状态统计表

| Name | % Busy | % Idle | Quantity | No. Of Jobs Started | No. Of Jobs Ended | Avg Job Time |
|-----------|-----------|--------|----------|------------------------|----------------------|--------------|
| Inspector | 38.8 | 61.2 | 1 | 106 | 106 | 3.66 |

通过这些报表可以看出,流水线上的机器利用率越来越低,劳动者的劳动时间 比例也比较低,从 path1、path3、path4 这个次序来看,路径上的零部件通过量也是 逐步减少,这是因为零部件的加工时间和在路径上的行进时间造成的结果。

7.8 EXCEL 数据交互模型

7.8.1 模型概述

这个模型介绍 WITNESS 系统如何同 EXCEL 数据表进行数据的交互。

有三种等概率的需求根据 EXCEL 的数据随机达到系统,并且进入队列,等待经 过三道程序的服务。有三种类型的业务组分别完成这三道程序,每类业务组中所拥 有的工作台数量以及服务时间通过读取 EXCEL 文件中的数据来确定;在仿真运行过 程中,系统实时的向 EXCEL 中写下当前各个业务组的工作状态;当需求得到满足, 离开系统时,向 EXCEL 中写下这一需求在系统中的时间。

模型的可视化效果如图 7.21 所示,也可以参看随书光盘的 models 目录下的 XLLinks.mod。



图 7.21 XLLinks 模型的可视化效果图



7.8.2 预备知识

7.8.2.1 part 元素的 Active With Profile 到达方式

有的系统中,需求的到达不符合标准的随机分布函数,而是在不同的时间段内 具有不同的分布密度。比如餐饮业,在一日三餐的时间,顾客到达的密度比较大, 而在上午九、十点钟和下午的三、四点钟,顾客到达密度相对较小。这是我们可以 通过设定需求到达呈现时效性和随机性,可以设定需求到达模式为 Active With Profile 模式的。

在选定了需求 part 元素的到达模式为 Active With Profile 后, part 的 detail 对话框 上会动态显示" arrival profile"页框,来进行 profile 的设计。

7.8.2.2 读取 EXCEL 文件中数据的方法

WITNESS 将 EXCEL 文件中数据读取出来之后,存放于变量或变量数组中,以便使用,读取的方法有两种:

- 通过仿真系统的初始化活动。使用此方法的步骤如下:
 - ◆ 选择系统菜单项 Model/Initialize Actions..., 激活程序编辑窗口;
 - ◆ 写入 XLReadArray ()函数,来读取工作表中特定区域的数据给变量;
- 通过变量的初始化活动。使用此方法的步骤如下:
 - ◆ 双击变量,打开 detail 对话框;
 - ◆ 选择 Actions 页框中的 Initialize...按钮;
 - ♦ 弹出程序编辑对话框;
 - ◆ 点击程序编辑对话框上的 Excel...按钮;
 - ◆ 选定数据表所处的工作簿;
 - ◆ 选定工作簿中工作表中的特定区域;
 - ◆ 单击确定即可;

7.8.2.3 将仿真数据写入 EXCEL 表的方法

将仿真数据写入 EXCEL 表可以通过在特定的活动中,写入 XLWriteArray()函数的方法来实现。

7.8.2.4 XLReadArray ()函数介绍

- 函数功能:
 - ◆ 该函数用来将 EXCLE 工作表中的特定区域的数据读给 WI TNESS 的变量或变 量数组。如果读取成功,该函数返回 1,否则,返回 0。
- 函数语法:

XLReadArray (WorkbookName , WorksheetName , Range , Array)

- 参数:
 - ◆ WorkbookName 字符型,指定包含需要数据的工作簿名称及其路径;



- ◆ WorksheetName 字符型,指定包含数据的工作表的名称;
- ◆ Range 字符型,指定包含数据的工作表中的区域;
- ◆ Array 名型,指定将读取的数据赋予的变量的名称;
- 7.8.2.5 XLWriteArray ()函数介绍
- 函数功能:
 - ◆ 该函数用来将 WI TNESS 的变量或变量数组的值写入 EXCEL 中去。如果写入 成功,该函数返回1,否则,返回0。
- 函数语法:
 - XLWriteArray (WorkbookName, WorksheetName, Range, Array)
- 参数:
 - ◆ WorkbookName 字符型,指定写入数据的的工作簿名称及其路径;
 - ◆ WorksheetName 字符型,指定写入数据的工作表的名称;
 - ◆ Range 字符型,指定写入数据的工作表中的区域;
 - ◆ Array 名型,指定所写入的数据变量的名称;

7.8.3 定义元素

- 通过菜单项 Window/Control...修改布局窗口的名称为 XLLinks;
- 通过在系统布局窗口单击鼠标右键,将弹出元素定义窗口,通过元素定义窗口
 定义下列元素:
 - part : part1 ;
 - buffer : store ;
 - machine : machine1, machine2, machine3 ;
 - variable :arrivals(16), cellvalue, tempararyvalue, x, processtime, rownumber, numberofmachines(3), cycletime(3);
 - attribute : partsize, arrivaltime ;

元素说明:

arrivals(16)一维变量数组用来存放从 excel 文件中读取的决定 part1 到达分布密 度的数据;

numberofmachines(3)用来存放从 excel 文件中读取的决定三类业务组的工作台的 数量;

cycletime(3) 用来存放从 excel 文件中读取的决定三类业务组的对每一个需求的 工作周期。

7.8.4 可视化元素

根据图 7.21, 分别设定 part、machine、buffer 三类元素的可视化效果。



7.8.5 详细定义

7.8.5.1 模型初始化程序详细定义

选择系统菜单项 Model/Initialize Actions...,激活程序编辑窗口,向其中写入如下程序:

XLReadArray ("\\XLLinks.xls", "InputData", "\$D\$6: \$D\$8", CycleTime)

XLReadArray ("\\XLLinks.xls", "InputData", "\$E\$6: \$E\$8", NumberOfMachines)

SET QUANTITY OF Machine1 TO NumberOfMachines (1)

SET QUANTITY OF Machine2 TO NumberOfMachines (2)

SET QUANTITY OF Machine3 TO NumberOfMachines (3)

XLReadArray ("\\XLLinks.xls", "InputData", "\$E\$15: \$E\$30", Arrivals)

RowNumber = 7

程序解释:

第一行:将与模型在同一路径下的工作簿 XLLinks.xls 中的工作表 InputData 中的 D6:D8 三个单元的数据读取出来,分别赋予 CycleTime(1)、CycleTime(2)、 CycleTime(3);

第二行:将与模型在同一路径下的工作簿 XLLinks.xls 中的工作表 InputData 中的 E6:E8 三个单元的数据读取出来,分别赋予 NumberOfMachines (1)、 NumberOfMachines (2)、 NumberOfMachines (3);

第三行到第五行:通过变量 NumberOfMachines 数组中的数据来设置工作台的数量。

第六行:将 InputData 中的 E15:E30 这十六个单元的数据读取出来,赋予数组 Arrivals。

7.8.5.2 对 part 元素 part1 的详细定义:

- general 页框属性定义:
 - ◆ part1.Arrival Type = Active With Profile !part1 到达模式为收文件驱动的方式;
 - ▶ part1.Lot=1
- Arrival Profile 页框属性定义如图 7.22 所示。



| eneral | Attributes R | oute Actions 1 | Reporting | Arrival Profile Notes |
|--------|----------------|--------------------|-----------|------------------------------|
| | Time | Volume | _ | <u>T</u> ime Display |
| 1 | 0:00 | Arrivals (1) | | 8 Hour Day 💌 |
| 2 | 0:30 | Arrivals (2) | | Wultiplier |
| 3 | 1:00 | Arrivals (3) | | marcipiter |
| 4 | 1:30 | Arrivals (4) | | Undefined |
| 5 | 2:00 | Arrivals (5) | | |
| 6 | 2:30 | Arrivals (6) | | <u>K</u> andom Number Stream |
| 7 | 3:00 | Arrivals (7) | | 1 |
| 8 | 3:30 | Arrivals (8) | | |
| 9 | 4:00 | Arrivals (9) | | <u>S</u> moothing |
| 10 | 4:30 | Arrivals (10) | | |
| 11 | 5:00 | Arrivals (11) | | <u>Cumulative lime</u> |
| 12 | 5:30 | Arrivals (12) | | Tereset |
| 13 | 6:00 | Arrivals (13) | | <u>import</u> |
| 14 | 6:30 | Arrivals (14) | - | |
| 1 | 7 | 1 () | | |

图 7.22 part1 的 Arrival Profile 页框属性定义示意图

从图 7.22 中可以看出,在每天 8 小时的工作时间内, part1 在每半个小时内的到 达密度受到 Arrivals(i)中数据的控制。

规则定义:

part1's output Rules :

◆ PUSH to Store ! 推至队列 Store;

活动定义:

part1's actions on create :

ICON = 8 !确定part1的动态图标;

 x = IUNIFORM (1, 3, 678)
 !通过整数均匀分布给变量x随机赋1、2、3三个值

 中的一个,为后面确定part1的属性partsize作准备;

IF x = 1

PartSize = "Small"

```
ELSEIF x = 2
```

PartSize = "Medium"

ELSE

PartSize = "Large"

ENDIF

ArrivalTime = TIME ! 将每个 part1 到达系统的仿真钟的时间值赋给 part1 的属 性 ArrivalTime ;

第二行到第八行,通过对变量x的值的判断,给part1的属性PartSize赋不同的



值。

part1's actions on leave :

ProcessTime = TIME - ArrivalTime ! 当前仿真钟减去需求到达时间,得出在 系统中的总时间:

XLWriteArray ("\\XLLinks.xls", "Output", "\$H\$" + RowNumber, ProcessTime) !将变量ProcessTime的值写入与模型系统路径相同的工作簿XLLinks.xls中的工作 表Output中去,表格由H列和变量RowNumber行所确定;

RowNumber = RowNumber + 1 !行数变量加1;

- 7.8.5.3 对 machine 元素 machine1 的详细定义:
- 属性定义:
 - ◆ machine1.Type=Single
 - ◆ machine1.Cycle Time= CycleTime (1) !由变量 CycleTime (1)来决定服务 时间;

 - ◆ machine1. Quantity=2; ! 设置工作台 machine1 的台数为 2 台;
- 规则定义:

machine1.Input Rules (From):

◆ PULL from Store

machine1.Output Rules (To) :

- ◆ PUSH to Machine2
- 活动定义:

machine1.actions on input :

```
TemporaryValue = PartSize
```

```
CellValue = 6 + N
```

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue) 本段程序通过中间变量TemporaryValue和CellValue,在EXCEL工作表Output的 特定区域中写入工作刚刚进入machine1的机器的下标以及对应工作的属性PartSize 的值。

machinel.actions on finish:

```
TemporaryValue = ""
```

```
CellValue = 6 + N
```

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue)

本段程序通过中间变量 TemporaryValue 和 CellValue, 在 EXCEL 工作表 Output 的特定区域中实现如果 machinel 中特定的机器完成的工作,转入空闲状态,就擦除 工作表中的对应数据。

7.8.5.4 对 machine 元素 machine2 的详细定义:

- 属性定义:
 - machine2.Type=Single



- ◆ machine2.Cycle Time= CycleTime (2) !由变量 CycleTime (2)来决定服务 时间;
- ◆ machine2. Quantity=3; ! 设置工作台 machine2 的台数为 3 台;
- 规则定义:

machine2.Input Rules (From) :

♦ WAIT

Machine2.Output Rules (To) :

- PUSH to Machine3
- 活动定义:

machine2.actions on input:

```
TemporaryValue = PartSize
```

```
CellValue = 6 + N
```

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue) 本段程序通过中间变量TemporaryValue和CellValue,在EXCEL工作表Output的 特定区域中写入工作刚刚进入machine2的机器的下标以及对应工作的属性PartSize 的值。

Machine2.actions on finish:

- TemporaryValue = ""
- CellValue = 6 + N

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue)

本段程序通过中间变量 TemporaryValue 和 CellValue,在 EXCEL 工作表 Output 的特定区域中实现如果 machine2 中特定的机器完成的工作,转入空闲状态,就擦除 工作表中的对应数据。

7.8.5.5 对 machi ne 元素 machi ne3 的详细定义:

- 属性定义:
 - machine3.Type=Single
 - ◆ machine3.Cycle Time= CycleTime (3) !由变量 CycleTime (2)来决定服务 时间;
 - ◆ machine3. Quantity=1; ! 设置工作台 machine3 的台数为1台;
- 规则定义:

machine3.Input Rules (From) :

```
♦ WAIT
```

Machine3.Output Rules (To) :

• PUSH to SHIP

```
    活动定义:
```

machine3.actions on input :

```
TemporaryValue = PartSize
```



CellValue = 6 + N

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue) 本段程序通过中间变量TemporaryValue和CellValue, 在EXCEL工作表Output的 特定区域中写入工作刚刚进入machine3的机器的下标以及对应工作的属性PartSize 的值。

Machine3.actions on finish :

TemporaryValue = ""

CellValue = 6 + N

XLWriteArray ("\\XLLinks.xls", "Output", "\$C\$" + CellValue, TemporaryValue)

本段程序通过中间变量 TemporaryValue 和 CellValue,在 EXCEL 工作表 Output 的特定区域中实现如果 machine3 中特定的机器完成的工作,转入空闲状态,就擦除 工作表中的对应数据。

7.8.6 仿真运行及结果分析

该模型的仿真运行及结果请参看随书附带光盘中的 models/XLLinks.xls 文件。



第八章 归档器及其应用

WITNESS 提供了归档器模块 (Documentor), 让用户提取模型信息以及其中的 元素的信息,形成报告文件,以便打印出来或是保存起来。

归档器所提取的报告主要包括:元素的备注、规则、事件、中断和其常用属性 信息,但是不包括统计信息。统计信息由仿真模型运行过程中的 reporting 菜单获得。

下面通过实例来介绍归档器的使用。假设需要形成报告的模型是前面章节中提 到的流水线平衡分析模型。则归档器的使用要经过如下步骤:

- 打开 WITNESS;
- 打开并运行模型程序 stage6.mod;
- 选择菜单 model/Documentor...命令;
- 在 Documentor 对话框中选择需要形成报告的数据;
- 点击 Documentor 对话框中的 Generate 按钮,生成报告文件;
- 打印或保存报告。

8.1 归档器对话框介绍

在本例中,前两步操作很简单,这里省略掉了。在使用归档器的第三步时,将 弹出 Documentor 对话框,界面如图 8.1 所示。

| Itens: | Model Title Information: | Report: |
|---|--|--|
| • Title Information · · · · · · · · · · · · · · · · · · · | Titl Base Model. Author Lanser Group. | Creation Date: Title Information Nodel Notes |
| User Actions | ▼ Frahled | Produce - [Name] |
| flements ff Ci | - Options: Some Add> | Produce - [General] |
| 1 00 | W Mgtes Delete | Produce = [Labor] |
| Inspect | ✓ General groperti | Produce - [Actions] |
| Dutpat | ₩ Interruptio | |
| Wash Yash Yeigh | Actions | |
| T Widget | Sort Bogn | |
| | Eilte German | 1 (Tere 1 147a |

图 8.1 归档器对话框界面



归档器主要分为左中右三大部分。左边"Items"中的"General"列表框中列举 了有关于仿真项目的一些信息,它们是:项目标题(Title Information),创建日期 (Creation Date),模型备注(Model Notes),项目初始化程序(Initial Action),用户 自定义程序(User Actions)五项内容;左边"Items"中的"Elements"列表框中列 举了仿真项目中的所有元素。中间部分的模型标题信息"Model Title Information" 中列举了当前仿真项目的名称、标题和作者信息;分割符选项 Separator;元素信息 选项"Options"复选框中提供了有关于元素的名称(Name),备注(Notes),通用 属性(General Properties),输入输出规则(I/O Rules),中断(Interruption),劳动者 规则(Labor Rules),活动(Actions)这些选项;元素是按照名称还是类型排序。右 边的"Report"中显示已经被选择的用来生成报告的信息列表。

8.2 选择形成报告的数据

在 WITNESS 模型归档器中,可以定义两种类型的报告数据:仿真项目的通用 数据和模型中各个元素的相关数据。

8.2.1 添加仿真对象数据

在设定报告内容时,从左边选定相应的对象,可以直接通过鼠标单选对象,也 可以通过按着 "shift"键和鼠标连续选择相邻的对象以及按着 "ctrl"键和鼠标进行 有间隔对象的选择。在选定 general 系列的通用数据对象之后,直接点击 "Add→" 按钮;在选定 elements 窗口中的模型元素对象之后,需要在 option 窗口选择元素对 象的数据类型,然后点击 "Add→"按钮;这样所选定的通用数据对象、元素对象及 其数据类型名称将显示在右边的 "Report"框中。

8.2.2 删除所选定的对象

如果选择对象错误,可以将它从 report 列表中删除。选定需要删除的对象,点击"Delete"按钮,所选对象就被从 report 列表中删除。

8.2.3 重排 report 列表中对象的次序

归档器对话框中的 report 列表允许对其中的对象次序进行重新排列。选择 report 列表中的对象,然后点击左边的 " Up " 和 " Down " 按钮,对象的位置将上下移动。

8.3 生成报告

在对报告数据设定完毕之后,将要生成报告。点击归档器下方的"Generate"按钮,将弹出 report 对话框如图 8.2。



| Sun Jun 20 16:31:05 2004 | |
|--------------------------|--|
| | |
| Stage6 | |
| Base Model. | |
| Lanner Group. | |
| | |
| Machine | |
| 2 | |
| Lowest | |
| Single | |
| 3.0 | |
| | |
| PULL from C2(N) at Front | |
| PUSH to C3 at Rear | |
| | |
| Operator | |
| Operator | - |
| | Sun Jun 20 16:31:05 2004 Stage6 Base Model. Lanner Group. Machine 2 Lowest Single 3.0 PULL from C2 (N) at Front PUSH to C3 at Rear Operator |

8.2 归档器生成的报告对话框

选择的报告数据将显示在 Report 对话框中。此时,可以直接对报告文件进行编辑,在窗口中单击鼠标右键,将弹出编辑快捷菜单,其中包括"Cut"、"Copy"、"Paste"、"Delete"、"Undo";可以按"Print"按钮将报告打印出来,也可以按"Save"按钮将报告保存为*.rtf格式文件。



第九章 优化器简介和应用

在仿真一个实际系统的时候,往往希望得出系统绩效最优时的系统配置;或者 希望得出系统绩效相对较好的几种方案的系统配置,通过比较和权衡,选择其中的 一种配置进行实际的实施。比如前面的生产线,给定一个利润目标函数 = 产出量 × 价格 - 设备成本 - 人力成本,它同产出率成正比,同设备数量和人员数量成反比, 在特定约束条件下,目标函数越大越好。为了得出目标函数的最大值,可以通过逐 步的改变系统中设备和人员数量,然后仿真运行,比较每次仿真运行的利润目标函 数。如果每种设备(共有四种)的数量可以是1台或2台,人员数量可以是1或2 人,则系统的配置方案就有2×2×2×2×2=32种,在进行方案比较时,就需要改 动并运行模型 32次,效率较低。

为了解决这一问题,Lanner 公司通过使用最新的优化技术研究成果以及启发式 算法,开发了一种独特的优化算法模块 OPTIMIZER,通过使用该模块,设定了目标 函数以及模型元素可能选择值,就可以连续的、按照特定算法快速有效的搜索出模 型的最优解决方案。

下面将通过一个优化实例来深入了解 OPTIMIZER 的作用及其操作方法。

9.1 示例模型流程介绍

有一家组装电脑的小型公司,通过促销广告来吸引订单,根据客户订单来决定 生产、检测、发运。其 WITNESS 仿真模型如图 9.1 所示,可以参看随书光盘 models 目录下的 Bprbase.mod。



图 9.1 优化示例模型的可视化界面



该系统的处理流程如下:

- 客户响应广告,提出订单 Request;
- 每个订单请求 Request 需要员工组 A (StaffA)或员工组 B 中的一名员工使用电脑或纸质记事簿记录下来,电脑和记事簿的数量可能是 2 到 3 个;究竟是需要员工 A 还是员工 B 进行记录。
- 在必要时候,需要同客户就订单问题进行电话沟通,这儿有 3 或 4 部电话,由 员工组A或员工组B进行联系;
- 电脑由工程师 (Engineer) 进行组装 (Build);
- 电脑由工程师或员工组 B 进行检测 (Test); 检测时间需要 45 或 50 分钟, 90% 的电脑通过检测, 10%的电脑需要重新组装;

9.2 优化变量

对该系统涉及到的变量及其意义和取值列于表 9.1,系统通过这些组合来得出获 得最优目标函数值的变量取值。

| 变量名称 | 说明 | 变量可能取值 | 变量取值数 | 累积组合 |
|-------------------|-------------|------------|-------|-------|
| Logdist.value | 记录订单所需的时间值, | 10 or 15 | 2 | 2 |
| | 取决于电脑记录还是记录 | | | |
| | 簿记录 | | | |
| Testcyc.value | 检测时间,取决于检测者 | 50 or 45 | 2 | 4 |
| Choice1.value | 选择员工进行订单记录 | 0, 1, or 2 | 3 | 12 |
| Choice2.value | 选择员工或工程师进行检 | 0 or 1 | 2 | 24 |
| | 测操作 | | | |
| StaffA.quantity | A 类员工的数量 | 3到5个 | 3 | 72 |
| StaffB.quantity | B 类员工的数量 | 3到7个 | 5 | 360 |
| Engineer.quantity | 工程师数量 | 2到4个 | 3 | 1080 |
| Act001.quantity | 订单记录站点数量 | 2 or 3 | 2 | 2160 |
| Act002.quantity | 回访电话机数量 | 3 or 4 | 2 | 4320 |
| Act003.quantity | 检测站点数量 | 3到7个 | 5 | 21600 |

表 9.1 优化变量列表

9.3 相关时间值

- 订单到达:1台/10分钟;
- 订单记录时间:取决于记事簿记录还是电脑记录,记录时间服从均值为Logdist,标准差为Logdist/10的正态分布,记事簿记录时,Logdist = 15;电脑记录时,Logdist = 10;对应设置于Act001的加工时间:NORMAL (LogDist,LogDist / 10.0,100 + N);
- 订单确认时间:服从最小值为 10,最大值为 40 的均匀随机分布;对应设置于 Act001 的加工时间:UNIFORM (10.0,40.0,200 + N);



- 组装时间:服从最小值为10,最大值为20的均匀随机分布;对应设置于Act001 的加工时间:UNIFORM (20.0,40.0,400 + N)
- 检测时间:取决于检测机器的类型,或 50 分钟,或 45 分钟;对应设置于 Act001 的加工时间:TestCyc。

9.4 员工选择规则

9.4.1 订单记录处理

根据变量 choice1 的值的不同,选择 StaffA 或 StaffB。如果 choice1 = 0,则仅有 StaffA 可以进行订单记录处理;如果 choice1 = 1,则既可以 StaffA 也可以是 StaffB 进行订单记录;如果 choice1 = 2,则仅有 StaffB 可以进行订单记录处理。对应设置 Act001 的 Labor Rule 规则如下:

IF Choice 1 = 0

StaffA#1

```
ELSEIF Choice 1 = 1
```

StaffA#1 OR StaffB#1

ELSE

```
! Choice 1 = 2
StaffB#1
```

ENDIF

9.4.2 订单确认处理

既可以 StaffA 也可以是 StaffB 进行订单确认处理。对应设置 Act002 的 Labor Rule 规则如下:

StaffA#1 OR StaffB#1

9.4.3 电脑组装处理

组装工作仅有 Engineer 可以处理。对应设置 Act003 的 Labor Rule 规则如下: Engineer#1

9.4.4 检测处理

根据变量 choice2 的值的不同,选择 Engineer 或 StaffB。如果 choice2 = 0,选择 B 类员工进行检测;如果 choice2 = 1,既可以 Engineer 也可以是 StaffB 进行检测。 对应设置 Act004 的 Labor Rule 规则如下:



IF Choice2 = 0

StaffB#1

ELSE

! Choice2 = 1 StaffB#1 OR Engineer#1

ENDIF

9.5 目标函数

进行优化模块的运行,必须设定目标函数,系统根据目标函数值的比较进行算 法搜索计算。本例中的目标函数计算总的收益减去生产成本所得的净利润。

9.5.1 函数体程序及其解释

```
RCount = 0.0
               !变量初始化
RCount = NSHIP (Request) * 100.0 ! 计算收益,每台电脑价格为 100
RCount = RCount - NQTY (Engineer) * 80.0 - NQTY (StaffA) * 60.0 - NQTY (StaffB) *
60.0
      !收益减去员工的工资,工程师每天工资80,员工工资每天60!
IF LogDist = 15
   LogCost = 1.0 !记事簿记录,每次需要成本为 1.0
ELSE
   LogCost = 50.0 ! 电脑记录,每次需要成本为 50.0
ENDIF
IF TestCyc = 50
   TestCost = 50.0 ! 使用 A 类检测设备,检测一台电脑的成本
ELSE
   TestCost = 60.0 !使用 B 类检测设备,检测一台电脑的成本
ENDIF
RCount = RCount - NQTY (Act001) * LogCost - NQTY (Act003) * TestCost
!收益减去记录的成本和检测的成本,得到净收益!
RETURN Rcount !返回净收益值
```

9.5.2 系统函数介绍

9.5.2.1 NSHIP(part_name)

功能:返回被送出系统SHIPPED的某种零部件元素的数量。 参数:part_name,名型,需要统计的 part 元素的名称。 示例:BILLABLE = NSHIP(CARTON)



在这个例子中,NSHIP 函数计算出被发送出系统的 CARTON 零部件的数量,然 后将值赋给变量 BILLABLE。

9.5.2.2 NOTY (element_name)

功能:返回模型系统中,特定元素的数量。

参数:element_name,名型,需要统计的元素的名称。

示例:TOTAL_DRILLS = NQTY(DRILL)

在这个例子中, NQTY 函数统计出模型系统中元素 DRILL 的数量, 然后将值 赋给变量 TOTAL_DRILLS。

注:该函数不能够用来统计模型中的 part 元素的数量;为了统计模型中 part 元素的数量,使用 NWIP 函数来统计模型中某种 part 元素的在制品数量。

9.6 优化

从变量表上可以看出,本模型的配置方案一共有 21600 种,采用手工方式,逐 个方案进行设计和运行,显然不太显示。而使用优化模块就能够很方便的得出优化 方案。

9.6.1 优化步骤

- 打开 WITNESS;
- 打开 WITNESS 安装路径下的 optimizer 文件夹下的 Bprbase.mod;
- 点击菜单 model/optimize...,将弹出 Model Optimization 4.0 设置窗口;
- 在 Model Optimization 4.0 设置窗口中进行设置;
- 优化运行;
- 结果分析。

9.6.2 Model Optimization 4.0 设置窗口介绍

Model Optimization 4.0 设置窗口如图 9.2 所示,窗口上有两个设置页框:Scenario和 Tracking。

9.6.2.1 Tracking页框

允许设置最优结果的显示范围,例如显示前100个最优方案数据,还是显示前10 个最优方案数据,只要在 Show Best Results 下面的文本框中填入100或者10即可。

9.6.2.2 Scenario页框

包括变量 Variables、目标函数 Objective Function、结果池 Answer Pool、运行控制 Run Control、算法 Algorithm、对话按钮六个区域。



| Model Optimization 4.0 | | | |
|--|---|--|--|
| <u>Variables</u> <u>LogDist .Value {15, 10}</u> TestCyc .Value {50, 45} choice1 .Value {0, 1, 2} choice2 .Value {0, 1} StaffA .Quantity {3 to 5} StaffB .Quantity {3 to 7} Engineer .Quantity {2 to 4} Act001 .Quantity {2 to 3} | | Add Change Delete Delete All Add Object Functi ObjFuncti ObjFuncti Object Functi Object Functi Object Functi | ive o n v Result is aximum inimum |
| Total Constrained | 21600 21600 | onstraints Result Re-evaluate Clea | t O r Test |
| Run Control Run 0.0 1440.0 Runs per 1 1 Sample Ru Abort Multiple Random Number (%) Average Evalu 100 Average Evalu | n Algorit Adaptiv etting: rs Estimat 400 eation Estimat Unknown | hm ve Thermostatistical s Optimize ed ed Total | Analyze Open Save As Close Help |

图 9.2 Model Optimization 4.0 设置窗口

● 变量 Variables 区域显示当前优化方案的变量列表,通过相关按钮可以进行变量 的添加 Add、修改 Change、删除 Delete、约束条件 Constraints 等设置。按钮作 用如表 9.2 所示。

表 9.2 优化对话框上按钮的作用表

| 按钮名称 | 按钮功能 |
|-------------|--------------------------------|
| Add | 显示添加变量对话框,可以添加一个新的变量 |
| Change | 显示改变变量对话框,可以改变当前所选变量的相关设置 |
| Delete | 删除当前所选变量 |
| Delete All | 提示、删除当前列表中的所有变量 |
| Constraints | 显示优化约束对话框,可以制定当前设定的变量之间的相互约束关系 |
| Re-evaluate | 计算和显示在制定的约束条件下的方案组合数 |

- 目标函数 Objective Function 区域包括一个下拉列表框,显示当前模型的可用目标函数列表,在这里选择需要优化的目标函数;还包括一个最优结果 best result is 区域,选择目标函数是越大越好 Maximum,还是越小越好 Minimum。
- 结果池 Answer Pool 区域显示当前优化方案的可用结果,也就是优化运行所保存 的最优结果数。选择 Clear 按钮清空结果池,选择 Test 按钮结果池中的数据是否 与当前的模型一致。
- 运行控制 Run Control 区域的 Warmup 文本框指定仿真模型运行稳定所需要的时间, Run 文本框指定每种方案的仿真时间等。


● 算法 Algorithm 区域用来设定所选的算法程序(共有六种)以及设置 setting 按 钮和优化 Optimize 按钮。

9.6.3 进行优化设置

- 在优化器对话框中选取目标函数 objfun,同时选定 Maximum,即优化搜索朝着 该函数值可能增大的方向;
- 设定表 9.1 中的优化变量的可能取值情况添加入优化器对话框,如图 9.2 所示;
- 优化仿真一天时间该公司的利润情况,设定运行时间为1440分钟;
- 优化算法采用 Adaptive Thermostatistical SA 模拟退火算法。
- 可以通过 Model Optimization 4.0 设置窗口的 Open...按钮打开随书光盘 models 目录下的 BPRBase.opt,来获得参考。

9.6.4 运行优化与结果分析

在经过上一步的优化设置之后,就可以进行优化的运行了。点击"Optimize…" 按钮,激活"Run Optimization"对话框,点击该对话框上的"Run"按钮,开始运 行优化。

在运行过程中,系统提供了当前输入变量的值"Input Variables"、目标函数的结果的图形"Objective Graph"、变量和函数的对应表"Result Table"的动态显示。

通过仿真运行得到最优的五种方案如表 9.3 所示。

| ObjFun | LogDist .Value | TestCyc .Value | choice1 .Value | choice2 .Value | StaffA .Quantity | StaffB .Quantity | Engineer .Quantity | Act001 .Quantity | Act002 .Quantity | Act003 .Quantity |
|--------|-------------------|-------------------|-------------------|-------------------|---------------------|---------------------|-----------------------|---------------------|---------------------|---------------------|
| 11758 | 15 | 45 | 0 | 0 | 4 | 6 | 4 | 2 | 3 | 7 |
| 11680 | 10 | 50 | 1 | 0 | 4 | 6 | 4 | 3 | 4 | 7 |
| 11668 | 15 | 50 | 1 | 0 | 4 | 7 | 4 | 2 | 3 | 7 |
| 11668 | 15 | 50 | 1 | 0 | 4 | 7 | 4 | 2 | 4 | 7 |
| 11667 | 15 | 50 | 1 | 0 | 4 | 7 | 4 | 3 | 3 | 7 |

表 9.3 仿真运行最优的五种方案

完成 65 次优化搜索之后,形成的目标函数曲线如图 9.3 所示。



图 9.3 完成 65 次搜索,所形成的目标函数的图形

可以看出,只要经过 64 次 Adaptive Thermostatistical SA 搜索, WITNESS 就得 到了最优的结果 11758 元/每天。

根据这五种最优方案,以及企业的实际情况,可以辅助企业进行系统配置方案 的决策。



第十章 多产品多阶段加工仿真系统设计

10.1 加工系统描述

有一个制造车间由 5 组机器组,加工三种产品。每种产品分别要求完成 4,3 和 5 道工序,而每道工序必须在指定的机器组上,按照事先规定好的工艺顺序进行。 仿真目的:

- 熟悉系统元素 ROUTE 的用法。
- 了解工艺视图 process views 的用法。
- 了解多产品多阶段加工仿真系统设计

假定在保持车间逐日连续工作的条件下,来仿真 365 个 8 小时工作日的工作, 计算每种产品在队列中的平均总等待时间和作业总平均等待时间,以及每组机器队 队列中的平均作业数、平均利用率以及平均等待时间。并试图进行改善。

该模型可以参看随书光盘 models 目录下的 MulStageMulProduct.mod 文件。

10.2 加工系统数据

第 1,2,3,4,5 组机器分别有 3,2,4,3,1 台相同的机器,三种产品原料 到达车间的间隔时间分别服从均值为 50,30,75 分钟的指数型随机变量。

三种产品的工艺路线如表 10.1 所示。于是,第1种作业首先在第3组机器上加工,然后在第1组、再后来在第2组机器上加工,最后在第5组机器上完成最后工序,

| 产品类型 | 机器组别 | 相继工序平均服务时间/MIN |
|------|-----------|-------------------|
| 1 | 3,1,2,5 | 30 , 36 , 51 , 30 |
| 2 | 4,1,3 | 66 , 48 , 45 |
| 3 | 2,5,1,4,3 | 72,15,42,54,60 |

表 10.1 产品加工工艺路线与各工序加工时间参数

如果一项作业在特定时间到达车间,发现该组机器全都忙着,该作业就在该组 机器处排入一个 FIFO 规则的队列,如果前一天没有完成的任务,第二天继续加工。 在特定机器上完成一个工序的时间是一种按二阶爱尔朗分布的随机变量,它的平均 值取决于作业的类别以及机器的组别。每种作业的每道工序的平均服务时间如表 1 所示,于是,一个第 2 类作业在第 4 组机器上(第一道工序)的平均服务时间要 66 分钟。



10.3 仿真模型的建立

10.3.1 元素定义 define

本系统的元素定义如表 10.2 所示。

| | 表 10.2 | 头1 | 净兀 紊正乂 |
|----------|---------|----|---------------|
| 元素名称 | 类型 | 数量 | 说明 |
| А | Part | 1 | 产品 A |
| В | Part | 1 | 产品 A |
| С | part | 1 | 产品 A |
| Machine1 | machine | 3 | 机器组1 |
| Machine2 | Machine | 2 | 机器组2 |
| Machine3 | Machine | 4 | 机器组3 |
| Machine4 | Machine | 3 | 机器组 4 |
| Machine5 | Machine | 1 | 机器组5 |
| Buffers1 | Buffer | 1 | 机器组1的输入缓冲区 |
| Buffers2 | Buffer | 1 | 机器组2的输入缓冲区 |
| Buffers3 | Buffer | 1 | 机器组3的输入缓冲区 |
| Buffers4 | Buffer | 1 | 机器组4的输入缓冲区 |
| Buffers5 | Buffer | 1 | 机器组5的输入缓冲区 |

₹10.2 实体元素定义

10.3.2 元素显示 display 的设置

各个实体元素的显示特征定义设置如图 10.1 所示。





图 10.1 各个实体元素的显示特征

10.3.3 各个元素细节 (detail) 设计

| | 表 10.3 产品 | 品的 general 页细节 | 殳 计 | |
|------|---------------|----------------|---------------|--|
| 产品名称 | Arrivals type | Inter arrival | to | |
| А | Active | NEGEXP (50,11) | Push to route | |
| В | Active | NEGEXP (30,11) | Push to route | |
| С | Active | NEGEXP (75,11) | Push to route | |

产品 A、B、C 的加工路径 ROUTE 设计:通过双击各个产品图标,在弹出式页 框中选择其 ROUTE 页, 如图 10.2 所示, 分别进行设计。



| ₩\$产品 | | |
|-------|--|---|
| N | Bacing Bacing | |
| ж | Detail Part - A | × |
| 11 | General Attributes Route Actions Reporting Notes | |
| c | 法中部回开发页 | |
| • | Stage: | |
| | Add/Remove | |
| | Destination: Symmetrice | |
| | R_SETUP: | |
| | R_CYCLE: 30 Pick from Display | |
| | | |
| | | |

图 10.2 ROUTE 页框界面

| 机器组名称 | From | Cycle time | to |
|----------------------------|--------------------|--------------------|---------------|
| Machine1 | PULL from Buffers1 | ERLANG | Push to route |
| | | $(R_CYCLE, 1, 1)$ | |
| Machine2 | PULL from Buffers2 | ERLANG | Push to route |
| | | $(R_CYCLE, 1, 2)$ | |
| Machine3 PULL from Buffers | | ERLANG | Push to route |
| | | $(R_CYCLE, 1, 3)$ | |
| Machine4 | PULL from Buffers4 | ERLANG | Push to route |
| | | $(R_CYCLE, 1, 4)$ | |
| Machine5 | PULL from Buffers5 | ERLANG | Push to route |
| | | $(R_CYCLE, 1, 5)$ | |

|--|

表 10.4 产品加工路径 route 页细节设计

| А | | | В | | | С | | |
|-------|------------------|---------|-------|------------------|---------|-------|------------------|---------|
| Stage | Destin_ ation | R_cycle | Stage | Destin_ ation | R_cycle | Stage | Destin_ ation | R_cycle |
| 1 | Buffer3 | 30 | 1 | Buffer4 | 66 | 1 | Buffer2 | 72 |
| 2 | Buffer1 | 36 | 2 | Buffer1 | 48 | 2 | Buffer5 | 15 |
| 3 | Buffer2 | 51 | 3 | Buffer3 | 45 | 3 | Buffer1 | 42 |



| 4 | Buffer5 | 30 | 4 | Ship | 0 | 4 | Buffer4 | 54 |
|---|---------|----|---|------|---|---|---------|----|
| 5 | Ship | 0 | | | | 5 | Buffer3 | 60 |
| | | | | | | 6 | Ship | 0 |

10.3.4 工艺流程图的显示

工艺流程图也称全局工艺流程图,是 WITNESS 提供的模型简化图。在工艺流 程图中,模型中的每个元素由一个方框和元素名称来表示,如果某一元素的数量多 于1个,也仅仅显示一个框图;各元素之间的输入输出关系由有方向的箭线表示; 不过它不显示元素中所包含的函数或变量等逻辑元素。



图 10.3 process view : Global

使用方法:选择 VIEW 菜单 - >Process Views...;得到 process views 对话框,在 该对话框中设定工艺流程图显示窗口、字体颜色、布局方式等内容。

设计完毕后,得到工艺流程图如图 10.3 所示。

10.4 模型运行和数据分析

模型仿真钟取系统默认的 1:1min,运行 365×8×60=175200 仿真时间单位, 使用系统提供的 report 工具。得到下列统计报表。

| | | 表 10.6 | 产品统计 | 信息 | |
|----------|-------------|-------------|--------|------------|-------------|
| Nam e | No. Entered | No. Shipped | W.I.P. | Avg W.I.P. | Avg Time |
| А | 3466 | 3417 | 49 | 22.03 | 1113.59 |
| В | 5940 | 5889 | 51 | 47.57 | 1403.09 |

| U) | ₩₩ 北京威特尼斯科技中心 | | | | | | | | | | |
|------|---------------|----------|---|----------|-----------|----|---------|---------|----------|-------------|-----|
| • | C 2292 | | | | 2241 | | 51 | | 27.86 | 2129.43 | |
| | | | | 表 10. | 7 机 | 器约 | 且统计 | 信息 | | | |
| | | Name | | % Idle | % Busy | | No. O | f Oper | ations | | |
| | | Machine | 1 | 3.98 | 96.02 | 2 | | | 11616 | | |
| | Machine2 | | 2 | 4.94 | 95.0 | 6 | | | 5682 | | |
| | Machine3 | | 3 | 27.58 | 72.4 | 2 | 11595 | | | | |
| | Machine4 | | 4 | 2.08 | 97.92 | 2 | | | 8144 | | |
| | | Machine | 5 | 21.22 | 78.7 | 8 | 5672 | | | | |
| | | | | 表 10 | .8 缓冲 | 中区 | 区统计位 | 言息 | | | |
| Nam | e | Total In | Т | otal Out | Now | In | Ma x | Mi n | Avg Size | Avg Time | |
| Buff | ers1 | 11620 | | 11619 | | 1 | 73 | 0 | 18.6 | 5 281 | .15 |
| Buff | ers2 | 5757 | | 5684 | , | 73 | 91 | 0 | 17.40 | 5 531 | .47 |
| Buff | ers3 | 11607 | | 11599 | | 8 | 30 | 0 | 1.39 | 9 20 | .91 |
| Buff | ers4 | 8194 | | 8147 | 4 | 47 | 142 | 0 | 44.2 | 5 946 | .17 |

从表 7 可以看出,产品在系统中的时间平均在 1000 多分钟,分别为 A:1113.59, B:1403.09,C:2129.43,时间太长。再从表 8 中可以看出机器组 1、2、4 是瓶颈, 开动率达到 95%以上。

5673

Buffers5

5682

9

35

0

4.31

132.78

10.5 改 善

通过上一部分的分析,对机器组 1、2、4 各添加一台机器,再运行 175200 时间 单位,改善后的运行统计报表如下。

表 10.9 产品统计信息

| Nam e | No. Entered | No. Shipped | W.I.P. | Avg W.I.P. | Avg Time |
|----------|-------------|-------------|--------|------------|-------------|
| А | 3466 | 3464 | 2 | 6.46 | 326.34 |
| В | 5940 | 5933 | 7 | 8.03 | 236.85 |
| С | 2292 | 2283 | 9 | 6.02 | 460.02 |

表 10.10 机器组统计信息

| Name | % Idle | % Busy | No. Of Operations |
|----------|--------|-----------|-------------------|
| Machine1 | 27.7 | 72.3 | 11684 |

| Machine2 | 35.64 | 64.36 | 5751 |
|----------|-------|-------|-------|
| Machine3 | 27.2 | 72.8 | 11682 |
| Machine4 | 25.89 | 74.11 | 8223 |
| Machine5 | 20.38 | 79.62 | 5751 |

表 10.11 缓冲区统计信息

| Name | Total In | Total Out | Now In | Ma x | Mi n | Avg Size | Avg Time |
|----------|----------|-----------|--------|---------|---------|----------|-------------|
| Buffers1 | 11693 | 11688 | 5 | 22 | 0 | 1.32 | 19.71 |
| Buffers2 | 5756 | 5754 | 2 | 12 | 0 | 0.66 | 20.02 |
| Buffers3 | 11685 | 11685 | 0 | 22 | 0 | 1.41 | 21.11 |
| Buffers4 | 8224 | 8224 | 0 | 26 | 0 | 1.69 | 36 |
| Buffers5 | 5751 | 5751 | 0 | 44 | 0 | 3.94 | 119.97 |

可以看出,产品在系统中的通过时间将为原来的 0.22。可见系统的柔性和顾客满 意度得到了较大的提高。

第十一章 垃圾回收物流仿真系统设计

11.1 垃圾回收物流系统介绍

城市某一区域,共有7个居民小区,每个小区有一个固定垃圾投放处,两个垃 圾箱分别投放完全废弃垃圾和可回收利用垃圾;每天有专门公司(垃圾处理公司) 派运输车收集垃圾,将垃圾从居民区运送之中转站,再由中转站运至目的地—垃圾 处理中心。

仿真程序研究如何设计物流系统,能够使收集系统在满足时间约束、载重约束 的条件下,使垃圾处理公司的物流总成本最低。系统配置的项目主要有车辆载重量、 随车工作人员数、客户满意度。

本节内容可以参看随书光盘 models/垃圾回收仿真文件夹下的 dumpreversesim.mod,模型运行结果参看此目录下的 reverse.xls 文件。



11.2 数据信息

仿真系统涉及的数据信息如下:

11.2.1 地理及需求信息

各个收集点所有的人口数、距离垃圾转运站、公司停车场的距离信息如表 11.1 所示。

各收集点之间以及停车场和转运站之间的距离信息如表 11.2 所示。

| | 衣 | 11.1 收集只信息 | |
|---------|---------------|---------------------|------------------|
| 收集点 | 收集点居民数 | 距离停车场距离 | 距离垃圾转运站距离 |
| 名称 | people(i)(千人) | distance(ip) (km) | distance(it)(km) |
| Dump001 | 1.5 | 6 | 16 |
| Dump002 | 1.45 | 16 | 27 |
| Dump003 | 1.65 | 26 | 24 |
| Dump004 | 1.85 | 38 | 36 |
| Dump005 | 1.4 | 28 | 26 |
| Dump006 | 2.05 | 16 | 14 |
| Dump007 | 1.6 | 10 | 8 |

= 11 1 旧住上伫白

表 11.2 点间距离

| 路径起点 | 路径终点 | 距离(km) | 符号 |
|--------------|-----------|--------|--------------|
| Tran_station | Corp park | 11 | Distance(pt) |
| Dump001 | Dump002 | 10 | Distance(12) |
| Dump002 | Dump003 | 10 | Distance(23) |
| Dump003 | Dump004 | 12 | Distance(34) |
| Dump004 | Dump005 | 10 | Distance(45) |
| Dump005 | Dump006 | 12 | Distance(56) |
| Dump006 | Dump007 | 6 | Distance(67) |

11.2.2 与收集车辆有关的信息

- 固定成本[price(i)]: 收集车辆的购买费用; •
- 可变成本[cost(i)]:车辆的行驶费用、维护费用和雇员工资; •
- 行驶速度:分为空车行驶速度[unloadspeed(i)]和载重行驶速度[loadspeed(i)];
- 收集时间[loadtime(i)]:在垃圾收集点收集垃圾所需的时间
- 卸载时间[unloadtime(i)]: 在转运站卸载垃圾所需的时间 具体信息如表 11.3 所示:



| 车辆 载重 固定成本 可变成本 载重速度 空载速度 人员 人员工资 收集 名称 t 万元 元/km km/h km/h 人 元/天.人 mi | 表 11.3 收集车辆信息 | | | | | | | | | |
|--|------------------|-----|-----------------------|----------------|------|------|-----------------------|-------|-----------------------|---|
| 名称 t 万元 元/km km/h km/h 人 元/天.人 mi | 两 载重 | 车辆 | 固定成本 | 可变成本 | 载重速度 | 空载速度 | 人员 | 人员工资 | 收集时间 | 卸载时间 |
| | 尔 t | 名称 | 万元 | 元/km | km/h | km/h | 人 | 元/天.人 | min | min |
| $4 = 1$ X_1 $2 + X_1$ X_1 60 72 P_1 60 $3 + y * (y + y) = 0$ | 1 X ₁ | 车1 | 2+ X ₁ | X_1 | 60 | 72 | P ₁ | 60 | 3+y*6/ P ₁ | 3+ X ₁ *6/ P ₁ |
| $4 = 2$ X_2 $2+1.2*X_2$ X_2 60 72 P_2 60 $2+y*2$ | 2 X ₂ | 车 2 | 2+1.2* X ₂ | X ₂ | 60 | 72 | P ₂ | 60 | 2+y*8/P ₂ | 2+ X ₂ *6/ P ₂ |

说明:

车1用来收集不可回收垃圾;车2用来收集可回收利用垃圾;

 X_1 , X_2 为现有两种车的吨位数, X_1 取值: 3, 4 或 5 吨; X_2 取值: 1.5; 2; 2.5; y为每次收集的垃圾量, 单位--吨;

 P_1 , P_2 为两种车可配备的随车工作人员数(可以是1,2,3)。

11.2.3 与垃圾相关的信息

人均垃圾量:平均每人每天产生的垃圾数量;

垃圾比例:各种垃圾成分所占比例。

假设人均垃圾量服从均值为 1.2kg/天的埃尔郎分布,垃圾成分分为两类:一类为可重复利用资源,一类为无用需销毁垃圾,两者比例为1:2。

11.3 收集垃圾的成本函数

现在假设仿真一年 365 天的垃圾回收工作,决定两种型号的车配置何种吨位以 及随车工作人员数,垃圾回收物流成本(Total Logistic Cost, TLC)最低。

TLC=车1成本+车2成本

车 I 所耗成本=I 的固定成本+I 的可变成本+I 的服务时间成本+I 的人员成本 即:

TLC=CC1+CC2

CC[i]=X[i]+V[i]+S[i]+P[i]*365*60

S[i]=(runtime[i]-240)*m

S[i]为垃圾回收的服务时间成本,m为收集时间对服务时间成本的惩罚因子。垃圾回收公司在十二点之前收集完毕,社会效应好,给予奖励;在十二点后完成,社会满意度低,影响公司信誉,成本增加。

runtime[i]为垃圾车将垃圾全部回收完成的最终时间,也即垃圾车每天的运行时间;

11.4 系统逻辑结构

此垃圾回收物流系统的逻辑结构分为五个子模块:垃圾产生模块、叫车模块、 收集模块、收工模块、数据处理模块。

155



- 垃圾产生模块 每天收集工作开始时,系统根据小区的人数、人均垃圾量、两种垃圾量的比例,通过爱尔朗随机分布函数,随机产生各个小区的两种垃圾数量,分别存放入各个小区的两个垃圾箱内。
- 叫车模块 在垃圾存放入各个小区的两个垃圾箱过程中,根据每种垃圾的总量
 以及每种垃圾车的载重量,生成叫车的次数。
- 收集模块 叫车次数确定之后,车辆根据需求,沿着最短路径依次收集垃圾。
 在收集过程中,在一个垃圾点,如果垃圾车收集满了或当天的收集工作全部完成,车辆驶向垃圾中转站,进行卸载操作;如果垃圾车未满,则驶向下一最近垃圾收集点继续收集。
- 收工模块 当天的收集工作全部完成之后,车辆驶向公司停车场,当天收集工 作完毕。
- 数据处理模块 每天车辆收工时,

11.5 仿真模型的建立

根据系统逻辑结构和数据信息,我们建立具有 7 个垃圾收集点、一个垃圾转运 站的 WITNESS 垃圾回收仿真模型。具体步骤如下:

11.5.1 元素定义 Difine

建立仿真模型时,首先定义仿真模型中所需的元素,并设计它们的可视效果。 我们将设计的元素分为两类 实体元素和逻辑元素,名称如下。

| 元素名称 | 类型 | 说明 |
|------------------------------|---------|-----------------------|
| $dumpa(1) \sim dumpa(7)$ | Part | 七个小区不可回收的垃圾 |
| $dumpb(1) \sim dumpb(7)$ | Part | 七个小区可回收利用的垃圾 |
| dumpcart(1) | Vehicle | 专门回收 dumpa 的垃圾回收车 |
| dumpcart(2) | Vehicle | 专门回收 dumpb 的垃圾回收车 |
| $buffera(1) \sim buffera(7)$ | Buffer | 各小区不可回收垃圾的垃圾箱 |
| $bufferb(1) \sim bufferb(7)$ | buffer | 各小区可回收利用垃圾的垃圾箱 |
| Corppark | Track | 公司停车场 , |
| Transfer_station | Buffer | 垃圾转运站 |
| road(i,j) | Track | 由节点实体 i 向节点实体 j 方向的道路 |
| road(j, i) | track | 由节点实体 j 向节点实体 i 方向的道路 |
| Geta(1)~geta(7) | Track | 车辆将通过它们来进行垃圾 a 的回收 |
| Getb(1)~getb(7) | Track | 车辆将通过它们来进行垃圾 b 的回收 |

表 11.4 实体元素列表:

表 11.5 逻辑元素——变量列表:

| 变量名称 | 类型 | 说明 |
|-----------|---------|----------|
| people(7) | integer | 各小区的居民数量 |



| capacitycart(2) | integer | 两种垃圾车每次可收集的垃圾数量,即载重量 |
|-----------------|---------|--|
| ndemand(2) | integer | 每天对两种垃圾车的需求次数 |
| labor(2) | integer | 两种垃圾车的随车工作人员数 |
| Salary | Real | 工作人员每天工资 |
| price(2) | Real | 两种垃圾车的购买价格 |
| costrate(2) | Real | 两种垃圾车的运行费率 |
| unloadspeed(2) | Real | 两种垃圾车的空载速度 |
| loadspeed(2) | Real | 两种垃圾车的实载速度 |
| loadnum (2) | integer | 两车辆到达垃圾收集点可收集的垃圾数量 |
| nfree(2) | integer | 两车辆剩余载重能力 |
| Nparts7) | integer | 每一垃圾收集点在特定仿真时点剩余垃圾量 |
| loadtime(2) | Real | 两车辆收集垃圾所需的时间 |
| unloadtime(2) | Real | 两车辆卸载垃圾所需的时间 |
| runtime(2,3) | Real | 存储车辆的时间参数,runtime(k,1)存储仿真钟的当前数 |
| | | 值 ,runtime(k,2)存储车辆 k 在当天的运行时间 ,runtime(k,3) |
| | | 存储车辆 k 在仿真过程中总的运行时间 |
| Timeweigh | Real | 时间惩罚因子 |
| timecost(2) | Real | 时间惩罚成本 |
| sumcost | Real | 目标函数 objfun 中用于统计系统运行的总费用 |
| | | |

定义一个实数型逻辑函数 objfun(),用于计算和统计系统运行一年 365 天所花费的总费用。

11.5.2 元素显示 display

dump traisfer station Corposek

各个元素的显示设置如图 11.1 所示。

图 11.1 垃圾回收物流仿真系统可视化界面



11.5.3 元素详细设计

在该子模块中定义随着仿真钟的推进,装载垃圾的逻辑条件。该系统的工作班 次制度采用每天工作 8 小时,每一仿真时间等价于现实时间一分钟。

为了实现系统的仿真运行,需要对系统中的元素进行详细设计。

系统初始化程序设计(initialize actions),通过选择系统菜单 model/initialize actions...菜单项,得到初始化程序编辑框,输入如下程序:

```
personum (1) = 1.5
personum (2) = 1.45
personum (3) = 1.65
personum (4) = 1.85
personum (5) = 1.4
personum (6) = 2.05
personum (7) = 1.6
!给各小区的人数赋初值,单位:千人
FOR num = 1 \text{ TO } 7
meandump (num) = 1.2 * \text{personum} (\text{num}) * 1000
NEXT
!分别生成每天七个小区产生垃圾的均值
set capacity of dumpcart1 to capacitycart(1)
set capacity of dumpcart2 to capacitycart(2)
!分别设定两个车辆的载重量,由变量数组capacitycart(1)和capacitycart(2)决定。
unloadtime = 0
                          !卸载时间
FOR num = 1 \text{ TO } 7
moddemanda (num) = 0
                              !决定叫车次数的变量
moddemandb (num) = 0
NEXT
FOR num = 1 \text{ TO } 4
roadchoicea (num) = 0
roadchoiceb (num) = 0
                             !最短路径控制变量
NEXT
FOR num = 1 \text{ TO } 3
runtimea (num) = 0
runtimeb (num) = 0
                             !车辆运行时间控制变量
NEXT
                             !每天的分钟数
intervaltime = 1440
timeweigh = 5
                             !时间惩罚因子
laborneeda = 1
```



laborneedb = 1 !对变量赋初始值 !两辆车随车工作人员数

● 各个小区垃圾的详细设计

表 11.6 垃圾元素详细设计列表

| Name | First arrival | Inter arrival | Lot size | То |
|------------|---------------|---------------|---|----------------------|
| Dumpa | 0.0001 | 1440 | ERLANG (meandump (1) / 3,3,1) | PUSH to buffera1 |
| 1 | | | | |
| Dumpa | 0.0002 | 1440 | ERLANG (meandump (2) / 3,3,2) | PUSH to buffera2 |
| 2 | | | | |
| Dumpa | 0.0003 | 1440 | ERLANG (meandump $(3) / 3,3,3$) | PUSH to buffera3 |
| 3 | 0.0004 | 1.4.40 | | DUCIL 1 CC 4 |
| Dumpa | 0.0004 | 1440 | ERLANG (meandump $(4)/3,3,4$) | PUSH to buffera4 |
| 4 Dumme | 0.0005 | 1440 | EDIANC (magnitum $(5)/2.2.5$) | DUCIL to have from 5 |
| 5 5 | 0.0003 | 1440 | EKLANO (meandump $(3)/(3,3,3)$ | PUSH to bulletas |
| Jumpa | 0.0006 | 1440 | ERI ANG (meandumn $(6)/336$) | PUSH to buffera6 |
| 6 | 0.0000 | 1440 | ERE/1100 (incandump (0) / 5,5,0) | |
| Dumpa | 0.0007 | 1440 | ERLANG (meandump $(7)/3,3,7$) | PUSH to buffera7 |
| 7 | | | | |
| Dumpb | 0.0001 | 1440 | ERLANG (meandump (1) / 3,3,8) | PUSH to bufferb1 |
| 1 | | | | |
| Dumpb | 0.0002 | 1440 | ERLANG (meandump (2) / 3,3,9) | PUSH to bufferb2 |
| 2 | | | | |
| Dumpb | 0.0003 | 1440 | ERLANG (meandump (3) / 3,3,10) | PUSH to bufferb3 |
| 3 | | | | |
| Dumpb | 0.0004 | 1440 | ERLANG (meandump $(4)/3,3,11$) | PUSH to bufferb4 |
| 4 D1 | 0.0005 | 1440 | EDIANC $(m = 1, m = (5) / 2, 2, 12)$ | DUCULA 1 Co. 1.7 |
| Dumpb | 0.0005 | 1440 | ERLANG (meandump $(5) / 3,3,12$) | PUSH to bufferb5 |
| J | 0.0006 | 1440 | EPI ANG (meandumn $(6) / 3 3 13$) | DUSH to bufferb6 |
| 6 | 0.0000 | 1770 | EXEARCO (meandump (0) / 5,3,15) | |
| Dumph | 0.0007 | 1440 | ERLANG (meandump $(7) / 3 3 14$) | PUSH to bufferb7 |
| 7 | 0.0007 | 1.10 | 2 | |
| 垃圾名 | * | 一天的分钟数 | Meandump 为通过各小区人数计算出 | 决定垃圾的存放 |
| 称 | | | 来的变量数组 | |
| | | | | |

*决定每天各个小区的垃圾依次产生,以便于生产叫车次数。

● 运输车辆详细设计

表 11.7 运输车辆详细设计列表

| Name | Unload speed | Load speed | to |
|-----------|--------------|------------|---------------------|
| Dumpcart1 | 1.2 | 1.0 | PUSH to corppark(1) |
| Dumpcart2 | 1.2 | 1.0 | PUSH to corppark(2) |
| 车辆名称 | 空载速度(km/min) | 实载速度 | 生成时,放入公司停车场的车位 |

垃圾箱详细设计

设计垃圾放入垃圾箱时的活动,也即每个垃圾箱的"actions on input"中的程序。 buffera1.actions on input:

IF MOD (NPARTS (buffera1),cartcapacity(1)) = 1

CALL dumpcart1, get1, road7_t, 1



VSEARCH road0_1, corppark, road1_2, road1_0, road2_3, road3_4, road4_5, road5_6, road6_5, road6_7, road7_t, roadt_7, road1_0, road3_6, road6_3, road7_0, road0_7, road2_1, geta1, geta2, geta3, geta4, geta5, geta6, geta7, getb1, getb2, getb3, getb4, getb5, getb6, getb7

ENDIF

moddemanda (1) = MOD (NPARTS (buffera1),cartcapacity(1))

程序解释:

第一行:判断当 bufferal 中的垃圾的数量同车辆 1 的载重量取余为 1 时,发生 if...endif 之间的活动;

第二行: 叫车 dumpcart1, 该车将在路径 get1 上装载垃圾, 在路径 road7_t 上卸载垃圾, 优先级为1;

第三行至第五行:在所有路径上搜索车辆 dumpcart1;

第七行:汇总非整车垃圾的数量,带到 buffera2 中继续计算并叫车。

Buffera2~buffera7的 actions on input:

IF MOD (moddemanda (i-1) + NPARTS (buffera(i)),cartcapacity1) = 1

CALL dumpcart1,geta(i),road7_t,0

VSEARCH road0_1, corppark, road1_2, road1_0, road2_3, road3_4, road4_5, road5_6, road6_5, road6_7, road7_t, roadt_7, road1_0, road3_6, road6_3, road7_0, road0_7, road2_1, geta1, geta2, geta3, geta4, geta5, geta6, geta7, getb1, getb2, getb3, getb4, getb5, getb6, getb7

ENDIF

moddemanda (i) = MOD (moddemanda (I-1) + NPARTS (buffera(i)),cartcapacity1)

i表示本垃圾箱的序号,为2,3,4,5,6,7;

第一行:将其上一个垃圾箱的非整车垃圾数量同本垃圾箱中的垃圾数相加,然 后与车辆一的载重量取余,当结果为1时,叫车。

垃圾箱 bufferb1~bufferb7 的'actions on input '同垃圾箱 buffera1~buffera7 的'actions on input "处理逻辑完全一样,只需要将 buffera 改为 bufferb、dumpcart1 改为 dumpcart2、 moddemanda 改为 moddemandb、 geta 改为 getb 即可。

● 两点间运输路径上的程序设计

下面以 road2_3 为例,加以说明其逻辑流程。其它路径类似,可以在路径的 general detail 中的 output to 中加以查看。

```
IF VEHICLE (road2_3,1) = dumpcart1
```

```
IF NPARTS (buffera3) > 0
```

```
PUSH to geta3(1)
```

ELSE

```
PUSH to road3_4
```

ENDIF

```
ELSEIF VEHICLE (road2_3,1) = dumpcart2
```



```
IF NPARTS (bufferb3) > 0
```

PUSH to getb3

ELSE

PUSH to road3_4

ENDIF

ELSE

Wait

ENDIF

第一行到第六行:决定车辆dumpcart1到达路径road2_3末端时,它的运行路径。 如果此时垃圾箱buffera3中有垃圾,则车辆dumpcart1驶向路径geta3,进行垃圾收集 (第二行到第四行);如果垃圾箱buffera3中没有垃圾,则车辆dumpcart1驶向路径 road3_4。

第七行到第十二行:决定车辆 dumpcart2 到达路径 road2_3 末端时,它的运行路径。如果此时垃圾箱 bufferb3 中有垃圾,则车辆 dumpcart2 驶向路径 getb3,进行垃圾收集(第八行到第十行);如果垃圾箱 bufferb3 中没有垃圾,则车辆 dumpcart2 驶向路径 road3_4。

如果是其他情况,等待(第十三行到第十四行)。

● 路径 get 系列的设置

在 Get 系列(Geta1~geta7, Getb1~getb7)路径中设置车辆的装载(loading)程序和条件,它们的设定以及程序的处理流程基本相同,下面举 geta2 加以说明。其中装载数量(loadnum)和装载时间(loadtime)在 general detail 页的 actions on front 中设定;装载程序和条件在 loading detail 页中进行设定。

General detail 页的 actions on front 程序如下:

IF NFREE (dumpcart1) >= NPARTS (buffera2)

loadnum = NPARTS (buffera2)

ELSE

loadnum = NFREE (dumpcart1)

ENDIF

loadtimea (1) = 3 + loadnum / 1000 * loadindexa / laborneeda

```
dayloadtimea = dayloadtimea + loadtimea (1)
```

程序解释:

第一行至第五行:如果车辆 dumpcart1 的空余容量 NFREE(dumpcart1)不小于垃圾箱 buffera2 中的垃圾数量,则收集数量 loadnum 为垃圾箱 buffera2 中的所有垃圾; 否则,收集数量 loadnum 为车辆的空余容量。

第六行:计算本次收集所需的时间 loadtimea(1),它是本次垃圾收集量、随车工 作人员数以及收集时间系数的函数。



| Detail Track - geta2 | | | × |
|--|------------------------------|--|---|
| General Vnloading Loading | ork Search Actions Rep | orting Notes | |
| Transfer Mode: | Quantity to loadnum | T <u>i</u> me to Load: loadtimea (1) | |
| $\frac{C_{ondition}}{MPARTS (buffers2) > 0}$ | Pull <u>A</u> ctions on Load | × | |
| | | | |

11.2 路径 geta2 中的 loading 详细设计图

路径 geta2 中的 loading detail 页框的设定如图 11.2 所示。

在选中 loading enabled (能够装载)前的复选框后,将会出现该界面上的其他内容。Transfer Mode (装载模式)有三种:if、call、always。我们选择条件模式 if,在条件 condition:框中输入条件表达式 NPARTS(buffera2)>0,即当垃圾箱 buffera2 中的垃圾数量大于零时,能够装载;装载数量等于 Quantity to 框中的变量 loadnum 的值;装载的时间需要 Time to Load 框中的变量 loadtimea(1);装载的源在 Input Loading Rule 规则中进行设定,为"PULL from buffera2",从垃圾箱 buffera2 中收集。

road7_t的设定

车辆每次到达路径 road7_t 的末端时,都要进行卸载处理,所以对路径 road7_t 的 详细设计项目包括卸载所需的时间,卸载模式等。

通过路径 road7_t的 General detail 页的 actions on front 中设定卸载所需的时间,程序 如下:

IF VEHICLE (road7_t,1) = dumpcart1

unloadtime = 3 + 6 * NPARTS (dumpcart1) / 1000 / laborneeda

dayunloadtimea = dayunloadtimea + unloadtime

ELSE

unloadtime = 2 + 6 * NPARTS (dumpcart2) / 1000 / laborneedb

dayunloadtimeb = dayunloadtimeb + unloadtime

ENDIF

程序解释:

第一行至第三行,根据函数 VEHICLE()的结果,如果到达车辆是 dumpcart1,就 通过车辆 dumpcart1 中所装载的垃圾数量 NPARTS (dumpcart1)、车辆 dumpcart1 的 随车工作人员数 laborneeda 来确定卸载时间 unloadtime (第二行),然后统计车辆 dumpcart1 的总的卸载时间 dayunloadtimeb (第三行)。

第四行至第七行,否则,也就是到达的车辆是 dumpcart2,就通过车辆 dumpcart2



中所装载的垃圾数量 NPARTS (dumpcart2)、车辆 dumpcart2 的随车工作人员数 laborneedb 来确定卸载时间 unloadtime(第五行),然后统计车辆 dumpcart2 的总卸载 时间 dayunloadtimeb (第六行)。

图 11.3 是路径 road7 t设置中的另一项内容,就是车辆卸载的相关内容。

在选中 Unloading Enabled (能够卸载)前的复选框后,该页面上的其他选项将 能够被用户设置。卸载模式 (Transfer Mode)也有三种模式 if、call、always,我们 选择模式 always,只要车辆到达路径 road7_t 的末端,就进行卸载活动。卸载数量 Quantity to 设定为 All,就是卸载车辆上的所有垃圾。卸载时间 Time to 设定为变量 unloadtime,决定本次卸载所需要的时间。停车 Park 设定为公司停车场 corppark,决 定在当天回收任务完成后,车辆泊放的位置。卸载到什么地方,将由" output unloading rule"中的程序决定,程序为:push to ship,将所有垃圾送出系统。

| Detail Track - road7_t | | × | |
|----------------------------------|-------------------------------|--------------------------------|-----------|
| General Unloading Loading | Work Search Actions Repor | ting Notes | |
| Vnloading Enabled | | | |
| <u>T</u> ransfer Mode: Always | Quantity to | T <u>i</u> me to unloadtime | |
| | <u>D</u> utput Unloading Rule | | |
| | Push | Park | L |
| | Actions on Unload X | corppark 🔳 | |
| | | | |
| | | | L |
| | | | |
| | | | <u>ام</u> |

11.3 路径卸载 Unloading 页框界面

11.5.5 数据处理子模块

每天车辆完成收集工作完成后,驶向垃圾处理公司停车场停泊时,都将进行一 些数据统计和处理活动。这些程序写在路径 road7_0 的"Actions on Front"中,程序 及其说明如下: IF VEHICLE (road7_0,1) = dumpcart1 IF NDemands (dumpcart1) = 0 runtimea (1) = TIME runtimea (2) = runtimea (2) + 1 runtimea (3) = runtimea (1) - 1000 * (runtimea (2) - 1) timecosta = timecosta + timeweigh * (runtimea (3) - 240)



```
daydistance = DISTANCE (dumpcart1) / 10 - lastdaydistance
         lastdaydistance = DISTANCE (dumpcart1) / 10
         dayunloadtimea = 0
         dayloadtimeb = 0
    ENDIF
ELSEIF VEHICLE (road7 0,1) = dumpcart2
    IF NDemands (dumpcart2) = 0
         runtimeb (1) = TIME
         runtimeb (2) = runtimeb (2) + 1
         runtimeb (3) = runtimeb (1) - 1000 * (runtimeb (2) - 1)
         timecostb = timecostb + timeweigh * (runtimeb (3) - 240)
         daydistancea = DISTANCE (dumpcart2) / 10 - lastdaydistancea
         lastdaydistancea = DISTANCE (dumpcart2) / 10
         dayunloadtimeb = 0
         dayloadtimeb = 0
    ENDIF
ENDIF
FOR num = 1 \text{ TO } 4
    IF VEHICLE (road 7 0,1) = dumpcart1
         roadchoicea (num) = 0
    ELSEIF VEHICLE (road7 0,1) = dumpcart2
         roadchoiceb (num) = 0
    ENDIF
NEXT
程序解释:
```

第一行至第十一行,如果即将停泊的车辆是dumpcart1,同时系统此时对车辆 dumpcart1的需求是零(第三行);先统计车辆dumpcart1在当天收集工作的完成时间, 并判断是否超过了12点,然后计算当天的时间惩罚成本timecost(第三至第六行);统 计车辆的运行距离(第七至第八行);对两个变量置零(第九至第十行)。

第十二行至第二十二行,统计如果即将停泊的车辆是dumpcart2的一些数据,解释同dumpcart1一样。

第二十四行至第三十行,对路径选择变量数组置零。

11.5.6 目标函数 objfun()中的程序

sumcost = 0 !统计总费用的变量置零

sumcost = initialcosta + initialcostb + sumcost

!将车辆的购置成本加到总费用变量中



sumcost = (laborneedb + laborneeda) * 60 * (runtimea (2) + 1) + sumcost

!将工作人员工资添加到总费用变量

sumcost = DISTANCE (dumpcart001) / 10 * permilecosta + DISTANCE (dumpcart002) /

10 * permilecostb + sumcost

!将车辆的运行费用添加到总费用变量

sumcost = timecosta + timecostb + sumcost

!将时间惩罚成本添加到总费用变量

RETURN sumcost !返回变量 sumcost 给目标函数

11.6 仿真运行与结果分析

我们现在采用如表 11.8 所示的两种配置, 来运行该仿真模型, 仿真时间为一年(即 365*1440 分钟)。

| 车辆名称 | 载重量 | 随车人 | 时间惩 | | | | |
|-------------|--|---|---|--|--|--|--|
| | (吨) | 员(人) | 罚因子 | | | | |
| Dumpcart(1) | 3 | 1 | 2 | | | | |
| Dumpcart(2) | 1.5 | 1 | 2 | | | | |
| Dumpcart(1) | 5 | 3 | 2 | | | | |
| Dumpcart(2) | 2.5 | 3 | 2 | | | | |
| | 车辆名称 Dumpcart(1) Dumpcart(2) Dumpcart(1) Dumpcart(2) | 车辆名称 载重量 (吨) Dumpcart(1) 3 Dumpcart(2) 1.5 Dumpcart(1) 5 Dumpcart(2) 2.5 | 车辆名称 载重量 (吨) 随 车 人 员(人) Dumpcart(1) 3 1 Dumpcart(2) 1.5 1 Dumpcart(1) 5 3 Dumpcart(2) 2.5 3 | | | | |

表 11.8 两种配置方案的异样数据表

其它的元素与数值在两个方案中的值完全相同,参见前面的表格。 通过仿真运行后得到统计数据如表 11.9。

表 11.9 垃圾回收仿真系统报表

| | 方案 | 运行 距离 km | 运行 时间 h | 时间 成本 万元 | 收集 时间 h | 卸载 时间 h | 可变 成本 万元 | 固定 成本 万元 | 人员 工资 万元 | 总成本 万元 | 总成本万 元 |
|---|-------------|----------------|---------------|----------------|---------------|---------------|----------------|----------------|----------------|-----------|-----------|
| _ | Dumpcart(1) | 73691 | 2310 | 10.2 | 744 | 407 | 22.1 | 5 | 2.2 | 39.5 | 66.2 |
| | Dumpcart(2) | 74336 | 2263 | 9.6 | 813 | 217 | 11.2 | 3.8 | 2.2 | 26.8 | 00.5 |
| _ | Dumpcart(1) | 54657 | 1366 | -1.1 | 342 | 157 | 27.3 | 7 | 6.6 | 39.8 | 62 |
| | Dumpcart(2) | 54489 | 1292 | -2 | 332 | 86 | 13.6 | 5 | 6.6 | 23.2 | 05 |

为了更清晰的比较两种方案的成本参数,将上表的数据绘制成直方图,如图 11.4 所示。





图 11.4 两种方案的成本参数比较图

从成本数据比较图上可以看出,当采用方案二时,车辆吨位数和随车人员的增加,使得固定成本、人员工资都比方案一要大;同时,回收速度要比方案一快的多, 使得可变成本和居民满意度较高;时间惩罚成本低于零,即取得了很好的市场评价 和公司品牌效应。在这两种方案下,每年方案二的物流成本比方案一要少 3.3 万元。



11.5 两种方案的运行参数比较图

从运行参数比较图上可以看出,两种方案的物流成本差异收集时间、卸载时间、 运行时间、运行距离的差异引起的。由于方案一的车辆吨位和随车人员都比方案二 要少,所以同是收集等量垃圾,它的收集时间和卸载时间都比方案二长,它的收集



次数要比方案二多,即运行距离比方案二长。

