

ED 使用指南

前言

如果你想开车，那么法律上你需要有驾照。这证明了你已经学会了如何控制一辆车并且不会对马路上其他行人或车辆造成严重的威胁。同样地，没有人会指望仅仅通过阅读使用手册就能学会开车。

仿真软件具有广泛的可能用途，就这点而言，没有所谓的驾照。但是为了理解软件和它在仿真问题上的应用，训练仍然是一个前提条件。可以与新车的使用手册作个类比：仅仅翻阅手册对于真正地学会驾驶是不够的。

Incontrol公司推出的Enterprise Dynamics®（ED）是一个用于离散事件仿真的软件程序。该软件有强大的在线支持系统。然而，~~初学者~~尤其是教育领域的初学者——单独使用时需要系统性地按部就班地学习ED的基本原则。

这本使用指南就是为了满足这种需求而写的。使用指南是ED的一个重要部分，它将指引你如何用最基本的技术来创造自己的仿真模型来分析结果。

我们相信，将来ED出现新版本的时候，这本使用指南也会随之调整和扩展。在此，要感谢所有对这本使用指南的撰写做出贡献的人，和使用过ED并对我们改进使用指南和改善系统提出建议的人。

我们欢迎各位读者对此使用指南或对软件提出意见和批评，以便我们及时改进。任何意见或批评请发至Support@EnterpriseDynamics.com。

目录

1	使用指南布局	4
1.1	学习视角	4
1.2	符号	4
1.3	使用指南结构	4
1.4	学习仿真	5
2	ED的背景	6
3	熟悉ED	7
3.1	启动Enterprise Dynamics	7
3.2	窗口部分	9
3.3	菜单结构	11
3.4	库及模型的结构	11
4	模型建立基础	13
4.1	将原子拖进模型	13
4.2	通道	17
5	分析结果	23
5.1	测量结果的方法	26
5.2	测量结果	26
6	发挥策略	42
6.1	调整输入策略	43
6.2	改变Queue Discipline（序列原则）	44
6.3	调整Send to（送到）语句	45
7	更多原子：从集合到解包	47
8	Enterprise Dynamics与Excel®	53
8.1	银行	53
8.2	与Excel连接	55
8.3	在Excel中写入数据	56
8.4	将数据从Excel中读入ED	57
8.5	故障诊断	58
8.6	结合或分开？	59
9	操作员	61
9.1	从操作员开始	61
9.2	在模型的维度内移动	64

9.3	走动(自由运动)	65
9.4	网络的一般概况	67
9.5	在网络上行走	68
9.6	优先权的分配	69
9.7	更多关于操作员的选项	70
10	输送员	71
10.1	输送员的作用	71
10.2	使用输送员	72
10.3	在网络中的(高级)输送员	74
10.4	连接分派员, 高级输送员与指定员	76
10.5	高级输送员的装货及卸货策略	79
10.6	多个输送员的使用	80
附录 1	菜单结构	
附录 2	一些原子的描述	
附录 3	引入4DScript	

1 使用指南布局

1.1 学习视角

这本使用指南将带领初学者学会如何使用Enterprise Dynamics (ED)仿真软件包

第三章开头是打开ED的界面，从此处开始通过逐步逼近的方法，你将逐步了解ED。本书主要通过举例的方式来分析具体应用中的小问题。由此，你将通过模型的建立以及对模型的具体应用了解到ED的功能。

这一方法有几个优点：

1. 用户将通过建立仿真问题的模型来学习ED，这与软件包在实际应用中的目的一致。
2. 与实际应用相结合能更有效地学习一个程序的各项功能。
3. 当遇到一个相似的问题时，用户能通过查阅例子而找到相关解决方案。

简而言之，这本使用指南不仅向你展示软件的用途，还告诉你如何使用以及什么时候使用它。此外，使用指南中的某些部分可做参考，尤其是附录中的概述。

需要弄清楚的是，有了这本使用指南，并不代表着使用手册——可在帮助菜单下找到或按F1——就是多余的。想在构建仿真模型上达到更高层次的用户将会发现使用手册非常实用，因为它提供了所用软件用途的一整套文件管理。

1.2 符号

若涉及子菜单，就会出现 | 的记号。因此File| Preferences (文件| 参数选择)指的是参数选择这一子菜单 (包含一系列标准函数)，这些函数也可在文件菜单下找到。重要的地方都已用**黑体**或*斜体*标出，案例分析和相关的问题也同样用了这类字体。文中的4DScript编码也用了**黑体**或*斜体*。

警告前面会出现“警告！”字样，在很多案例分析中都会出现的重要提示前也会出现“提示”字样。

1.3 使用指南结构

第二章对ED程序作了简要介绍，第三章才开始真正接触ED软件包，首先是对菜单结构的概述和介绍打开ED界面后的各分区。

第四章通过只用一个服务器处理序列问题展示了软件包的基本原则。接下来，为了理解“通道”的概念，增加了两个服务器。在这一实例中应用到的原子是Source（源），Queue（序列），Server（服务器），Sink（接收器）。

第五章主要讲述使用不同方法观察一项研究的结果。我们用了一个木匠工厂的例子，解释了批量的使用。用户也将学习到应用以下几项：Monitor（监控器），Summary Report（总结报告），各种图表和新的Experiment Wizard（实验向导）。另外，我们以初学者的水平进入一个仿真研究，包括：建立模型，验证，实验布局及分析结果。

第六章的重点在于观察产品通过预先设定的方法到达下一个原子（输入策略），被置入一个序列中（序列原则），或使用后重新分配（送到）。

第七章介绍了8个新的原子，包括集合原子和输送机原子。第八章讲述ED与Excel之间的连接，并简单介绍了4DScript和标签。

附录作为参考使用也很重要。附录1简单介绍了菜单结构，附录2对最重要的原子进行了详细描述。附录3作为对4DScript的引入，是ED的编程语言的起点。

1.4 学习仿真

仿真不是一项简单的技术。要明智地使用仿真程序，就需要对离散仿真的理论背景进行深入了解，如分布可能性，模型建立过程，包括认证技术以及实验的设计。虽然使用指南对此也作了阐释，但它毕竟不是关于仿真本身的教科书。若想对离散仿真有更多了解，请查阅现有书籍和讲座材料。

对仿真的研究和模型的建立本身就是一系列实际操作的过程。

我们将会不断更新Enterprise Dynamics，请注意浏览我们的网站。我们也正在计划将这本使用指南的前面部分翻译成多种语言，供大家在网站上下载。

Enterprise Dynamics包含了使用指南中提到的模型的例子，若不能确信你自己的模型是正确的，请参考使用指南中的例子。

2 ED的背景

Enterprise Dynamics是一个以对象为导向的建模、仿真、可视化并控制动态过程的软件程序。用户可从标准库中取用对象—称为原子—来建立自己的模型。ED在建立每个模型时都是以这种概念的原子为基础。

一个原子可以代表一台机器，一个柜台或一个产品，但它也可代表没有物理特征的物质，如图表。就原子的不同类型而言，我们将涉及到基本原子（有5个原子会被经常用到：产品，源，接收器，服务器和序列），输送原子（与输送有关的原子）及实验原子等等。

由于ED的开放式结构，高级用户能建立并使用自己的原子，如建立一个有特殊用途的机器模型。从这一点上来看，ED包含100个标准原子，而且这个数字还在不断增长。对于初学者而言，只需从使用最频繁的大约30个原子中选择足够的材料配合自己的实际应用即可。

因此原子实际上就是未加定义的建模物体，可用来迅速建立模型并展开研究。ED同时有一套内置的编程语言，我们称之为4DScript，可用于在模型中处理现实中的特殊情况。这套语言目前由大约1100字组成，在这本使用指南里我们只对4DScript作简单介绍。

由于ED的开放式结构，用户能自己拓展此软件包或设计自己喜欢的布局。Incontrol公司也提供了以下几种套件：

- 用于生产、原材料处理及分配的物流套件
- 用于机场及运输的机场套件
- 用于教学目的的教育套件，包括物流套件的影印版，包含了许多详细阐述的案例，教师可分配让学生自己学完

这本使用指南可用于物流套件，而它作为一个产品，也是教育套件的一个部分。

Incontrol公司是一个仿真方案的供应商：除了建立、开发、销售ED软件，它还提供计算机仿真领域的扩展服务，如培训和咨询。咨询服务涉及开展仿真研究及应用软件建设。

3 熟悉ED

3.1 启动Enterprise Dynamics

通过开始菜单启动Enterprise Dynamics，首先出现（图3-1）的闪屏，在此，你可能需要选择一个应用来开启（图3-2）。在这里选择一个文件名以.app结尾的文件。

Enterprise Dynamics经常会选择应用文件，这样，用户将不能看到选择应用的窗口！

Enterprise Dynamics的作用可在应用文件中找到。也可对ED的作用进行编程以决定用户首先看到哪个菜单以及哪些原子能直接使用。用户能自行调整应用文件。

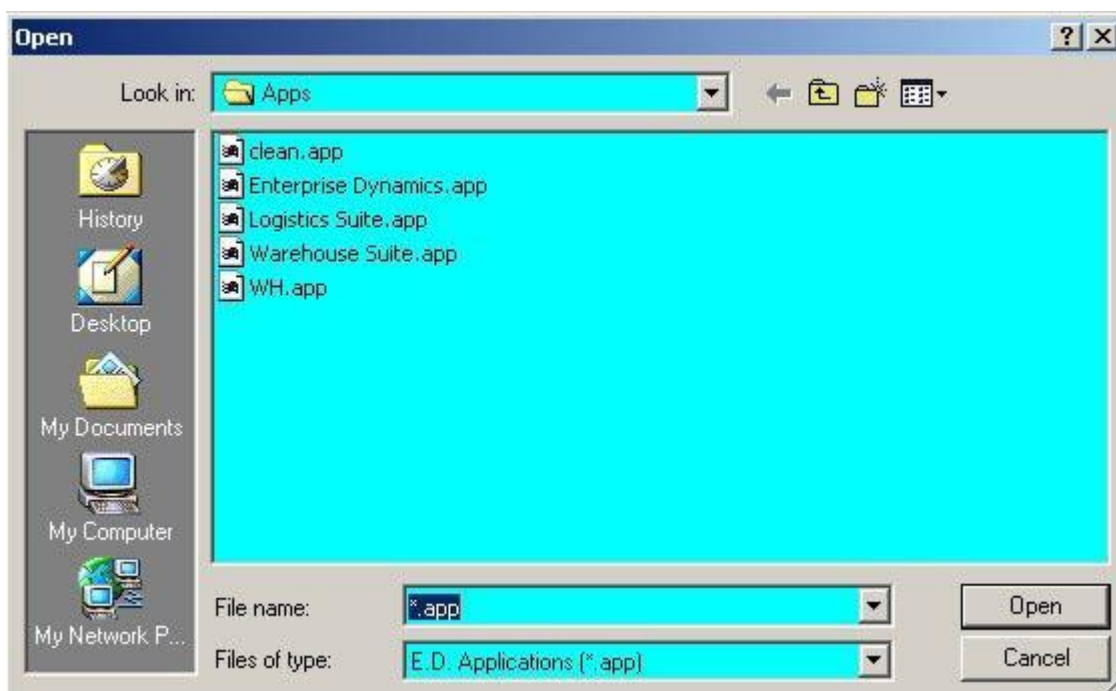


图 3-1: 应用软件选择

3.2 窗口部分

当Enterprise Dynamics完全启动时，出现的窗口近似于图3-3。

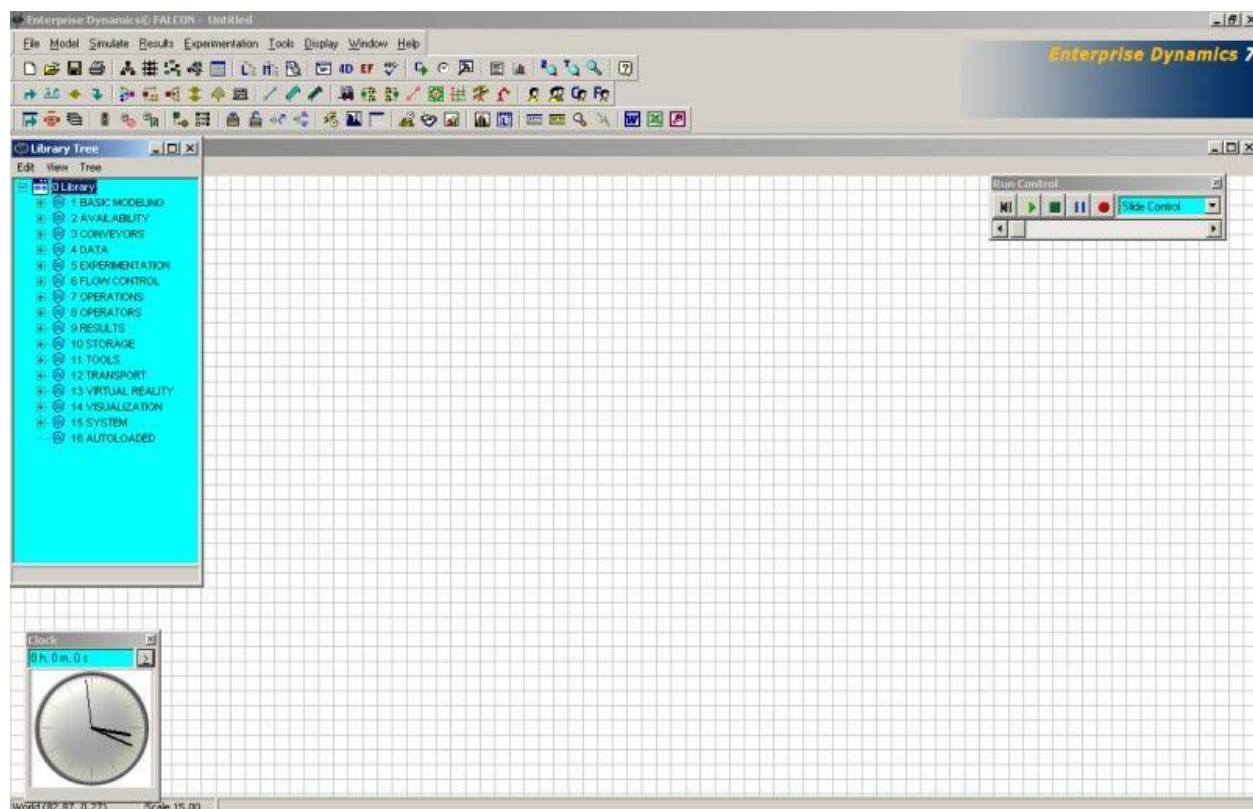


图 3-2: 启动Enterprise Dynamics后的窗口分布

窗口被分为以下几个部分：

- **菜单栏：**用于打开和保存文件。在3.3中将会更全面地涉及到菜单结构。
- **快捷按钮：**通过这几个快捷按钮，具体的原子就能被拖进模型中，因此能对Enterprise Dynamics进行命令（如保存一个模型）。点击快捷按钮时，原子会自动出现在模型中，或者一个命令被执行。
- **库：**库中包含用户能置入模型中的所有原子。每个原子有特定的功能，将不同原子正确地绑定在一起，可以重新创造Enterprise Dynamics的业务流程（‘模型’）。模型建立将在第四章中做详细描述。
- **模型分布窗口：**模型在这里被建立。

- *运行管理*: 在此可重新设置参数，启动模型，控制执行速度。
- *时钟* 在仿真过程中显示模型所需的模拟时间（不是真实时间！）

3.3 菜单结构

菜单的功能和外观与Windows其他的应用软件如Word和Excel中的一样。最常用的菜单选项在以下表格中作进一步解释。

主菜单显示在菜单栏中，可被分为以下几个子菜单：

文件	新建，打开或保存文件，或控制标准功能如打印或另存为。
模型	创造、观察模型。
仿真	实际运行 <i>单个</i> 仿真过程。
结果	产生 <i>单个</i> 仿真运行报告和图表。
实验	通过 <i>多个</i> 仿真运行对实验进行设计、执行和评估。
工具	包含建立原子的工具如各种编辑器。
显示	管理2D（二维）或3D（三维）模型的可视化。
窗口	包括各种窗口，如4DScript或图标的概述。
帮助	包括全套指南以及公司信息、版本信息

注意模型选项、仿真、结果按仿真研究中的步骤依次出现！

在每个主菜单项下都隐藏着子菜单。附录1中对这一 ~~结构及其~~ 包含的每个子菜单进行了阐释。以 *斜体* 显示的菜单选项只对高级用户有用，初学者可直接跳过。在附录中别有用心地添加了整体概述，因为使用指南的这一部分也常被用来作为参考书。通过阅读附录1可对程序的结构有个大概了解。若想在以后独自创造模型，那么对这一结构的知识将会显得非常有用。

3.4 库及模型的结构

在ED中我们用一个树型结构对原子的构造进行可视化分析。我们用它来表明哪些原子包含哪些另外的原子。如主树展示了应用、库及开放模型的全图。

另外两棵重要的树分别是：

- 库树（见图3-3），这里列有用户能置入模型中的所有原子。原子共分为两组，如输送组和操作组。通过选择一个原子并将该原子拖进模型窗口中（‘模型版面’），就将该原子添加到模型中了。
- 模型树，这里列有模型中将会用到的所有原子。

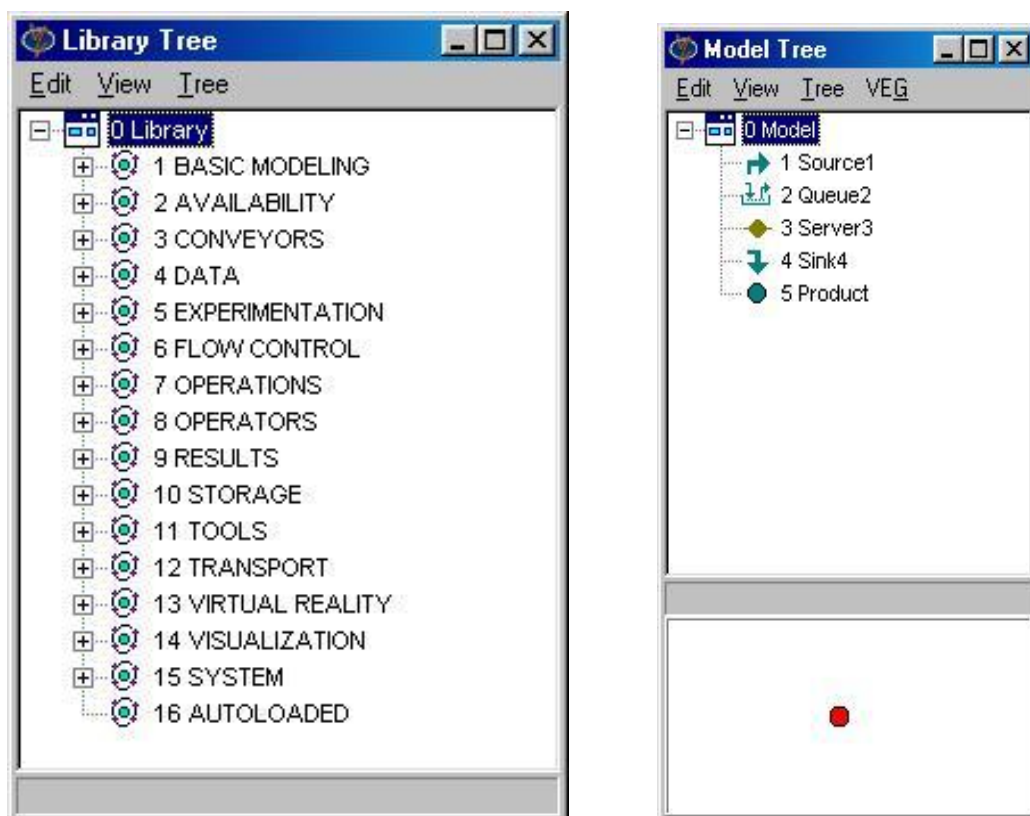


图 3-3: 例1中的库树和模型树

通过在菜单选项中选择库树或在模型菜单中选择模型树，库中的两个部分可互换。此操作也可通过快捷按钮完成。图3-4 显示此功能的相关快捷按钮。



图 3-4: 库和模型树的快捷按钮

4 模型建立基础

前一章只讨论了Enterprise Dynamics的理论部分，这一章我们将开始在Enterprise Dynamics中建立简单的模型。这一章的目标仍然是学习ED，而不是完整的完成一项仿真研究。

在本章结尾，用户应能建立起一个能同时使用几台机器的模型。

4.1 将原子拖进模型中

在模型创造中第一步是将正确的原子放置到模型中。在这一部分中，我们将开始建立一个简单的模型，包含以下4个部分（见图4-1）：

- **Source（源）：** 这个原子的作用是在模型中产生产品
- **Queue（序列）：** 这个原子是顾客和产品等待的区栏
- **Server（服务器）：** 这个原子相当于一台机器或一个计数器，其他原子进入服务器进行处理，并在服务器中停留一段时间（处理时间）
- **Sink（接收器）：** 产品和顾客通过这个原子离开模型



图 4-1: 快捷按钮：源，序列，服务器，接收器

例 1

每小时平均有20位顾客来到邮局，工作人员平均只有2分钟时间来帮助每位顾客处理事件。当然，每个小时内顾客人数会有变化。

平均分给每位顾客的2分钟时间也会有变化。只需买邮票的顾客所需的时间少，而要开一个账户的顾客所需的时间就多。顾客都是以先进先出的顺序排队等候。

有些顾客抱怨排队的问题，由于邮局的经理比较关心顾客接受到的服务，于是他就想对这个问题进行一番调查。

问题及任务

1. 邮局工作人员的办事效率多高？那意味着什么？
2. 你预计平均序列长度是多少？
3. 你认为邮局的经理有什么样的性格特征？

利用仿真这一工具，我们能知道平均序列长度，因为这是我们需要建立的第一个模型，我们将一步步进入这个模型。

因此我们将运用在这部分开头提到的4个原子，并按上文提到的顺序将它们放入模型中，第一个是**源**，接着是**序列**，然后是**服务器**，最后是**接收器**。

通过点击快捷按钮（见图4-1）或将原子从库中拖出可对原子进行选择。若是后者，可点击库窗口中“1 BASIC MODELING（基本模型建立）”中的“+”。

点开后将你在屏幕上看到如图4-2:

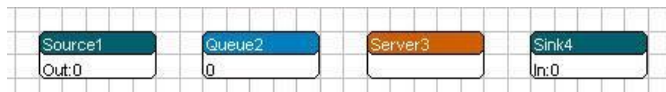


图 4-2: 第一个模型

若出现错误，可点击相关的原子并按下“delete”键将原子从模型中删除。

为了检测顾客是否按正确的顺序经过模型，我们将通过运行管理窗口开启仿真过程。如果这个窗口不能显示，则选择菜单选项中的仿真选项。在运行管理窗口中，需选择子模块幻灯片控制（见图4-3）。这个模块能调整仿真过程的速度。



图 4-3: 选择幻灯片控制

开始仿真之前，先点击重启键，即运行管理窗口的最左边的按钮。点击重启键后，一个蓝色的圆点将会出现在源原子的旁边。这是一个产品原子。通过点击开启键（绿色的三角形），产品，即此例中的顾客，会经过其他原子，即此例中的邮局。若有必要，可拖动运行管理窗口中的速度按钮来调整速度！

在这个模型中柜台（服务器）前应出现一个序列。因此蓝色圆点只在柜台（服务器）中可见。另外，服务器中会显示百分比，反映到目前为止服务器的利用率。若百分比无法显示，你可能需要同时点击鼠标左键和右键并将鼠标前移或后移来放大或缩小屏幕。你也可接通2D Visual Trace（二维目视追踪）（Display Option中的一个子菜单）来更好地观察产品原子的移动。

虽然原子是按顺序放置的，但是我们仍需输入每小时的顾客流量以及每位顾客所需的时间。在Enterprise Dynamics中输入周期时间前，首先要知道Enterprise Dynamics中的*所有周期时间都以秒为单位*。若工作人员接待一位顾客的时间是两分钟，你在Enterprise Dynamics中就需输入120秒。同样的，

我们得用秒来计算中间到达时间，即前后顾客之间的交叉时间。

首先，我们改变源的参数，使得每小时到达邮局的平均顾客人数为20。右键点击（或双击）源原子就会出现一个输入窗口（见图4-4）。打开这个窗口通常会出现“General”选择框。前后顾客之间的交叉时间可输入中间到达时间栏。我们用分布可能性来建立一个到达过程中不确定性的模型。这个例子中使用了负指数分布，这也是经常在不可预测的到达或服务过程中经常被使用的可能性分布。在Enterprise Dynamics中，会用到以下的4DScript代码：negexp(e1)，e1代表指数分布的平均值或期望值。现在在inter-arrival time中输入180，点击OK确认。

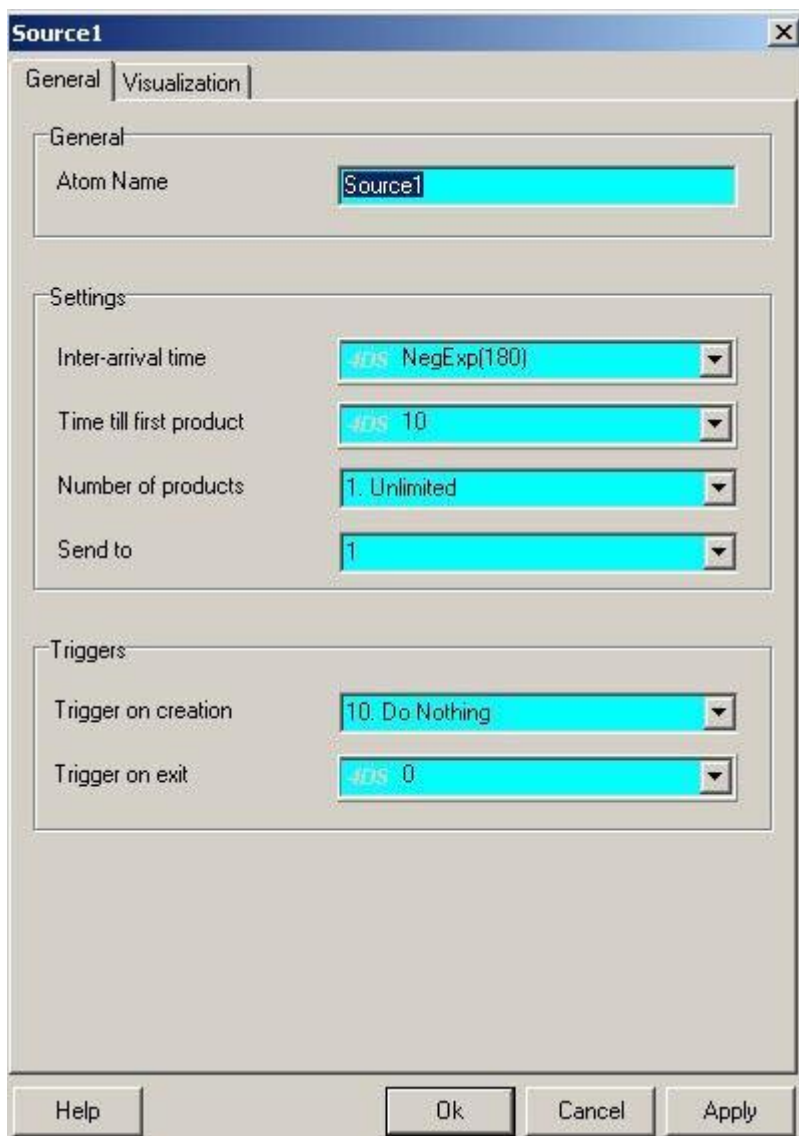


图 4-4: Source的输入窗口

重要:

在输入栏的右侧有一个向下指的三角，双击这个三角将会打开一张列表，上面有可用的并且未命名的选项。首先查看一遍上面是否有对你有用的选项！附录2对这些选项进行了解释，其中对最重要的原子及它们的输入栏也进行了完整的描述。“4DS”代表能输入4DScript命令，即ED的编程语言。

接下来，每位顾客所需的时间也需要输进去。此例中也用到了负指数分布。现在可通过右键点击或双击服务器原子在“General”选择框上调整服务时间（见图4-5）。选择“(negexp(10))”并点击（将出现4DScript窗口）。通过这种方式调整周期时间，以使每位顾客所需平均时间为两分钟。

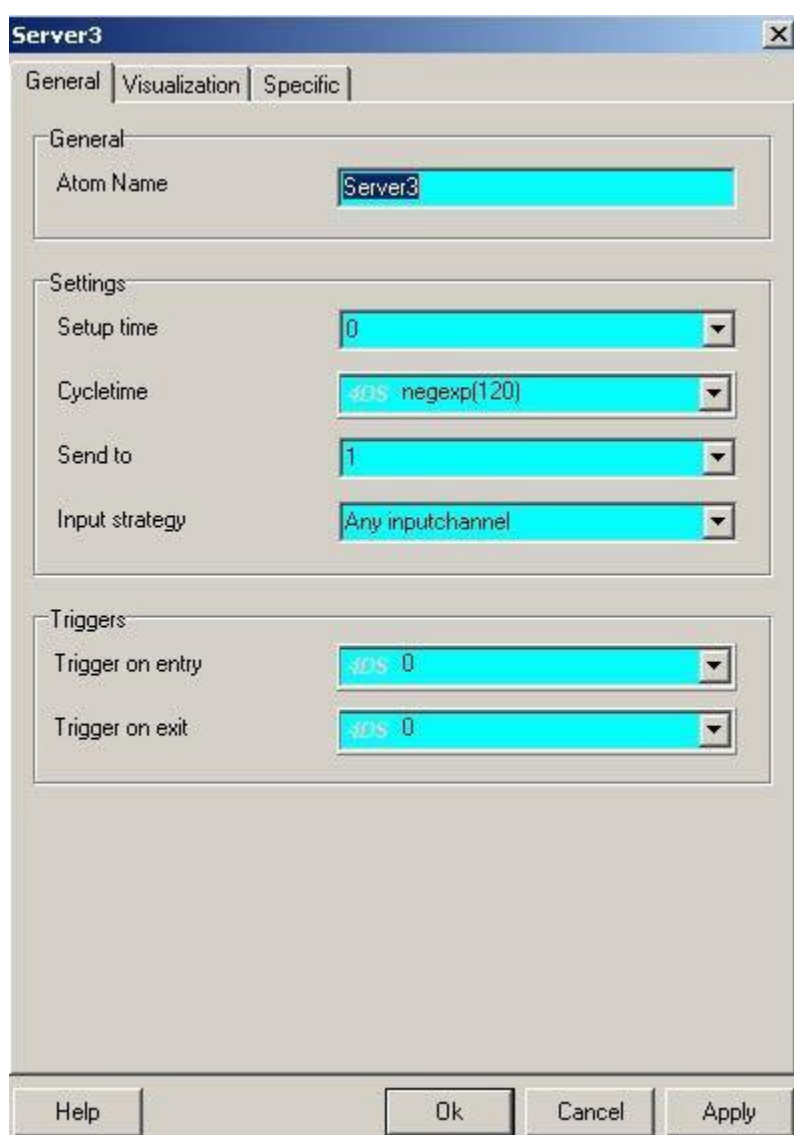


图 4-5: Server的输入窗口

通过运行管理对仿真过程进行重新设置并重启。由于我们已增加前后顾客之间的交叉时间，最好加快仿真过程的速度。充分放大即可看到排队的顾客人数。

你可将蓝色圆点用另一个图标来代替，用来代表我们所举例子中的一位顾客。可通过双击（或右键点击）蓝色圆点使它位于源的左边。在二维图标输入栏内，可先选择“Visualization”选择框，然后双击蓝色圆点，可出现Resource Manager，用户可从中选择（某些）图标。点击你想要选择的图标，然后在Resource Manager中点击OK，产品输入窗口中的蓝色圆点即被你的新图标所代替。点击OK确认所作更改。

Enterprise Dynamics在三维显示中的图标也同样能更改。点击3D Icon选项，从列表中选择Person图标。可显示二维及三维中的模型。打开三维可视化：在主菜单选项中选择Display（显示），然后选择三维中的可能性。要想导航三维模型，可移动鼠标和点击鼠标。

提示：可供选择的图标很多。可通过File | Import将现有的图标添加到标准列表中。

在这一例中，选择一个不同图标只是为了改善显示效果。然而，若要区别不同类型的顾客时，多个图标的使用能增加清晰度。例如，用绿色图标代表一组顾客，而用蓝色图标代表另一组顾客。

点击File|Save可将模型保存。为了可供用户参考，这个邮局的模型可在使用指南所有模型中Postoffice1.mod的文件名下找到。

4.2 通道

（续）例 1

邻近地区已有两家邮局关闭，经理希望他的邮局能有很多新顾客，甚至期望在不久的将来顾客人数能达到原来的两倍。那么，每小时到他的邮局的平均顾客人数为40人。为了计算出邮局里现有的柜台能否处理增加的客流量，经理决定实施一个新的仿真研究。

4. 实施这一情景的仿真有意义吗？为了回答这个问题，可用预期利用值，并通过更改postoffice1.mod的模型使其应用于新情景，并运行仿真来检测你的假定。

（续）例 1

看起来一个柜台无法应对增加的客流量，为了保险起见，经理决定开放3个柜台。于是，他又向看看这一情景对序列的影响。

首先，我们将新增的柜台置于现有的柜台下（选择正确的原子来建立柜台的模型）。然后，我们需要更改新增柜台的周期时间的参数，使其与第一个柜台的参数一致。你可能需要翻回4.1章节，因为现在的模型近似于图4-6。

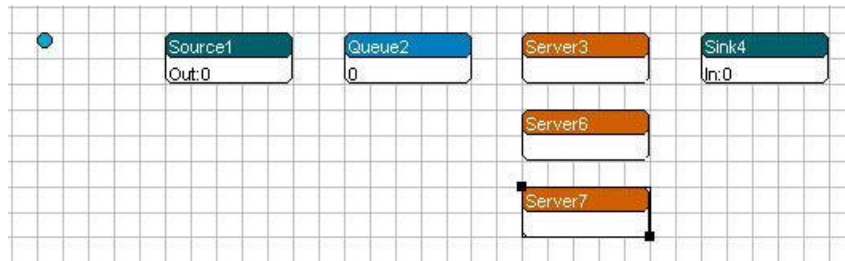


图 4-6: 有三个柜台的邮局

提示: 原子可以被复制—包括所有的输入栏和参数-! 首先, 点击要复制的原子, 按F6, 然后那个原子的副本就产生了。若要复制多个原子, 可用鼠标右键选择这些原子所在区栏, 或按下Ctrl键逐个选择自己要复制的原子, 然后按F6或Ctrl-V即可。注意: 若选择Ctrl-V, 副本将会直接出现在原来的原子下面, 因此你需要重新安置副本。若重新安置多个副本, 你也可结合以上提到的选择程序和方向键。

然而, 启动仿真后, 你会发现顾客仍然被送至第一个柜台。为了纠正这一点, 我们得使用Enterprise Dynamics中的引导顾客的方法。

在Model Layout窗口下的View下拉菜单中, 选择Channels (见图 4-7)。这样, 原子的通道就出现了。ED中这些通道的作用是沿着这一路径发送产品并传递信息, 因此, 这一章节将重点讲到。

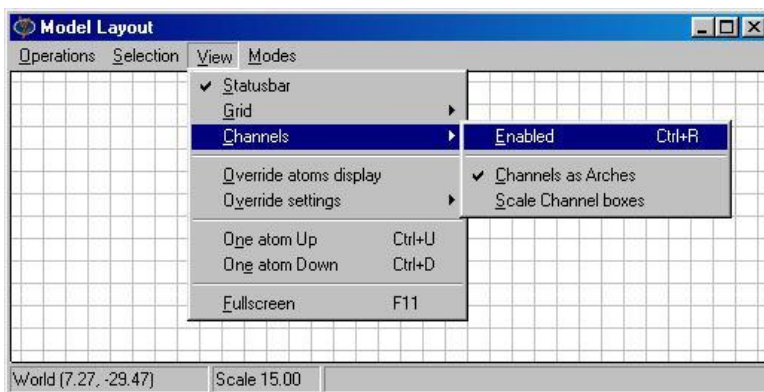


图 4-7: 打开通道

打开可见通道的选项 (见图 4-8)。

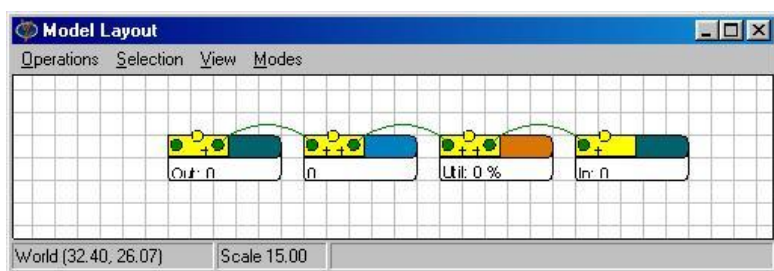


图 4-8: 打开通道的邮局的原始模型

若两个通道之间的线路是直角的而不是弯曲的，那么就可能需要打开View菜单下的 Channels as Arches 选项(仍见图 4-7).

通道打开后，一个方块被添加到所有原子左上角。这个方块包含了相关原子的输入和输出通道。(见 图 4-9)

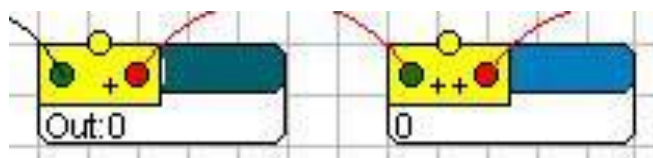


图 4-9: 已连接的通道

这个方块中含有几个圆点，左边的圆点是输入通道，右边的是输出通道。通道可以打开（呈绿色）也可以关闭（呈红色）。若输入和输出通道都是打开的，那么输入和输出之间已连接（呈绿色），否则，通道还未连接（呈红色）。

中间的圆点（黄色小方块上方）用来接收（而不是发送！）信息。在指南中，这一通道被称为Central Channel（中心通道），用来记录与原子有关的信息。一个原子内只有一个中心通道，但是几个原子能被连接到同一个中心通道。

产品通过输入通道进入一个原子，并通过输出通道离开这一原子。一个输入通道通常需要连接到一个输出通道或一个中心通道，而一个输出通道也通常需要连接到一个输入或中心通道。每个输入或输出通道只能被连接到其它通道上。

通过点击圆点旁边的“+”号，输入或输出通道的数量可增加。用鼠标右键点击通道圆点，则可纵览原子的所有连接。输入或输出通道的数量达到两个以上时出现“-”号的作用就不明而喻了。

重要: 虽然通道有很多用途，但是通道的基本功能是（通过读取：序列，柜台）暗示这个原子中的产品的下一个可能去处。因此，通道提供了模型中产品的路径。

图 4-10 代表了一个原子的4个输入通道，6个输出通道及中心通道。

请解释为什么在选项2或选项3中，要么是第一台服务器，要么是最后一台服务器的使用率最高，而选项4中服务器的使用率在运行仿真一段时间后却没什么变化！

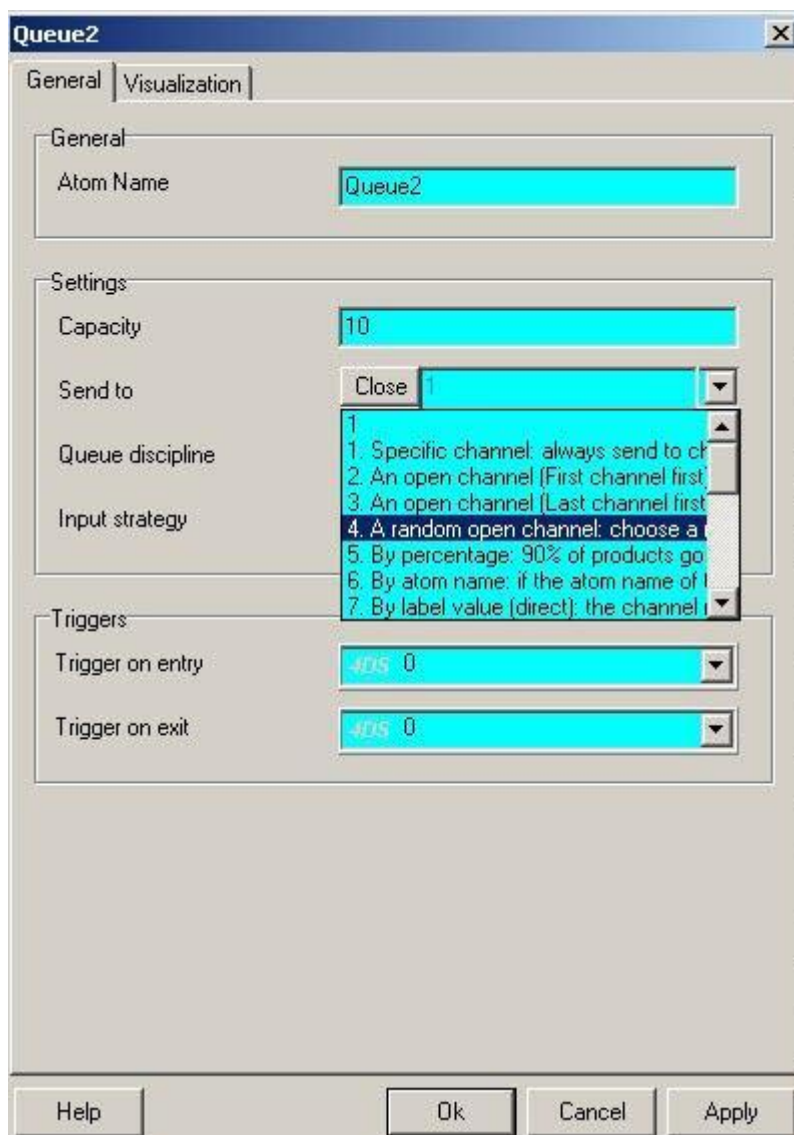


图 4-12: Queue: Send To （序列：送到）

在Send To（送到）栏中，你可以输入一个数字—通过写一段4DScript编码产生输出通道号码，你也可以用预定义编码。对初学者而言，无疑后者是最简单的方法。

点击输入栏中向下的小三角会出现一张列表，用户可从中选择选项。选择一条明确的界线后，就能对蓝色字体显示的文本进行修改。显然地，只有还有待于修改的可能时，这项才适用。

现在选择一个策略使三个柜台的使用率相同。若柜台的使用率不能直接显示，可使用鼠标的左键和右键对屏幕进行放大或缩小。选择Display | 3D Model View可同样观察三维中的模型！

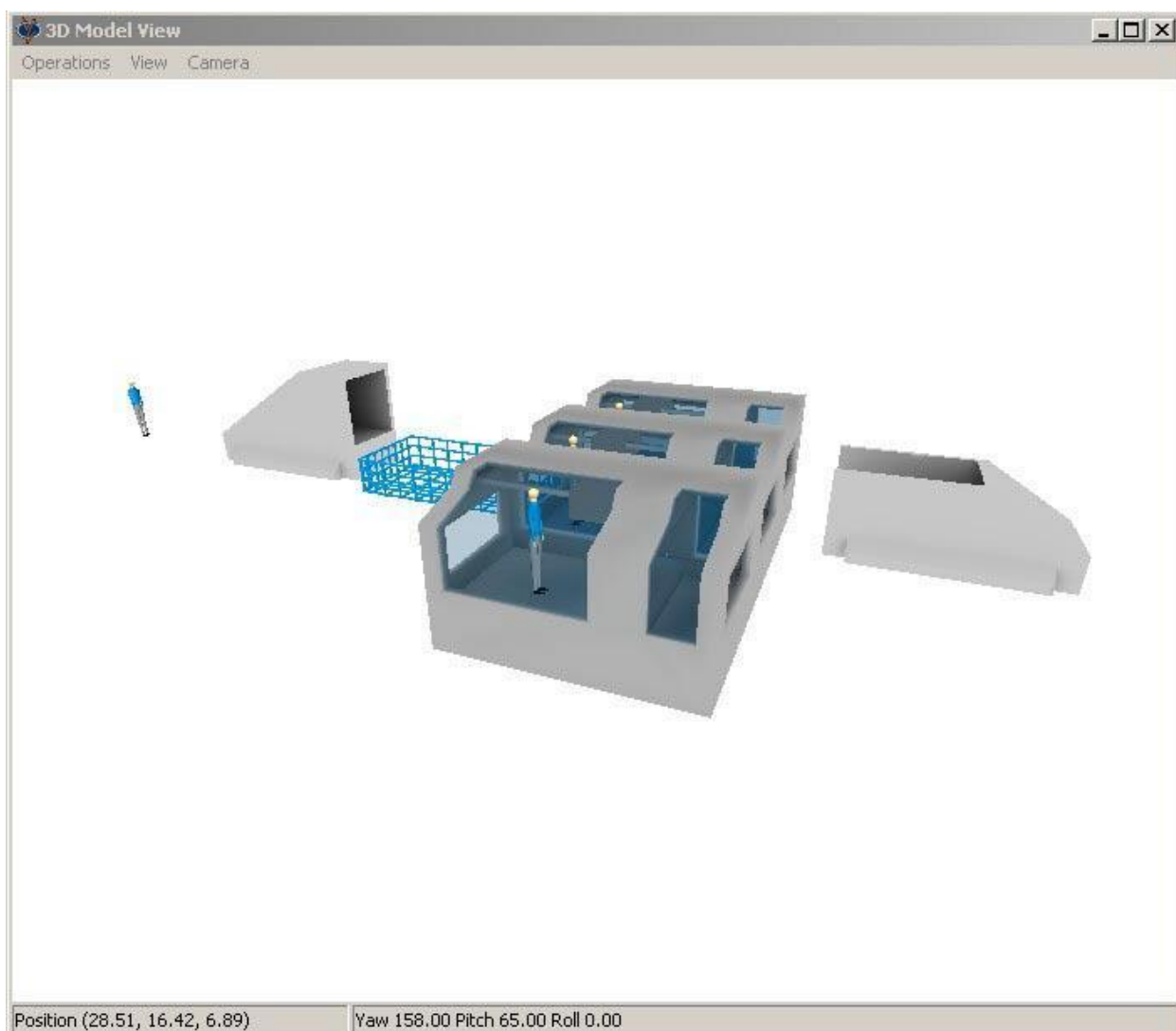


图 4-13: 将三个柜台模拟成机器的三维模型

5. 三个柜台的布局足够了吗?

现在可以将模型保存起来了, 为了参考使用, 这个邮局的模型可在指南模型中在 **Postoffice2.mod** 的文件名下找到。

5 分析结果

在上一章中我们用具体的、逐步的方法模拟了一个邮局。从中你已经知道如何将原子放置到模型中，并在原子之间建立起通道。你也知道了如何在一定的时间间隔内将流动的产品原子置入模型中，以及使它们在规定的处理时间内停留在机器（柜台）中。也就是说，你已经掌握了建立模型的基础的原则。

在这一章中，我们将要建立一个更复杂的模型，尤其是要观察怎样跟踪并测量仿真研究的结果。同样，我们从一个（简单的）例子开始，（建立好模型后），解释测量结果的不同方法。然后我们将在模型中反复练习。

例 2 木匠的工厂

基本背景:

一般情况下，木匠工厂内部有自己的设计部门，这个部门很难管理，因为每个命令都是一个单独的方案，每个生产步骤的处理时间都必须计算地非常准确。这样的环境下通常会出现大量中间库存，冗长的生产时间和可变的瓶颈问题。

木匠工厂主要生产窗和窗框，管理层想对瓶颈、生产数量及生产时间等问题进行仔细研究。因此，他们决定实施一个仿真研究。为了简化仿真研究，这一例中只处理窗的生产。

窗的生产过程可分为以下几个步骤。首先，大型木材被送进来后，用锯子锯成10小段。然后这些小段木材根据要求经过铣刨机又被切割成一定的图样。经铣刨机铣过后，这些就被送往台虎钳。

在这个步骤中，每次会有四段小木材被放进框架中并用胶水粘好。因为生产过程中有平行的两台铣刨机和两台台虎钳，因此小段木材只需一次就能铣好并粘好。在所有相继的生产步骤中都有存储空间，原材料的供应也不会间断。为了控制库存量，每个中间存储空间不得超过100件小段木材。

将运送进来的木材锯成小段木材的时间统一在2-3分钟，在铣刨机上的处理时间也通常控制在平均36秒，有2秒的标准偏差时间。

台虎钳将一个窗粘好的时间恰好为两分钟。生产过程从09:00一直持续到17:00，中间不间断。第一天未能处理好的产品将留置第二天再进行处理。

预先分析

1. 画一幅生产过程图，表明每小时各生产阶段——原始木材，铣过以后的小段木材及窗——的生产产量。
2. 工厂每天能生产多少窗框？

现在开始建立Enterprise Dynamics的模型。分布可能性可在周期时间的下拉菜单中找到。注意，对于正态分布，公式 $\max(0, \text{normal}(36, 2))$ 是用来防止出现负的服务时间。使用批规则1进入，在服务器（标签单“tabsheet”（标签单））上用批数量为10的规则B处理，这时便能按1:10 的比例将木材截短，使用批规则B进入，并选择批数量为4的规则1处理，那么便能按4:1 的比例将短的木材还原为框架。更改服务器的名字可使模型更容易解读。

如果不出现错误，你的模型应近似图5-1中的模型。

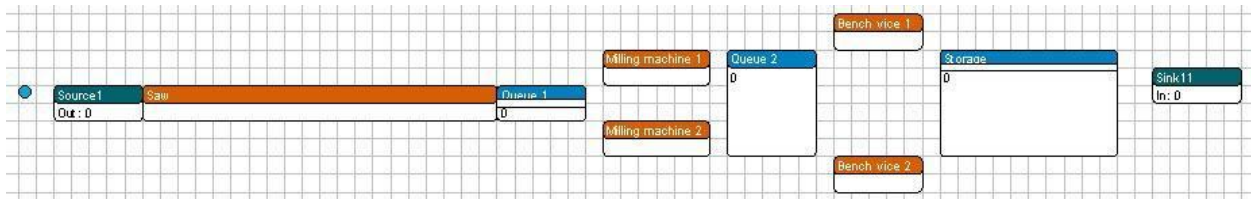
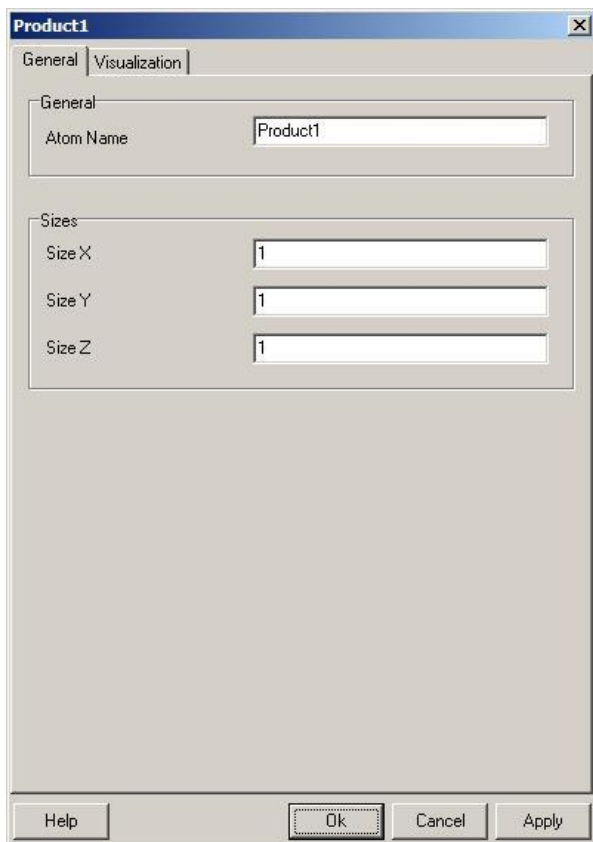


图 5-1: 木匠工厂的分布

通过点击原子即可更改原子的大小，然后可将原子从左上角或右下角拖到你想要的位置上。

产品原子（蓝色圆点）可被改成一个褐色的类似木材的图标：只需双击或右键点击产品并将输入栏改成如下图。



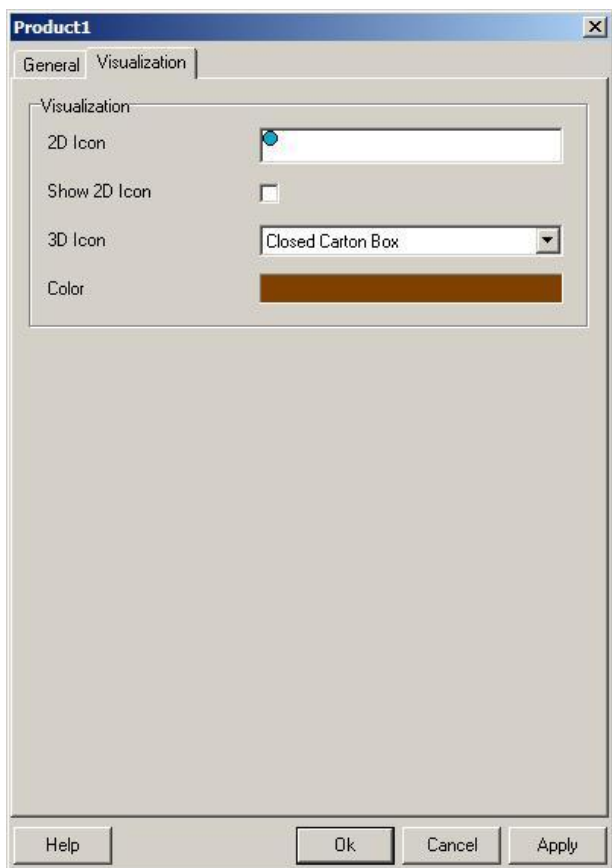


图 5-2: 改变产品的显示

保存你所做的更改。看看出现了什么变化？

蓝色原点的二维图标仍然存在，但是消除*Show 2D icon* 后只剩褐色的背景。它长0.2米，宽1米，近似笔直的木材。在三维中，0.2米的高度（z轴）有另外的作用。注意，产品的图标在二维和三维中不一样。在三维中，可选择如托台或其它的图标。

在二维中，原子后面的可以看见的格子可视为坐标，每个正方形的大小都是1 x 1米。可以看见的黑色的正方形即位于坐标(0,0)。物理维度在这个抽象的木匠工厂中起不到任何作用，但是在其它应用中，如有较短步行距离的仓库，或得行使一定距离的叉车，有很重要的作用！

如果你不能确认自己的模型是正确的，你也可参照下个章节中提供的文件名为 **timber1.mod** 的模型。

5.1 测量结果的方法

既然我们现在已经有一个可利用的模型了，我们就能开始测量结果了。在Enterprise Dynamics中，有几种方法可供我们选用。

1. 原子显示的信息

每个Queue（对列）会显示对列中的产品数量，每个Server（服务器）会显示它的利用率，每个Source（源）或Sink（接收器）显示了多少产品已进入或已离开模型。这些信息对于在仿真过程中查看模型是否有条不紊地工作是极其有用的（这也是模型认证的一部分）。

2. Monitor（监控器）

Monitor（监控器）能给出一个具体原子的图形信息。用户可选择需要查看的信息。此信息对于查看在仿真过程中模型是否有条不紊地工作也是很有用的。

3. Results菜单中的Summary Report（总结报告）和Graphs（图表）

仿真运行时的中间结果可在这里显示。这个方法对于快速浏览系统状态及改变各种参数后的效果的快速反馈是很方便的。

4. 实验

5. 这种方法与前三种都不一样，它只用于实际研究中。测量周期的长度及模拟这一周期的频率，例如半年测量10次，都是预先设好的。我们也必须明确我们想要测量的变量。实验结束时，变量的平均值是在可信限内给出的。

5.2 测量结果

在5.1章节，我们已介绍了四种能够显示仿真研究结果的方法。在这一小节中，我们将把这四种方法应用到木匠工厂中。所以，必须在Enterprise Dynamics中打开木匠工厂的模型。

1. 原子显示的信息

Enterprise Dynamics中的时钟必须显示在屏幕上（Simulate的子菜单）。开启仿真，看在8小时内工厂生产了多少产品。通过选项Simulate | Set停止时间，Enterprise Dynamics能在8小时后准确地停止仿真周期。

3. 可重复实施这一步骤，记录生产的窗的数量。这与前面所做的分析结果是否一致？这一过程中的瓶颈是什么？

2. Monitor（监控器）

监控器的使用很简单。需打开库树The Library tree (不是模型树Model Tree)。在Results组中找到监控器原子，并将监控器原子拖到模型中。如果通道已打开，原子应近似于图5-3。

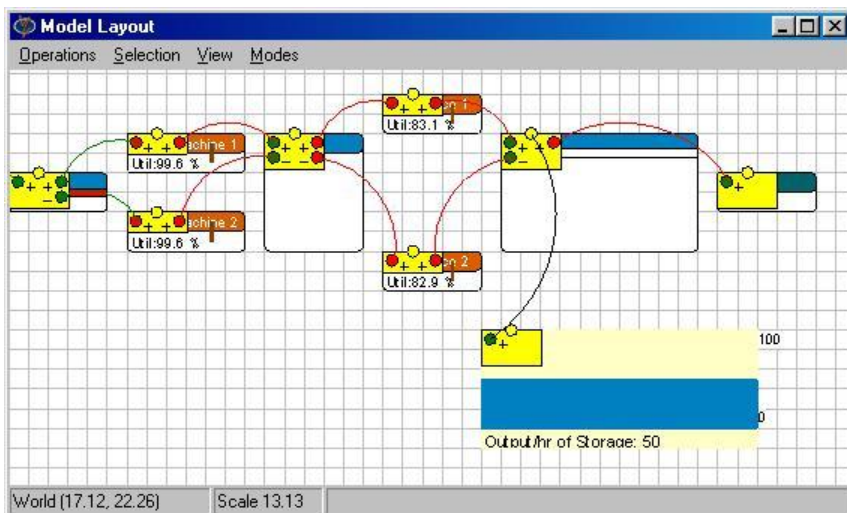


图 5-3: 有（已连接的）监控器原子的部分模型

第一步是将监控器与你想观察的原子连接起来。此例中，我们想要知道每小时工厂生产的窗的数量，那么，监控器就需与Sink（接收器）前面的Queue（序列）相连接。将监控器的输入通道与序列的中心通道（即信息通道）相连接。你也可双击或右键点击监控器原子，然后选择将要观察的原子。

第二步，你需要输入下一步将要观察什么。这也可通过双击或右击监控器原子并选择Monitor Variable（监控器变量）中的选项来完成。如图5-4的列表将会出现。在列表中选择“Output per hour”选项。

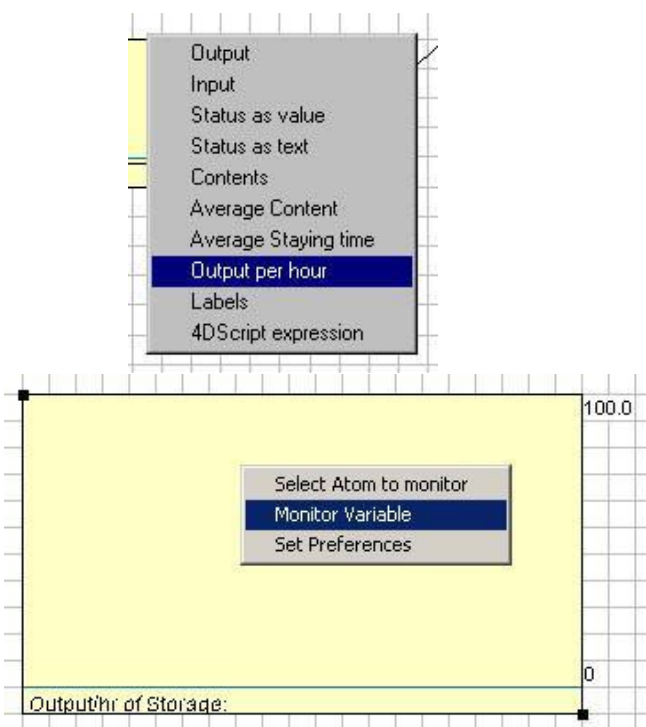
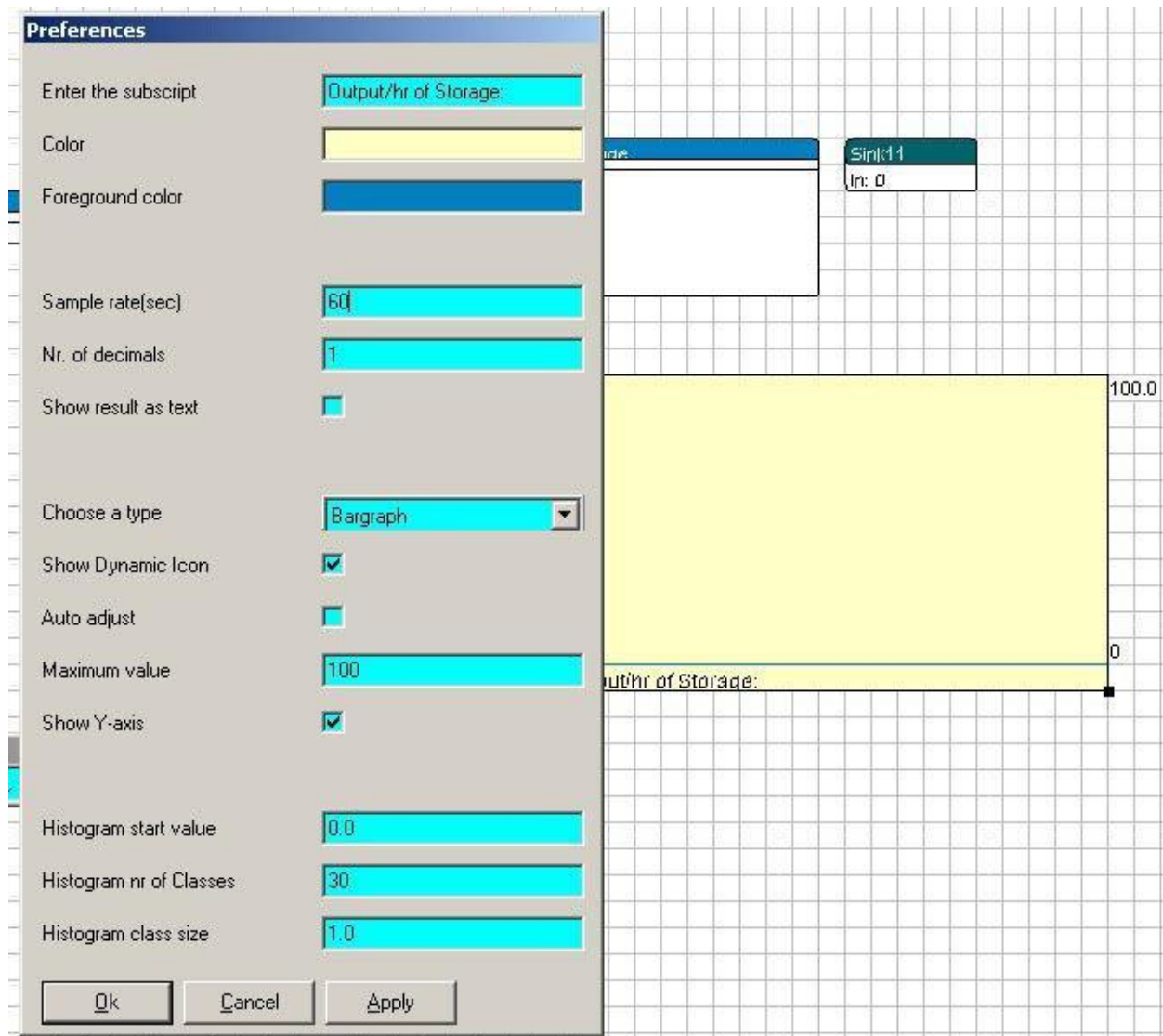


图 5-4: Monitor Variable（监控器变量）

如果所要求的变量没有出现在列表中，则需输入4DScript。可通过选择4DScript Expression来完成，然而，对于初学者而言，我们并不推荐这种方法。

现在你可以看到为什么我们要在Sink（接收器）之前添加Queue（序列）：目的是为了使用预定义陈述中的每小时输出量（而Sink没有输出）。

第三步，监控器的其它参数也可调整。双击监控器，选择Set Preferences选项，如图5-5的窗口将会出现：



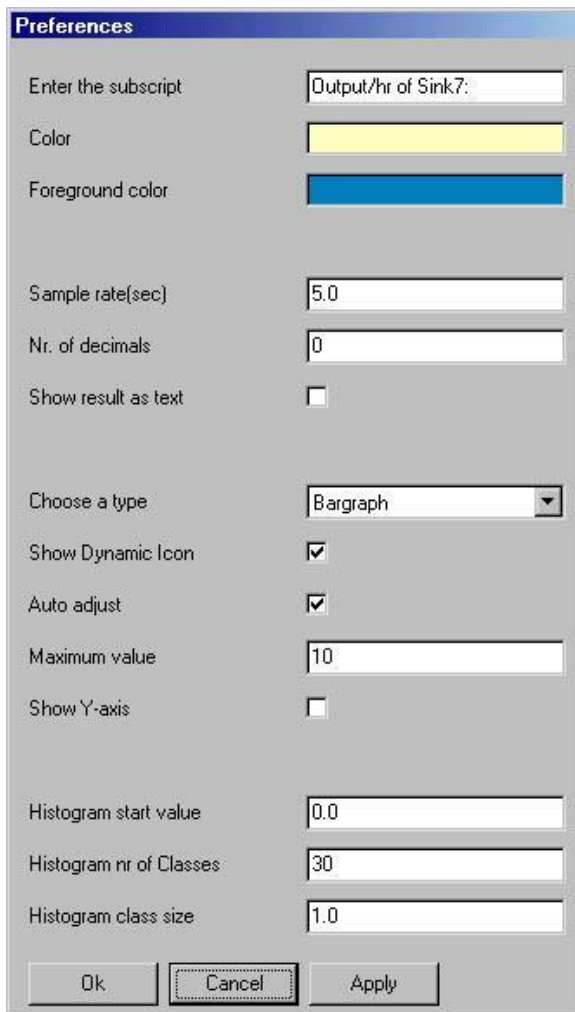


图 5-5: Monitor preferences (监控器参数选择)

对样本率的标准设置是5秒，将它改成60秒，那么（模拟中）每分钟将进行一次测量。由于平均输出不可能仅限于整数，所以把小数都调整为1。为了感受不同的设置，你也可以在*Auto adjust*, *Maximum value*, and *Show Y-axis* 中任意设置！

我们将100设为最大值，如图5-3。现在可启动仿真。

提示: 若使用很多监控器原子，并将样本率设为默认值5秒，将导致模型运行速度非常慢！记住监控器原子模拟中每5秒即显示，这会占用很多处理时间！这种现象在即使模型版面窗口已关闭的情况下也会发生。一般来讲，样本率最好设为60秒或者甚至是3600秒。所以，在监控器原子中使用样本率默认值5秒时要注意。

4. 平均每小时工厂生产多少窗户？

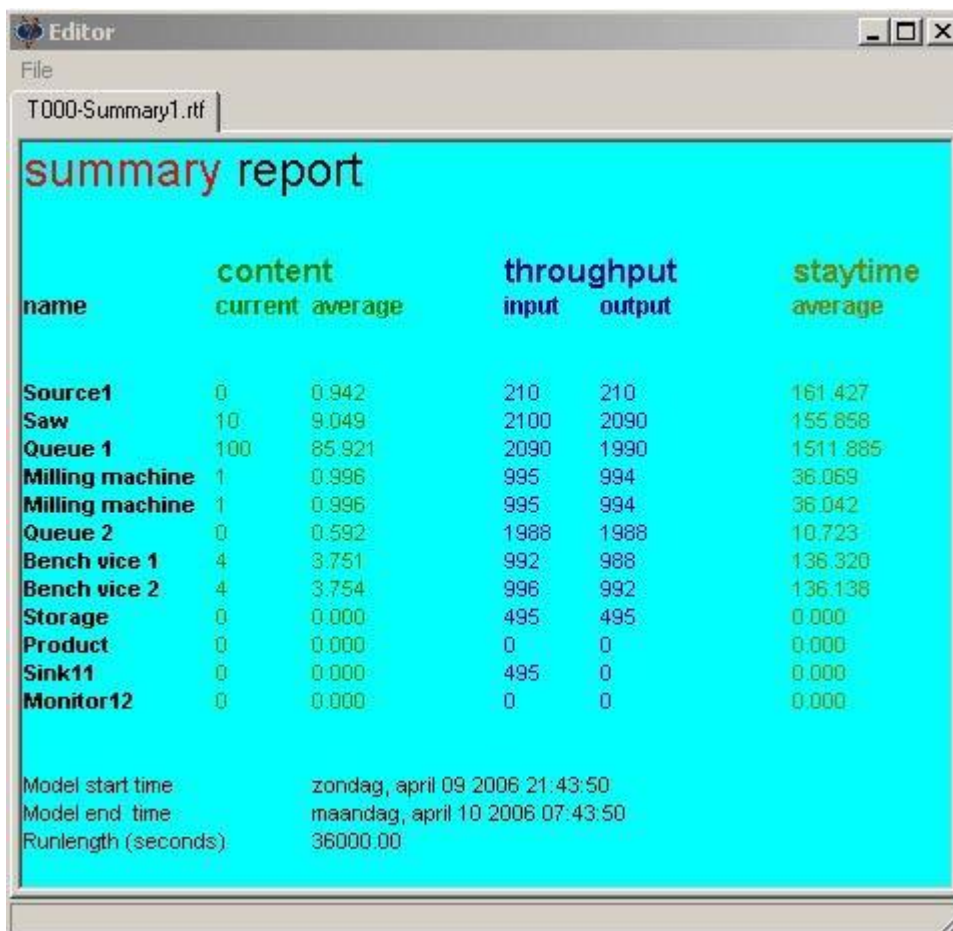
3. 报告及图表

测量结果的第三种方法是选择Results菜单中的Summary reports和Graphs选项。

首先我们来看Summary reports。它能让你对模型的状况有大致掌握，模型中需要区别以下几个特征：

- Current Content现有容量：报告产生时一个原子内的产品数量。
- Average Content平均容量：原子内平均产品数量。
- Throughput Input通量输入：已进入原子的产品数量。
- Throughput Output通量输出：已离开原子的产品数量。
- Staytime average平均停留时间：产品停留在原子内的平均时间。

By通过在Results菜单中选择Summary Report选项，就会产生如图5-6的报告。



name	content		throughput		staytime average
	current	average	input	output	
Source1	0	0.942	210	210	161.427
Saw	10	9.049	2100	2090	155.858
Queue 1	100	85.921	2090	1990	1511.885
Milling machine	1	0.996	995	994	36.069
Milling machine	1	0.996	995	994	36.042
Queue 2	0	0.592	1988	1988	10.723
Bench vice 1	4	3.751	992	988	136.320
Bench vice 2	4	3.754	996	992	136.138
Storage	0	0.000	495	495	0.000
Product	0	0.000	0	0	0.000
Sink11	0	0.000	495	0	0.000
Monitor12	0	0.000	0	0	0.000
Model start time: zondag, april 09 2006 21:43:50					
Model end time: maandag, april 10 2006 07:43:50					
Runlength (seconds): 36000.00					

图 5-6: 总结报告

从图5-6中可以看出，在milling-machine 1上平均36.069秒就出现一个产品，在bench vice 2上已有992个产品被生产出来，而且4个产品仍然在上面。

从saw及两个bench vices那三行中可以看出，对服务器上批量过渡的所有测量都是以“小”单位进行的，即每批所包含的单位量。例如，看到Source1那一行的output是210，Saw那一行的input是2100。因此锯子上的测量是针对于小段木材而不是大型木材，而台虎钳上的测量是针对于铣过的小段木材而不是窗框。这是ED7中引进的一种新的测量方法。（这对于新接触ED的使用者没有多大影响，但是对ED很熟练的使用者，这会影响到采集B：1服务器的状况以及1：B服务器的阻断状况。经过全面检查会发现，这种方法是唯一在所有条件下都能产生一致结果的方法。另外，saw锯子的输出量（而不是输入量）也有可能不是10的倍数（如2098而不是2100），因为Queue1已排满，因而阻断了其它小段木材的进入。为了理解这一点，请注意Queue 1是一段一段将小段木材送到两台铣刨机上的。）

根据这份报告，每小时平均生产的窗的数量为49.5！

图表选项将以图形的方式呈现一个变量。使用这种方法时，需从History选项中选择你想在图表中呈现的原子。你可从Simulation菜单中选择History选项。

警告！ 这里要将‘General history’选中(打勾)，还要选择单个原子（通过选择‘One on’）（见图5-7）。初看可能选择所有原子的历史‘All on’会更好，但是，用这种方式记录原子的历史，尤其是大型模型中，会导致仿真速度放慢。

在这一阶段，由于图表原子还有待于编程，我们只通过菜单选项中的Graphs来创建图表。图表原子对于高级用户来说会比较方便，因为所有可能的变量都能通过4DScript进行定义。



图 5-7: History（历史）

点击Results菜单中的Graphs选项，会出现一个窗口，你能从中选择你想要以图表形式呈现的原子。选中原子后，会出现如图5-8的窗口。

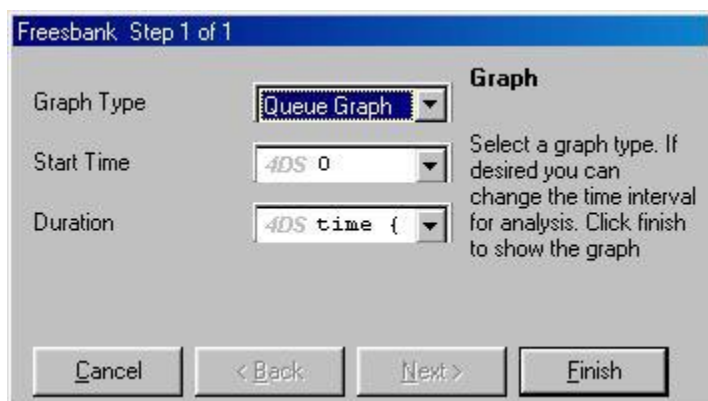


图 5-8: Graph输入窗口

首先我们先复习上面提过的两个输入栏：

- **Start time**起始时间
需要在这里输入图表的起始时间。如果这里的值显示为0，图表将会从仿真运行初始时开始呈现，如果输入的值是10，那么图表将会跳过最初运行的10秒钟。
- **Duration**持续时间
在这个输入栏，你需要说明图表记录的总时间。例如，若输入的值是3600，那么图表将会记录从起始时间开始的1个小时。“time”是默认值，代表仿真运行结束时所经过的模拟时间。

在第一个输入栏Graph Type中，你可以选择你想要的图表，共有5种类型。

1. Queue Graph（序列曲线图）

在此曲线图中，你能将一个原子内的产品数量呈现出来。通常，在Server（服务器）原子中，产品的数量不会超过1，但是在Queue（序列）原子中，序列会在图表中呈现出来。图5-9代表了木匠工厂研究中仿真运行开始后，第二个小时到第四个小时的模拟时间内的第一个缓冲器的曲线图。

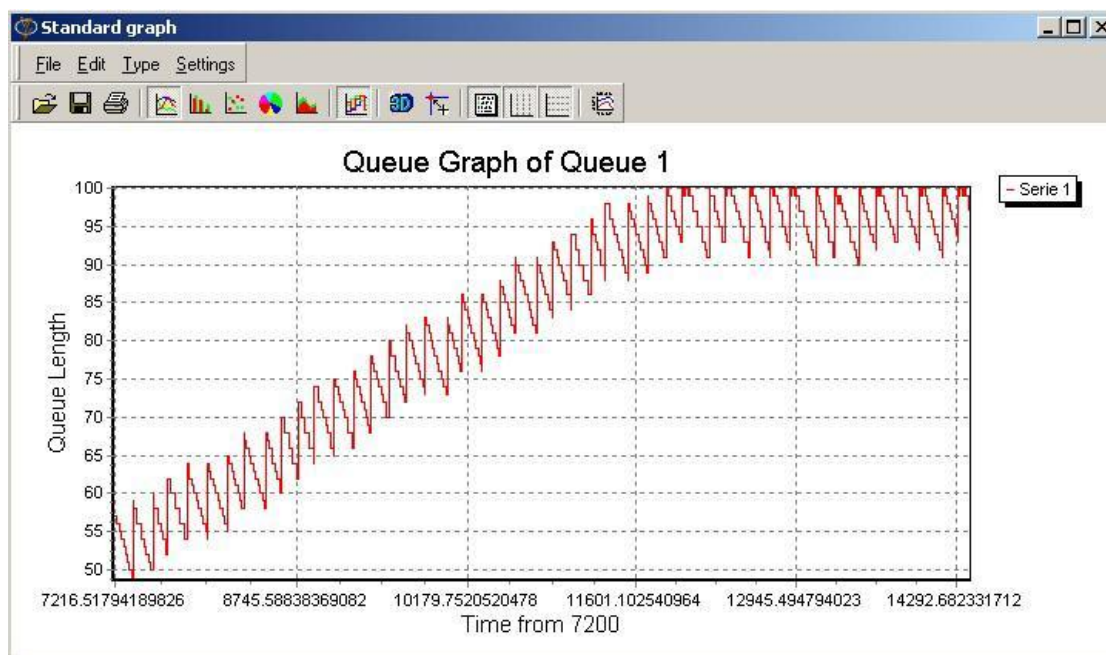


图 5-9: Queue graph序列曲线图

2. Queue Histogram（序列柱状图）

这里反映的是每个‘储存量’所需的时间百分比。

见图5-10，这里显示的是仿真开始后第二个小时到第四个小时内的Queue 1的不同序列长度的分布图。

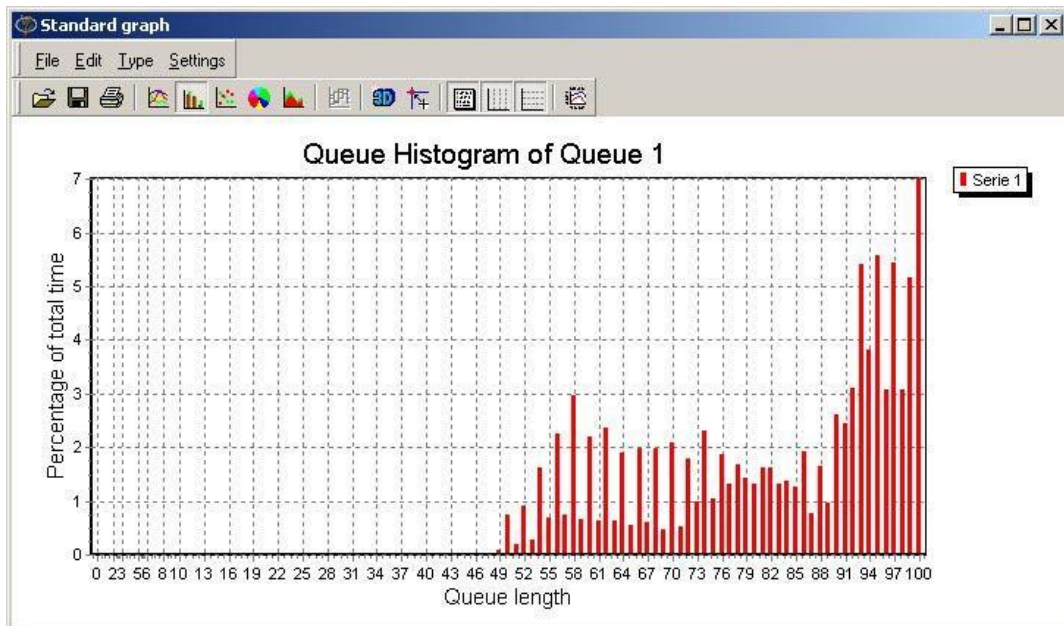


图 5-10: Queue histogram（序列柱状图）

3. Status Pie (状态饼图)

你可以在这里用一个饼形图呈现一个原子的状态，如图5-11，反映了木匠工厂研究中第一、第二个模拟小时内第一台铣刨机的状态饼图。

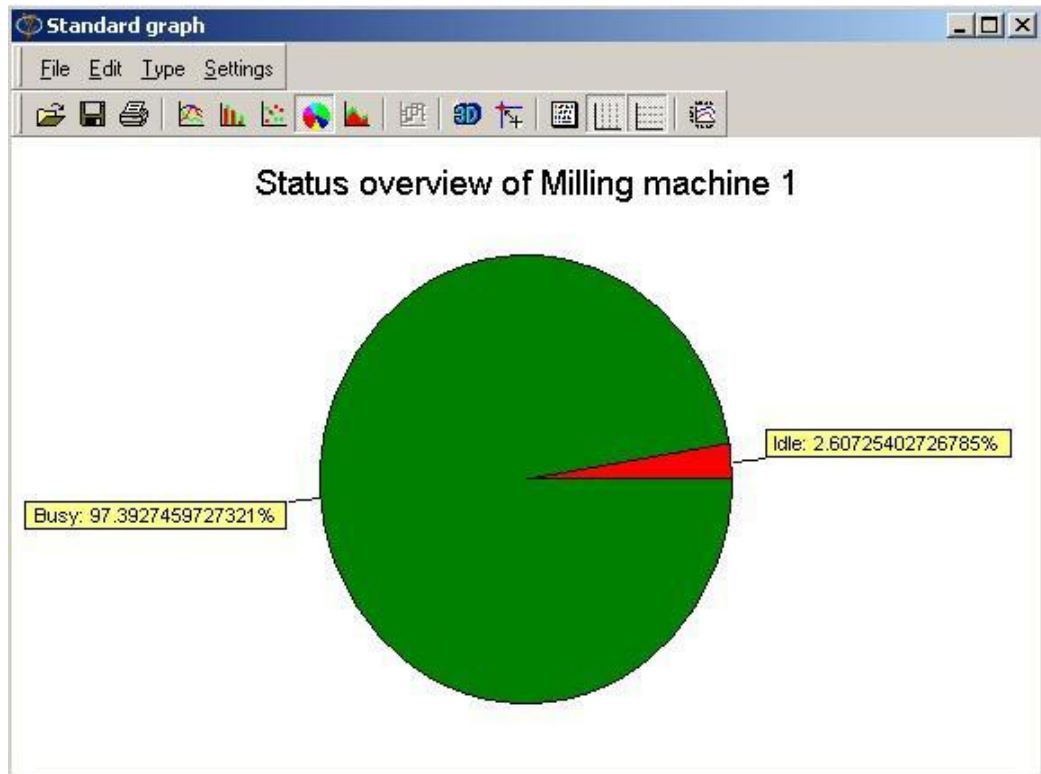


图 5-11: Status Pie (状态饼图)

4. Status Bar (状态圆柱图)

通过选择Status Bar选项, 也可将一个原子的状态用圆柱图呈现出来。图5-12即是图5-11中的测量结果用圆柱图呈现出来。

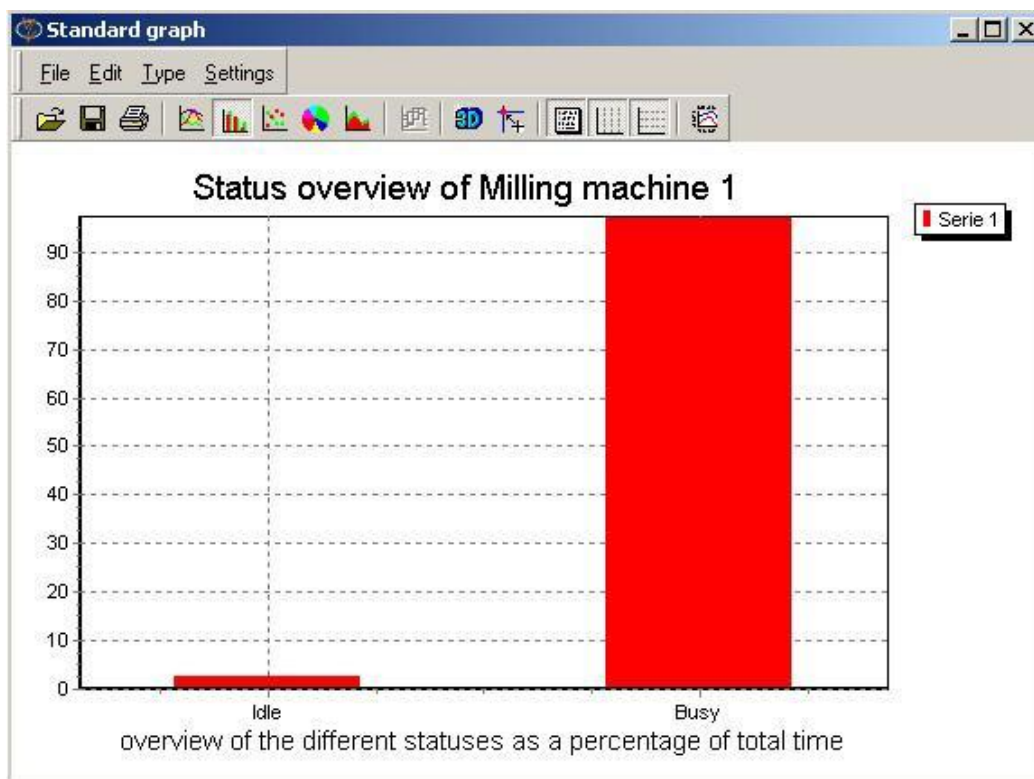


图 5-12: Status Bar (状态圆柱图)

5. Wait Histogram（等候柱形图）

等候柱形图反映的是一个产品在原子内的等候时间。如图5-13，反映的是第一、第二个模拟时间内小段木材在Queue 1中的等候时间分布图。

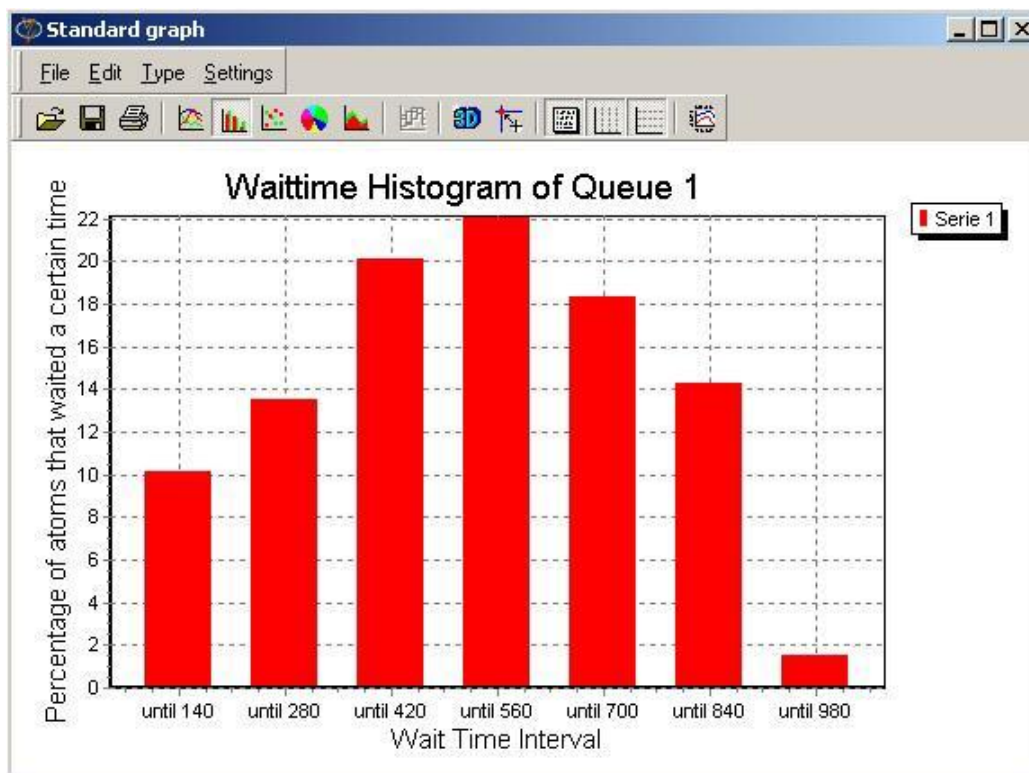


图 5-13: Wait Histogram（等候柱形图）

4. 实验

以上三种测量结果的方法对于在仿真过程中直接显示结果非常有用，但是对模拟结果作出结论就显得不太合适。这时，实验的方法就显出它的优势。

到第五个版本为止都是用两个原子来完成此操作：即Experiment Atom（实验原子）和一次及多次Performance Measures (PFM)性能测量，这两个是要得到的输出变量。但是在第六个版本中我们在菜单中添加了Experiment（实验）选项。

如果你在一个老的模型中加载Experiment原子和PFM原子，我们强烈建议你用新的实验向导来设计一个新的实验：这可能会花几分钟时间，但是绝对值得一试！

对于实验向导及结果分析的详细解释，我们可参照附录4或通过帮助菜单中的指南选项。

区分Simulate和Results选项与Experimentation选项很重要：

在建立模型及测试模型的过程中我们使用前两个选项来得到单个运行的结果，而Experimentation选项在以后的过程中将会被使用到：我们或多或少会信任我们建立的模型，并想通过多次运行来得到结果，因为通过单个运行的结果我们无法得出结论。谁能保证（在相同条件下）另一次运行的结果会一模一样呢？

（续）案例研究任务

5. 设计一个实验，其中暖机时间为10小时，测试时间为100小时，并且每单位时间内（小时，天或星期或模拟周期内）窗的平均产量有95%的准确度。实验结果是否与预先计算的一致？是否与通过其它测量方法得到的结果一致？

我们将一步步示范如何完成这项任务。

基本上实验分四个步骤：

步骤1 确定实验设置

确定实验的设置，如运行次数，观察时间，暖机时间等等。

步骤2 确定性能测量

确定原子及一组原子的性能测量

步骤3 实验

确定以后实验的实际运行时间。这一步使用者不需要做任何事情！

步骤4 报告所做确定及分析

如果你的监控器原子的样本率仍然设为60秒，甚至是5秒，最好这时把它设为36000秒（10小时），或者是1E9（10亿秒，大约32年），因为在实验中你将不再需要用到监控器原子。而这将使你的模型在实验中运行得更快。

通过点击菜单栏中的Experimentation可将实验向导打开：首先我们设置好Experiment Settings中的观察时间，运行次数，暖机时间（见图5-14）。

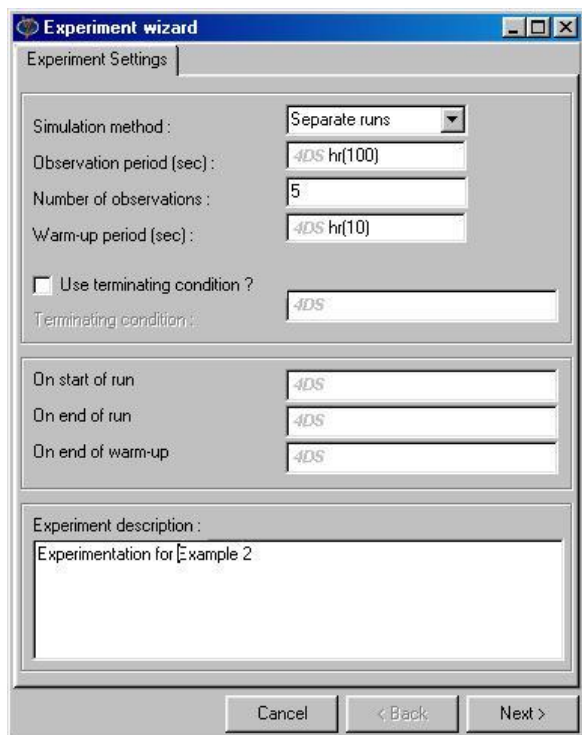


图 5-14: Experiment Settings（实验设置）

在Sink（接收器）中名为`Produced per hr`的一个Performance Measure (PFM)或输出变量被确定(见图5-15)。为了得到每小时窗的数量，这100小时内的输入被划分为100项。一定要在PFM中选择“[USER DEFINED]”选项，不要忘了移除“[USER DEFINED}”这一行，并且还要输入“input(cs)/100”。“cs”是当前选择的4DScript编码，也可是一组原子。

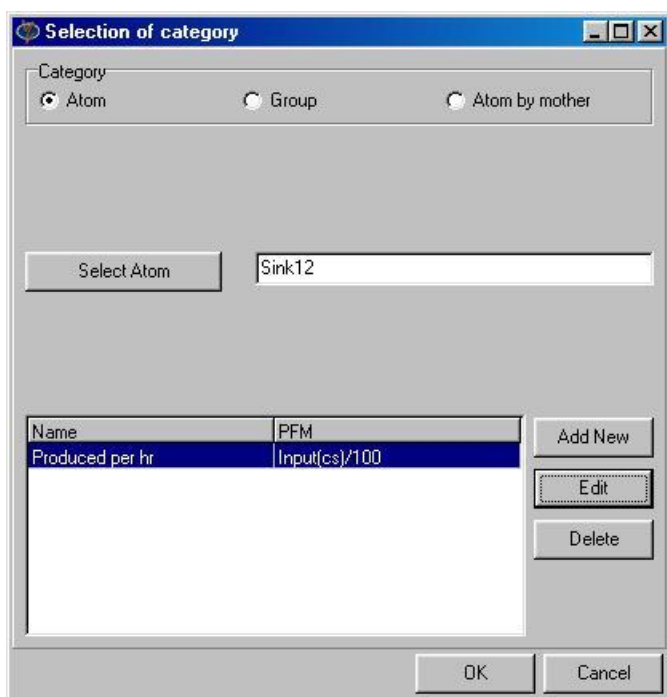


图 5-15: 确定PFM

模型的实际运行-在实验开始后- 就是步骤3(如图5-16) ，这也许需要几分钟时间：在每次100小时的运行中，500个窗需要2000段小段木材，200段大型木材！

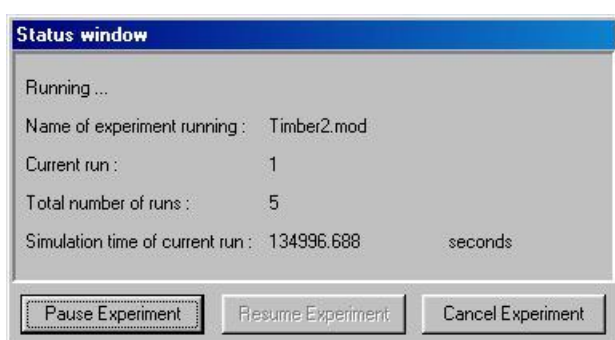


图 5-16: 实验过程

实验运行以后做出一份报告就很容易。见图5-17，这是一份报告的例子。这张表中的结果可以输入Excel (以.csv的文件名保存表格)或以更流行的方式输入QuickReport (见Report Preview)

Results Table							
Observation period :	360000						
Warmup period :	36000						
Number of observations :	5						
Simulation method :	Separate runs						
Description :	Experimentation for Example 2						
Atom :	Sink11						
		Average	St.Deviation	L-bound(95%	U-bound(95%	Minimum	Maximum
Produced per hr		50.000	0.019	49.977	50.023	49.980	50.020
XXXXXXXXXX	Raw Data	XXXXXXXXXX					
Sink11	Produced per hr						
49.99	50.02	49.98	50.02	49.99			
Report Preview Save Table as .csv Close							

图 5-17: Results Table (结果表格)

这对我们的实验作出了结论。要想彻底检查可能的结果请参照我们的实验文档（见 [Help | Tutorials](#)）

这次实验的模型保存在timber2.mod的文件名下。

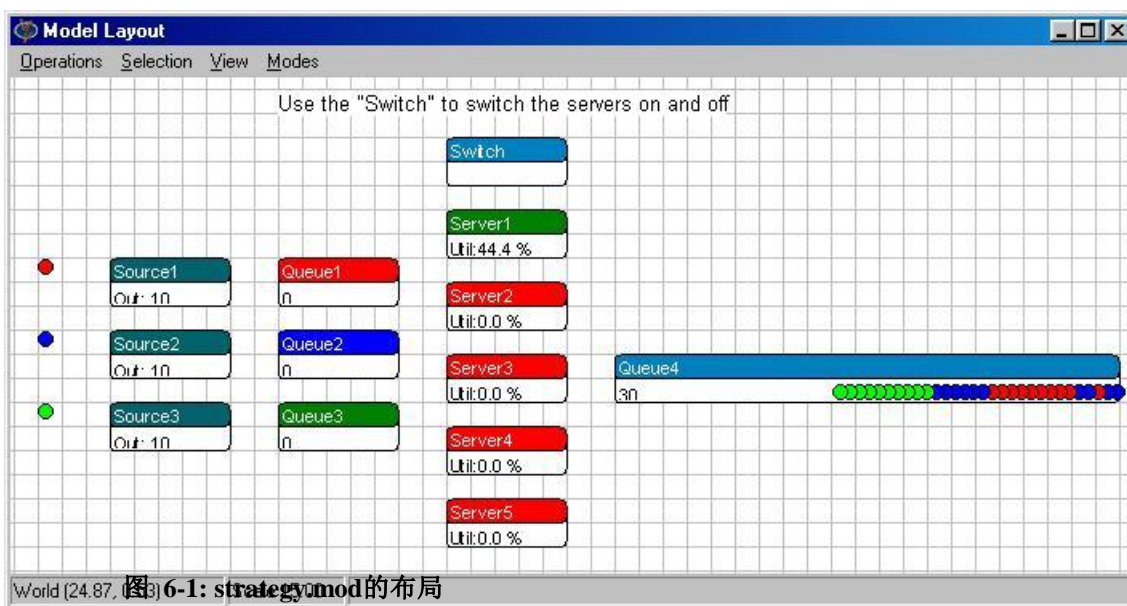
6 发挥策略

这一章的重点是，通过预定义的策略，产品进入下一个原子（输入策略），从序列中选出（序列原则），经过处理后重新分配（送到）。

为了示范这一点，我们建立了一个简单的模型。它由三个Sources（源），三个Locks（锁定）（一个控制产品供给的原子），五个Servers（服务器），四个Queues（序列）和一个Switch（开关）——一个自造的原子（Enterprise Dynamics使它具有适应性）。这个原子的作用是打开和关闭服务器。这里没有Sink（接收器），因而所有的产品都留在最后的Queue（序列）中（如图6-1）。

这个模型保存在strategy.mod的文件名下。

*提示：*最重要的原子及它们的作用，包括这些策略，都在附录2中有详细说明。在建立你自己的模型时可参照里面的注解！



*警告：*由于最后的序列原子的源编码已调整，已使用了一个新设计的原子，初学者无法自己建立这个模型。但高级用户能通过标准原子并在2D Draw事件处理器上调整最后序列的编码，使其能显示100个产品。因为开关也不是一个标准原子，高级用户需要手动打开和关闭服务器。在其他原子中可用性控制（Availability Control）原子也适用。

从以上三个源中，三种顾客进入序列。他们的到达以这种方式安排，左边的三个序列马上就挤满了。在每种顾客中，锁定每次只允许10位离开源。在序列中，顾客通过服务器的帮助到达最后的序列。开关

的作用是打开和关闭服务器。

原则上，三个序列能通往所有的服务器，到达第一个服务器可经过通道1，到达第二个服务器可经过通道2等等。然而，在strategy.mod的原始模型中，只有服务器1是打开的（经开关）。所以在仿真中，只有第一个服务器是绿色显示，而其它的都是红色显示。另外，默认参数也使用到了，这代表：

1. 所有输入策略在输入通道上，
2. 所有序列原则遵守先进先出，
3. 所有送至序列的都被送至4. 任意开着的通道：在所有打开的输出通道中任意选择一个通道。

这代表每个开着的服务器都可能是序列的下一个站点。

在最后的序列中可见产品及它们的图标。因此，产品进入序列的顺序能清晰地看见。(若序列原则设为FIFO)。

由于产品离开源时得到一个标签，所以可根据标签将它们在最后的序列中分类。标签可视为挂在产品上的标记，可表明颜色，重量，型号或类似的特征。标签用4DScript定义，即Enterprise Dynamics的编程语言，并在大部分模型中都很重要。

查看strategy.mod的其它参数，可通过仿真运行了解更多。

完成以上步骤后，我们现在可以开始在实验中发挥策略了……

6.1 调整输入策略

输入策略控制从前一个原子到达后一个原子的通路（如从一个输出通道连接到另一个输入通道）。输入策略的作用就是定义产品进入可用通道的顺序。

你可以把输入策略想象成那些绿灯时间较长的路况上的交通信号灯，调控不同信号灯的顺序。

在三条输入策略中，第一条将所有输入通道打开，而后两条每次打开一个输入通道。

作为练习，我们现在来看看输入策略的改变对未经更改的原始模型上的服务器1的影响。按照以下方式更改输入策略，观察几次运行后，思考系统的执行发生了什么变化。

1. 任意输入通道

激活后，这一策略将一个原子的所有输入通道打开。如果与输入通道连接的原子不只一个能输送产品，那么与小号码输入通道连接的原子将会享有优先权。只要产品不断从第一个通道进入，其它的通道将被阻断。

2. 最大序列

激活后，这一策略打开一个原子的所有输入通道。如果与输入通道相连接的原子不只一个能接收产品，有最长序列或最大容量的原子将享有优先权。注意，若几个原子的序列同样长，则将选择小号码的输入通道。

3. 最长等候

激活后，这一策略打开一个原子的所有输入通道。若与输入通道相连接的原子不只一个能接收产品，等候时间最长的原子将享有优先权。若几个原子的等候时间同样长，则将选择小号码的输入通道。注意，与前一个选项不同的是，这并不代表序列的长度也接近。

4. 循环赛

这一策略首先打开第一个输入通道，然后等待产品通过这个输入通道被输送出去。在第二次循环时，则轮到第二个输入通道，并依次按此进行。当产品经过最后一个输入通道时，又返回到第一个通道中。

重要备注：这一策略在第一个产品进入后就被激活！因此，若存在三个输入通道，这一策略将以x,2,3, 1,2,3, 1,2,3的顺序运行，而x可以是1,2或3！

5. 通道 1.

在这一策略中，你可以输入一个具体的输入通道，并让所有产品都进入这一通道。若输入1，产品则只能进入通道1。注意，这个原则不能在第一个产品进入后即生效，因为一开始所有的通道都是打开的。

6.2 改变Queue Discipline（序列原则）

这一小节的目的是改变最后一个序列的序列原则，观察其结果。

将第一个服务器上的输入原则改为最长序列，这将确保产品无序地进入服务器，而使其他四个服务器关闭。因为若五个服务器都是打开状态，则很难看清原子的移动情况。有可能涉及以下六个参数（可参见附录2：序列原子）：

1. 先进先出

原子在序列中按进入的顺序排放

2. 后进先出

进入的原子总被放在序列的最前面，因此，产品离开序列的顺序与进入序列的顺序正好相反

3. 任意进出

这一序列原则将进入的产品放在序列中的任意位置。

4. *按照**标签**的顺序从小到大排列*
标签中值最小的产品排在序列的最前面。在此例中所有产品原子都有详细标签，可通过‘产品’标签顺序的排列顺序对这一选项进行检测。**警告：**若产品未经适当排列，将会在标签名称前或名称后出现空格。
5. *按照**标签**的顺序从大到小排列*
标签中值最大的产品排在序列的最前面。
6. *用户可自行定义*
用户可自行定义产品在序列中的位置。可将值5代入进行检测。

6.3 调整Send to（送到）语句

在这一章的第三节中，我们将讨论与服务器相连的三个序列中一系列‘Send to（送到）’语句。‘Send to（送到）’语句将确定产品的输出通道。用户可输入数值，或在21 (!)预定义选项中选择。可参见附录2对源原子的详细描述。

首先，我们打开五个服务器，否则可输送的产品很少。打开后服务器将呈现‘绿色’允许产品经过。

在序列上尝试一下设置：

1. *具体通道:始终送到通道1.*
产品原子始终被送到指定的输出通道。如输入通道3。
2. *一个打开的通道(从第一个通道开始): 查找通道，从第一个通道开始，送到第一个打开的通道*
产品原子被送到Enterprise Dynamics找到的第一个打开的输入通道。查找通道时从第一个输出通道开始（通道1）。
3. *一个打开的通道(从最后一个通道开始): 查找通道，从最后一个通道开始，送到第一个打开的通道*
查找通道时从最后一个输出通道开始，产品原子被送到Enterprise Dynamics找到的第一个打开的输入通道。
4. *任意打开的通道:在所有打开的输出通道中选择任意一个通道*
Enterprise Dynamics会任意选择一个通道。如果仿真运行时间长，那么服务器等的利用率会同样高。

5. *按照百分比: 90%的产品经过通道1, 剩余的产品经过通道2。*
一定百分比的产品送到特定的通道, 而剩余的送到另外的通道! 用户可自行输入通道及百分比。尝试将75%的产品送到服务器1, 将剩余的25%送到服务器5。
6. *用户自行定义: 自行输入4DScript表达式, 输入值介于1与通道总数之间: 1. 可按4DScript编辑器的小按钮。*
用户自行编写4DScript编码确定输出通道。点击文本旁边的小三角会出现4DScript 编辑器。例如, 输入2。

这一节总结了策略的使用。当然, 使用ED的编程语言——4DScript, 可产生更多的策略。

7 更多原子：从集合到解包

在前面几章中，我们只描述了基本原子。随后我们练习建立模型以及用简单模型进行实验。在这一章中，我们将介绍8个新原子，用这些原子我们对更复杂的情景也能建立模型。另外，我们将详细介绍ED中的可视化。

这章将要描述的原子是：

- **集合**
当几个原子组合成一个新原子时就成为集合原子。原先的原子或者被储存（被包装）或者被破坏（被集合）
- **解包**
这个原子能分解已被集合的原子。例如，如果一个集合原子填满了整个容器，解包原子能将它再次分解。
- **容器**
容器被用来储存或包装产品。如一个盒子或一个托台。
- **储运机**
储运机既是一个输送系统又是一个缓冲器。若运输机前的装载被阻断，那么负载将会继续向前移动直到另外已储存的负载阻断其移动。
储运机被用来建立辊子输送机的模型。
- **非储运机**
非储运机与储运机非常相似，不同的是若过程被阻断，它将不再向前输送产品。因此产品之间的距离始终相同。非-储运机被用来建立链式输送机。
- **多功能服务**
多功能服务原子可视为一组平行的服务器：它具有服务器原子的基本功能，并且允许同时处理多个产品或处理单个产品。
- **锁定**
锁定的功能是只让一定数量的产品通过，而其他产品被阻断。
- **解锁**
当一个产品离开解锁原子时，锁定原子才能让另一个新原子通过。这样就能控制模型中的工作量及运输方式的总数。

以上所有原子都会在例子中详细描述，附录2中有对这些原子进行全面阐释。

在一个工厂的分派部门，产品都堆积在托台上。产品和托台通过两台分开的输送机到达这座托台堆垛机或托台堆垛机械上，堆垛机不分型号，能将产品向上堆积，或将产品相邻堆积。

堆积了产品的堆垛机经下一台输送机到达包装机器，产品在这里用塑料包装。包装机器能同时包装来自不同托台的产品，也能包装来自单个托台的产品。

该系统由以下几个元素组成：

- 每隔5秒，一个产品就经过辊子输送机到达托台堆垛机。
- 按照负指数分布，平均每隔40秒一个托台经过另一个辊子输送机到达。
- 倘若所有产品已到达，托台堆垛机每次放8个产品在托台上，并且每个托台需要整整20秒时间。
- 链子输送机将所有托台输送至包装机器。
- 包装机器最多能同时用塑料薄膜包装4个托台，而对于每个托台，（根据负指数分布）平均需要120秒将托台上的产品用塑料薄膜包装好。
- 输送机的长度在这里不会有影响：选一个大约10m长度的就行了。速度始终是1m/s。每个产品长度50cm，托台的宽度和高度分别是1m。

我们要在Enterprise Dynamics中建立这个系统的模型：我们用储运机来代替辊子输送机，而用非储运机来代替链子输送机。另外，用集合原子代替托台堆垛机，用多功能服务原子代替包装机器，用容器代替托台。

但是……在正式开始前，先试试这一系统能否承受即将引入的负载量！

大多数原子被移入模型中都不会出现问题，因为它们与其他原子以正常的方式连接。

托台必须经过第一通道进入（一个小正方形）集合原子，而产品必须经过第二通道进入集合原子。

双击集合原子确认每个托台上的产品数，这将在一个输入窗口中显示各种制图表。你可以在制图表中（如图7-1）输入每个输入通道上要集合的原子数。对原子和集合原子的详细描述可参照附录2！

集合原子自动将产品堆积到托台上。为每个产品预留的空间在容器原子中设定。为了避免所有产品像柱子一样向上堆积，必须在产品原子中调整产品的型号。这点可参看例2种的解决方法！

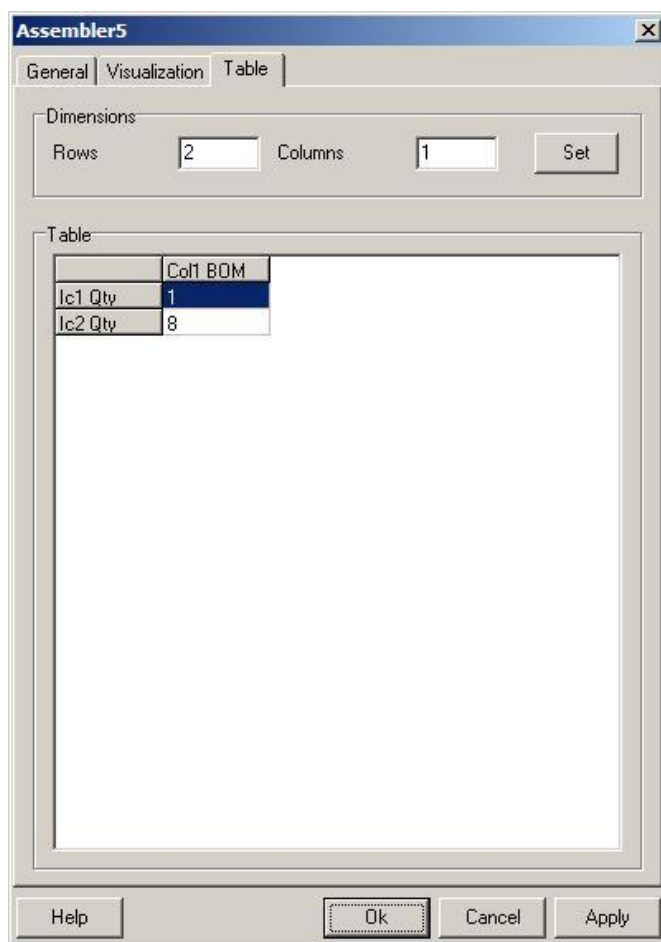


图 7-1: 集合原子的材料单

当所有的原子被移入模型时，模型的分布应近似于图7-2。

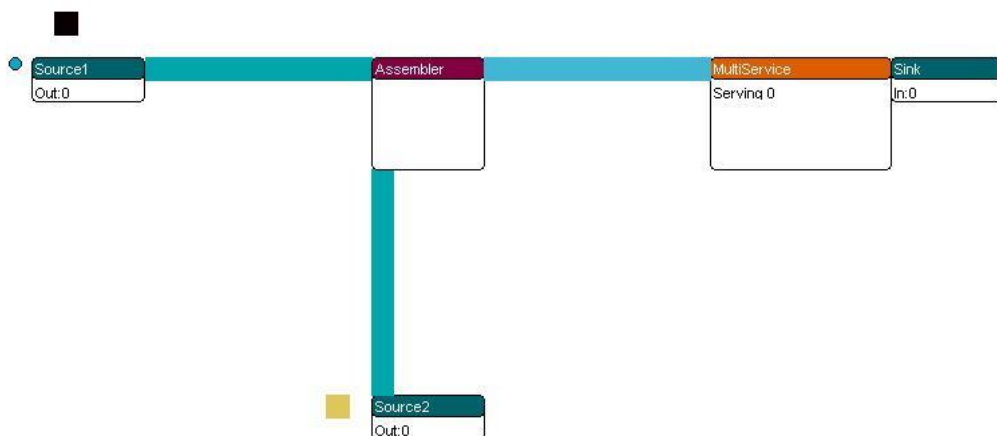


图 7-2: 输送系统的分布

保存你的模型。如果不能确信自己的模型是否正确，你也可使用下一节提供的文件名为 conveyor1.mod 的模型。

让系统运行一段时间，观察它在三维中的运行情况。试着改变产品或容器的型号，看会有什么变化！再试着改变传输速度或处理时间：例如，给多功能服务原子设定一个更长的处理时间，这将引起包装机器的堆积堵塞，并清晰地呈现非储运机的工作过程。

1 为了控制工作量，系统每次处理不超过5个托台。请适当调整模型。

这可通过在模型中移入一个锁定原子及解锁原子。锁定原子及解锁原子在生产过程中被插入：锁定原子是在限制开始时被插入，而解锁原子是在限制结束时被插入。当预设数量的原子经过后，锁定原子会自动关闭，而解锁原子却正好相反。

若只插入锁定原子，则每次只有比如，5个原子能进入系统。再通过插入解锁原子，我们就能确保在同一时间内在系统内的原子不超过5个。锁定原子总是位于源后面，因为如果不这样，源就会误认为锁定原子也是一个产品，从而将它送至系统中。解锁原子总是在原子数不需限制时被插入，如某台机器后或接收器的前面。在此例中，解锁原子被直接安放在Sink（接收器）的前面。

通过插入解锁原子，当一个原子离开解锁原子时，锁定原子会自动打开。锁定原子与解锁原子自动相连。然而，若必须经过手动连接，那么锁定原子的第二个输出通道必须与解锁原子的第二个输入通道连接起来。要想仔细观察锁定原子与解锁原子的运行模式，可暂时将锁定原子设成1个或2个托台后就处于关闭状态。可参见附录2！

现在可以调整模型或直接打开文件名为conveyor2.mod的文件。

2 公司现在想在包装过程后将托台与产品分开，并用另外一个传送系统将它们送回托台堆垛机。在系统内的托台数仍不超过5个。将模型作适当调整。

为了能再次使用，必须做两点调整：1）将产品与托台分开，2）将托台运送回集合原子。为了将托台从产品中分开，就需要用到解锁原子。位于多功能服务原子后，解锁原子将产品送至第一输出通道，而将托台送至第二输出通道。通过在第二输出通道后连接上链子输送机，再将它与现有的辊子输送机相连，托台就能再次被使用。

由于托台不离开系统，解锁原子就显得多余了。我们也可省去锁定原子，而在服务器上将产品数量设为5。若辊子输送机也算进模型内的话，分布图应近似于图7-3和7-4(conveyor3.mod)。

将输送机设为一定长度，可见托台的循环使用。

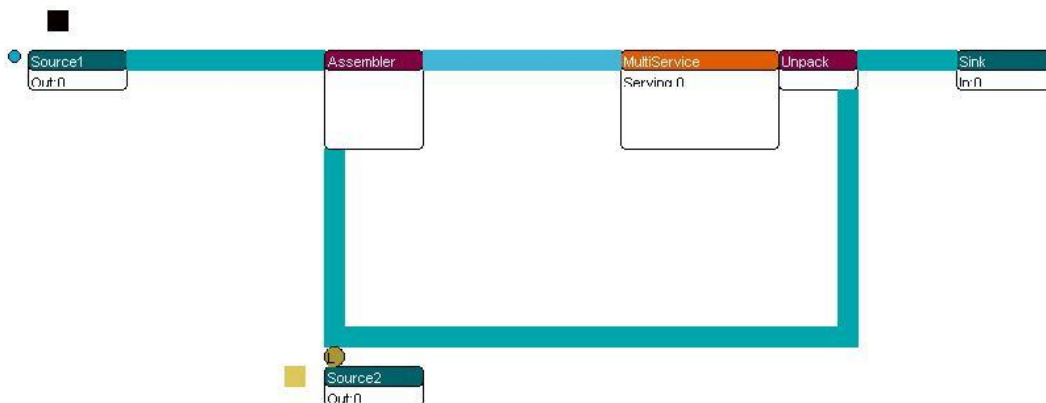


图 7-3: 二维中托台的再利用

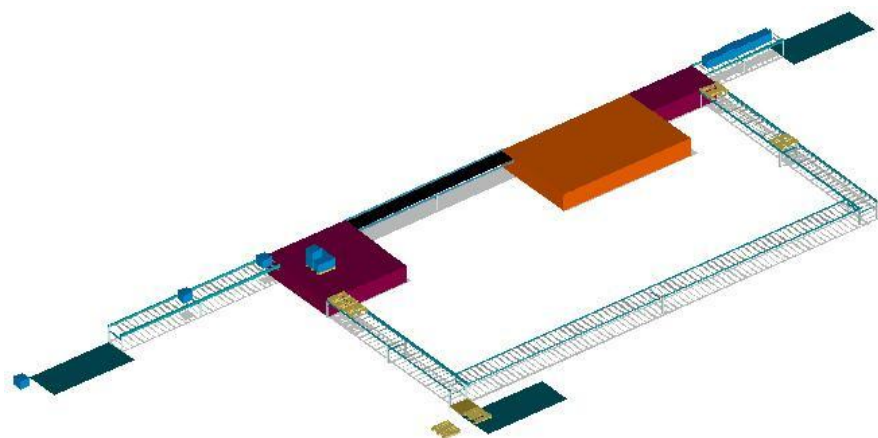


图 7-4: 三维中托台的再利用

当然，除了方形输送机，你也可以使用环形输送机。图7-5中的分布图展示的是环形的，可调节高度的输送机。

两个模型都能在conveyor3.mod及conveyor4.mod的文件名下找到。

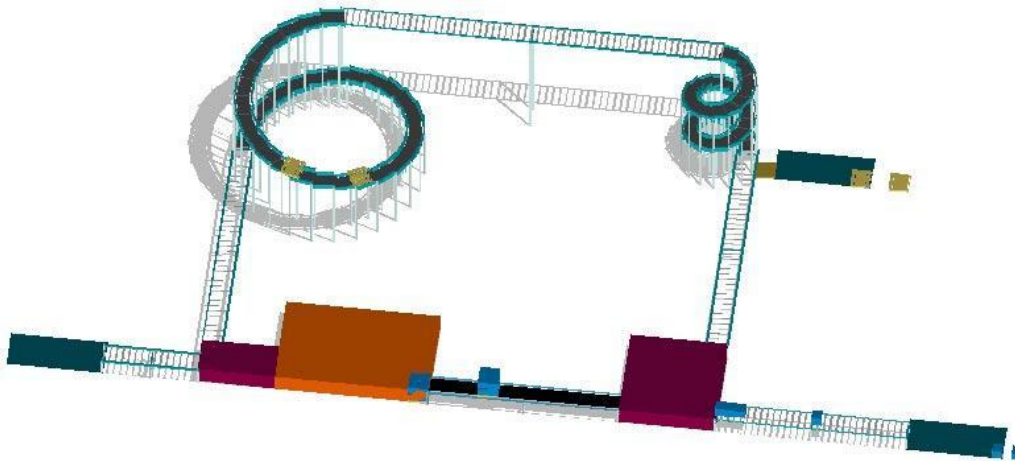


图 7-5: 可调节高度的输送机

最后，对于conveyor3.mod的模型我们来问几个问题：

1. 托台循环使用的周期多长（不包括等候时间）？
2. 系统中若托台太少，会影响每小时产量，而额外的托台——在一定时间内——对于提高每小时产量没有帮助。请解释原因！

通过实验确定系统中托台的最佳数量。（可通过监控器观察接收器中每小时产量，并通过锁定原子或源改变托台的数量）

8 ENTERPRISE DYNAMICS与EXCEL®

我们可以从Enterprise Dynamics中访问其它软件程序，如Excel, Word或Access，以在磁盘中输入或写入文件。在这一章中，我们将通过之前的邮局的例子来讨论ED与Excel之间的连接。

我们将继续介绍一系列4DScript命令及ED中标签的重要概念。

在这一章结束时，用户将知道如何访问Excel，另外，还将掌握4DScript的初级基础。

8.1 银行

例 4

我们选了鹿特丹的一家银行作为实验对象，测试服务顾客的新方法。现有需要不同服务时间的几种类型的顾客。

为了便于解释，在此例中我们将顾客的类型限为两种，那么所需服务时间为1分钟的顾客归为类型A，每小时有平均50位类型A的顾客来到银行；而所需服务时间为10分钟的顾客归为类型B，每小时平均有5位类型B的顾客来到银行。

这两种类型的顾客都排在一个序列中，有两个柜台提供服务，按照“先来先服务”的原则。

假定整个到达过程呈指数分布分布，且整个服务过程是持续的。

问题与任务

- 1 这一系统的利用率有多高？
- 2 假设每种类型的顾客有各自的序列及柜台（服务器），应用序列理论，计算他们的平均等候时间以及平均序列长度。
- 3 你能否对一条序列时的系统的特征进行评估？请说明为什么会有这样的特征，或说明为什么没有？
- 4 哪个系统的平均等候时间更短？请说明你的理由！

先尝试着回答这几个问题，理论上来说这几个问题都很重要；我们将稍后再来解决。对于M/D/1、M/D/2系统中的序列理论不熟悉的使用者可跳过问题2、3。

现在，我们先来建立只有一个序列的模型。建好之后，我将会访问Excel。图8-1即是已连接好通道的模型分布。

为了区分两种类型的顾客，类型A顾客用蓝色圆点表示，类型B顾客用红色圆点表示。

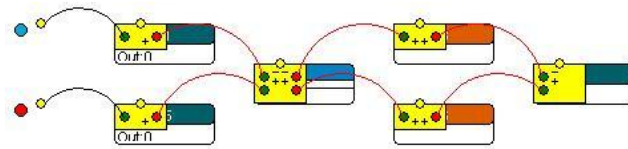


图 8-1: 银行的分布

现在需要在服务器上设定银行工作人员为一位顾客服务所需的时间。所需服务时间取决于顾客的类型，也就是说，需要在几个公式中进行选择！

这个问题可以由通常贴在产品上的标签来解决。标签可以代表颜色，重量，条形码或服务时间。一个产品可以同时有几种标签，而且每个产品的标签的种数也可以不同。

我们最好尽早把服务时间设定好，将这一时间标在标签上，同时说明开始服务的时间。这样做的好处是我们不用在服务器上查看顾客的类型，而只需看顾客标签。另外，有三种类型或更多类型的顾客时，这一做法也适用！

当产品从第一个和第二个源出来到达‘Trigger on Exit（开始退出）’时，我们就将服务时间的标签挂在产品上。

```
setlabel([servicetime],mins(1),i)      为类型A顾客
setlabel([servicetime],mins(10),i)     为类型B顾客
```

重要!

这里需要对ED的内在结构及编程语言4DScript有所了解。附录2中，在“服务器”下面有对Setlabel及Label命令的语法和使用进行解释。

然而，要想更多了解4DScript的句法和结构，可从现在开始参阅英文版的附录3！

提示!

- 1 为了系统地增强对4DScript的认识，你可以在后面的学习中不断参阅附录3。到时掌握使用手册中的1000条命令就显得轻而易举了。
- 2 在栏中双击可输入4DScript，激活4DScript编辑器。按F2键可导出4DScript名词列表以及对每条命令的简单注解。

在服务器的周期时间内需要用到上面所说标签，并作为服务时间使用：

`label([servicetime], first(c))`。 所以，不是直接输入一个数字，而是要输入一串4DScript命令来产生一个数字。

Label命令的语法及first(c)的含义可参照附录3。

整个模型可在bank1a.mod的文件名下找到。这个模型经过简单修改就能成为一个有独立序列的系统 (bank1b.mod)。

- 5 在两个系统中分别运行100小时，计算出平均等候时间。结果是否与你的直觉一致？

8.2 与Excel连接

我们不单单依赖于ED中的数据，而是将如等候时间、生产时间等原始数据输入Excel中。通过Excel中的数据分析选项，我们能确定如平均值等数据。

一般程序

1. 与创建模型一样，在目录中创建一个新的Excel文件。
多数情况下将会是C:\Program Files\Enterprise Dynamics\Work.

别忘了关闭Excel! 此例中的文件名为bank.xls.

2. 将Excel原子置入模型中。这一原子可在库中‘Data’类别下或在Speedbar上找到。然后这一原子将‘建立’起ED与Excel之间的连接。在每个模型中，最多包含一个Excel原子，且不经通道与其他原子相连。

3. 双击Excel原子将出现以下屏幕：

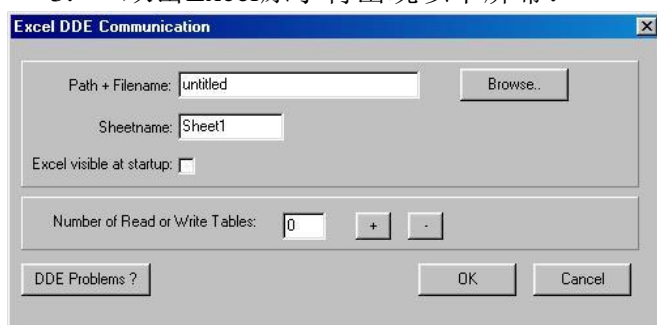


图 8-2: Excel对话框

点击Browse选择新建的Excel文件或查找旧文件，选择文件所需的工作表。ED中的工作表名称必须与Excel文件中的工作表名称一致！在‘Excel visible at startup’这一栏中打勾，可见Enterprise Dynamics背景下的Excel。完成之后，点击OK，就打开Excel文件，在任务栏中就出现了Excel图标。

可在Excel原子中见到——若缩小到一定程度——已建立好的连接。关闭模型同时会断开与Excel的连接。重新打开模型时会自动建立起连接；因此以上1-3步的程序不需每次都手动重复！

*顺便提一提：*这一连接被称为DDE连接(DDE = Dynamic Data Exchange)。因此要进行交换的两个程序都必须打开。另外，只有一个Excel文件能被打开，因此不能选择分开的输入及输出文件。

然而这在ED中却可以做到，因为ED中的ActiveX是连接两个程序的高级的，且更稳定的工具。这一选项在这里没有提到。

若所有步骤都正确的话，就能创建如下的分布图：

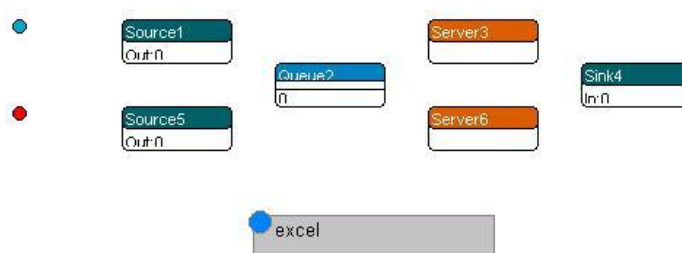


图 8-3: 与Excel连接的银行

若连接中出现问题，请先参照章节8.5中‘故障诊断’！

8.3 在Excel中写入数据

为了看看此例中的连接情况，我们将每位顾客的等候时间输入Excel中。这可通过使用

Queue中‘Trigger on Exit（开始退出）’上的命令：

`excelwrite(output(c),1, age(i))`

Excelwrite(a,b,c) 将c表达的结果输入Excel文件的 (a,b) 单元格中。*age*（年龄）命令记录了产品从到达模型开始的时间。在此例中，到达时间从产品离开Queue（序列）时开始算起，因而与产品的等候时间吻合！

Output(输出) 命令代表到那一刻为止经过一个原子的产品总数，相当于一个出纳员。

总结：第*i*位顾客的等候时间在Excel文件bank.xls中位于第*i*行，第一列。

现在将仿真速度设为一个较低的速率，重新启动仿真。接着开始运行并利用任务栏将ED转换成Excel文件。如果不出错，将会出现‘奇妙’的事情：

与ED中的过程一样，等候时间被写在bank.xls的第一列！

考虑一下这个问题：若对一项数据可行，那么对更多项数据也可行。显然，这就代表可将大量原始数据储存或输入到Excel中，随后可应用Excel工具栏中的所有工具。

对那些屏幕上没有任何显示或只显示一连串#####的用户，可先查看‘故障诊断’。若显示所有数据的单元格所在的列太窄，就会出现这样的问题。这时，只需增加那一列的宽度即可。

将生产时间输入第二列时，只需通过Sink（接收器）中Trigger on Entry（开始进入）的命令：`excelwrite(input(c),2,age(i))`。

警告：第100位顾客的等候时间在第100行，第一列，但是这位顾客的总时间并不一定在第100行，第二列。请解释为什么！

6 在Excel中计算前1000位顾客的平均等候时间及平均总时间。

与Excel连接的模型可在bank2a.mod及bank2b.mod的文件名下找到。Excel文件命名为bank.xls.

对于狂热者：

7 在第三列中输入类型A顾客的等候时间，在第四列中输入类型B顾客的等候时间，并计算一顾客要足够多一平均等候时间。这一结果是否与原来的结果一致？

8.4 将数据从Excel中读入ED

我们已经成功地将ED与Excel连接起来，并将输出数据写入Excel，但是，也经常需要将数据从Excel中读入ED。如何将数据从Excel中读入ED是我们将要介绍的最后一个主题。

在我们的例子中，我们也可能实际算出顾客的中间到达时间。而我们就要将这一到达模式读入ED中并进行仿真运行。

运用一个公式，在postoffice.xls的第一列和第二列中输入100个（正的！）数值。当然，你也可以用刚刚输入的等候时间……

不管用哪种方法，这时最好将之前的excelwrite命令删除。

在Interarrival-time下的Source（源）输入窗口中，给类型A顾客输入一下命令：

excelread(output(c)+1,1)

在Queue（序列）的Trigger on Entry（开始进入）中输入：*if(output(c)=100, closeoutput(c))*

excelread(a,b)命令读出Excel(a,b)单元格中的值，而closeoutput命令确保100位顾客后，源原子停止向外输出，从而也停止了‘reading-in data process（读入数据过程）’。如果不执行这一命令，ED会继续读入数据，而将空单元格也当成是值为0的中间到达时间。

想想类型B顾客的语句应如何读入！

模型可在bank3.mod的文件中找到。

总之：到达模式由源产生。对于有固定进度表或固定时间表的生产，ED使用Arrivallist原子。这一原子也可用来在Excel适当格式中剪切和粘贴（大型）进度表。所以外部表格中的数据也适用，那么，至少对于数据输入来说，与Excel的开放式连接是多余的。

通过表格在ED读入数据会大大加快速度，所以在大多数情况下都推荐使用这一方法！

8.5 故障诊断

由于各种各样的问题，DDE可能无法连接：有时是因为ED中的不正确的设置，但也有可能是由于Excel的窗口设置导致的。我们从经验中得知，这类问题需要立即解决；一旦问题处理好了，连接就能正常工作了。

1. 找出你的Excel版本使用的语言。显然每种语言涉及到行和列时会不同，而可应用的参考文字必须能让ED识别！

可通过在File | Preferences中用Excel的默认语言在行及列中输入文字。

英语(默认语言)为: C和R (列和行), 德语为: K和R, 荷兰语为K和R, 等等……(如图8.4).

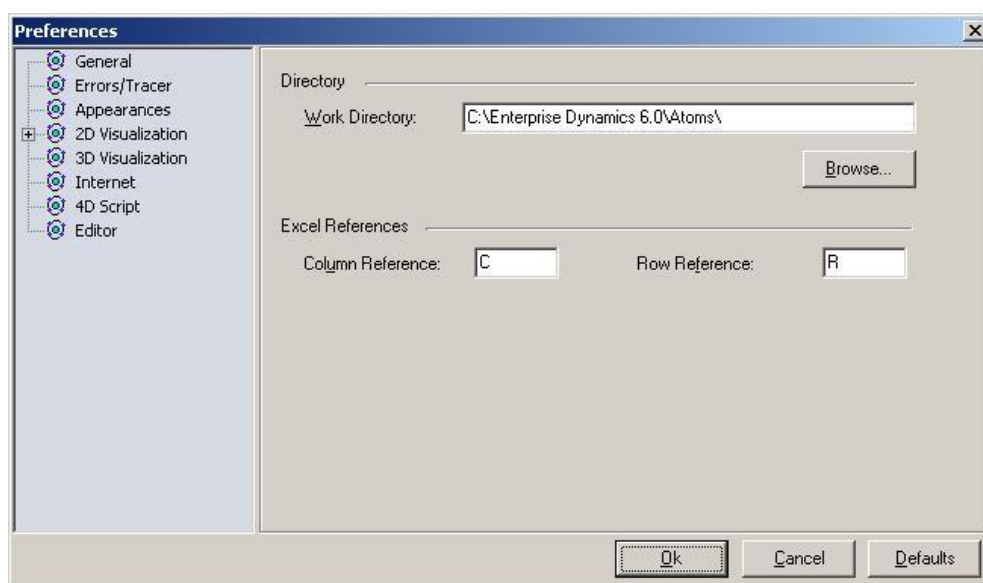


图 8-4: The File | Preferences 窗口

2. 复习章节8.2中的建立连接的3条指示:
 - a. 工作目录中Excel文件的名字是否正确(见图8.4检查工作目录)。
 - b. 你是否选择了正确的工作表?
 - c. 你是否偶然创建了几个同样名字的Excel文件?

如果你不能确定，可将所有Excel文件关闭，严格地重复1到3的步骤。

3. 已建立连接但Excel中出现变码：#####。
这是因为不同国家使用句号和逗号的十进制字符不同：
点击Start, Settings, Control Panel, Regional Settings | Numbers, 选择：.(句号)
作为十进制字符，选择：,(逗号) 作为数字组符号。正确设置好这一项
后，可能还需要调整Excel中列的宽度。

8.6 结合还是分开？

新邮局的结果值得我们注意：单个序列的模型(bank1.mod) 的平均等候时间比分开序列的模型(bank2.mod)的等候时间长！

我们发现以下结果：

平均等候时间：	单个序列 (分)	分开序列 (分)
类型A顾客	6.18	2.45
类型B顾客	6.18	25.7
所有顾客平均值	6.18	4.56

图 8-5: 仿真运行结果

为什么会有这样的结果？我们不是一直都认为结合就代表着提高吗？

原来，在分开序列的模型中，类型A顾客将会在第一个柜台前形成序列，而为类型B顾客服务的柜台人员将会有空闲弄弄指甲。所以，问题的关键是：变化！

当只有一种类型的顾客时，‘结合’会减少平均等候时间，但是，就像例子中的一样，当有两种类型的顾客时，就不会减少平均等候时间了。

根据序列理论（这里不深入讨论）：在分开序列的系统中，有两个独立的M/D/1模型。因此，对于最重要的序列特征可用一套闭公式。

这些公式得出类型A和类型B顾客的平均等候时间(W_q)分别为：
 $W_q = 2.5$ 分钟 和 $W_q = 25$ 分钟。

因此所求的所有顾客的平均等候时间为 $50/55 \times 2.5 + 5/55 \times 25 = 4.54$ 分钟

在单个序列的系统中，形成了一个有两种类型顾客的M/G/k系统。为了应用序列理论中的公式，我们把两种类型的顾客‘融合’成一种。他们每小时的平均到达率为55，预期服务时间 $E[S]=50/55 \times 1 + 5/55 \times 10 = 20/11$ 或者1.81分钟。

服务时间的方差 $\text{Var}[S]$ 为6.69，方差系数C，即 $\text{Var}[S]/(E[S])^2$ ，是2.025。

基于这些数据，得出等候时间大致为——这里没有给出公式—— $W_q = 6.25$ 分钟。

我们的运行结果与这些理论值基本一致：

在分开序列中平均等候时间减少了27%!

原因？在分开序列中，不规则顾客所需服务时间是规则顾客所需服务时间的10倍，他们将不会影响到规则顾客的等候时间。尽管这增加了不规则顾客的等候时间，但所有顾客的平均等候时间大大减少了。

8. 调整分开序列的模型，使得类型A顾客可以换柜台。这就是说，当为类型B顾客服务的柜台上没有顾客时，类型A顾客序列中最前面的顾客可以到类型B顾客的柜台上接受服务。当然，你也可以试试，两种类型的顾客都能交换柜台，在另一个柜台没有顾客的时候。当另外一个柜台没有顾客时，两种类型的顾客都能去没有顾客的那个柜台接受服务。

这又将对顾客的平均等候时间产生什么影响？

9 操作员

这一章我们主要讲到操作员。操作员是很多处理场所都需要的能力来源。虽然，操作员一般来说是一个人，但也可以是一台设备。我们以一个例子开始，在这个例子中我们需要用到操作员，但可以忽视机器之间上下移动的时间。随后，我们将用不同方式在Enterprise Dynamics中模拟距离和移动过程。然后，这个例子还将继续延伸，操作员将在方格和图中的路线内自由移动。

9.1 从操作员开始

例 5: *Red (红) 和Blue (蓝) 的操作员*

两个半成品，HalfBlue (半蓝) 和 HalfRed (半红)，到达两条相同但分开的生产线。每条生产线由一台钻床及钻床前面的储藏设备构成。产品经过钻床后，由一台10米长的输送机输送到一台数控机床，再从数控机床上经过一台5米长的输送机送去最后的质检。红和蓝两种产品都按到达顺序依次接受质检，然后离开系统。半成品之间互不相连，经平均1小时的中间到达时间到达。产品在钻床上平均需要20分钟，质检平均需要6分钟。所有时间呈指数分布。另外，数控机床持续工作，每次处理时间为10分钟。

操作员既控制最后的质检，也控制两种产品的钻孔过程，这就是说，操作员需要在钻床和最后的质检中来回移动。现在我们假定移动时间是可以忽略的。

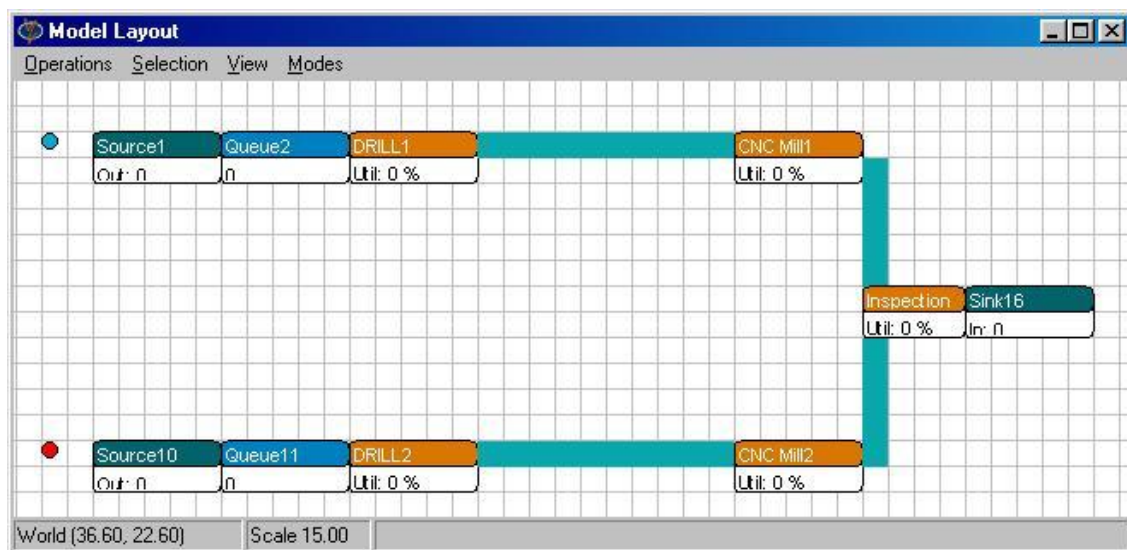


图 9-1 半成品的生产线

图9-1即这个生产系统在Enterprise Dynamics中的分布。
赶紧自己来建立一个系统!

问题及任务

- 1 (理论上来说) 操作员利用率有多高?
- 2 确认钻床, 数控机床及质检(场所)的利用率分别是33%, 16% 及20%。

让模型运行一段较长的时间, 并检查结果, 你得到的结果应和上面的结果差不多!

钻床及质检的场所都用Servers(服务器)来代替, 因为这些操作都需要一定时间。但是, 仍然存在不足: 这一分布是对的, 但是你要留一个人在两台钻床和质检之间来回走动。

所以最好是将模型建立在执行操作的具体地点, 如此例中的Server(服务器)。当一个产品到达钻床或质检场所时, 就会发出讯号, 需要操作员的辅助。只有当操作员到达时才能开始操作。若不再需要操作员的辅助了, 他就可以离开了。在Enterprise Dynamics中这个问题通过Operator(操作员)及 Team(团队)原子解决。每个Operator(操作员)都是Team(团队)的一部分, 一个Team(团队)可以有一个或多个Operator(操作员)组成。

现在我们将一个Operator(操作员)和一个Team(团队)原子拖进模型。这两个原子都能在Operators下的库中找到。右键点击Operator(操作员)就能打开(图9-2)的窗口, 其中有3个选项:

Edit Operator (编辑Operator)
Connect to Team (连接到Team)
Connect to Network (连接到网络)

选择Connect to Team(连接到Team)选项, 然后选择Team原子。现在Operator(操作员)就是Team(团队)的一部分了。另外, Team(团队)的名字也从Team23改成了Team。

这一步虽然不是必要的, 但是对以后的参考却很有用。

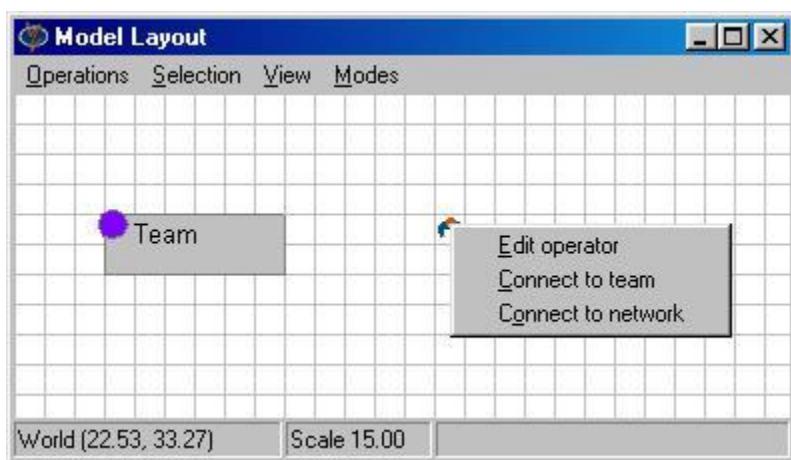


图 9-2 Team(团队)与 Operator(操作员)

顺便提一下，两个原子在模型分布中的位置是没有联系的……

我们仍然需要设定在质检和钻孔时通知Operator（操作员）。由于这在一个产品到达时是必要的，那么必须要在Trigger（）上设置这一选项。在每个服务器原子上的默认列表为

Trigger on Entry（开始进入）:calloperators(atombyname([Team],model),1)

Trigger on Exit（开始退出）: freeoperators(atombyname([Team],model),i)

AtomByName 在模型中搜索相应名字的团队原子，所以如果使用了其它名字就可能需要更改。*calloperators* 1就是代表召唤Operator（操作员）（系统默认），用户可以更改这个数字。*freeoperators*代表Trigger on Exit（开始退出）让所有与产品相关的Operators（涉及到的原子）离开。

由于这两个命令起着很大的作用，完整的句法如下所列：

calloperators(e1,e2,{e3,...,e24})

这项命令包含24个变量，其中变量3-24是可以任选的。以下代表：

e1 = 指代团队
e2 = 一项任务所需的操作员人数
e3 = 一项任务的优先权，数值越大，代表越优先

e4..e24 = 所需操作员的名字

例: calloperators(atombyname([Team],model),2,1,[John]) 要求2名Operators（操作员），优先权为1，其中一名必须是John。如果有两个以上名字，ED从具体的小名单中选取两个名字。操作员被指定到写着这个调用语句的原子的第一个产品上，这个原子上的所有运行都会推迟到所有操作员都到达后才开始。

freeoperators(e1,e2)

e1 = 指代团队
e2 = 指代执行任务的原子

例: freeoperators(in(1,c),i) 使与团队的第一个输入通道相连的原子的第一个产品的所有操作员自由。

在使用指南模型中打开operator1.mod，或在你自己的模型中正确地调整服务器原子。运行模型，确保Operator（操作员）不断从一台钻床移动到另一台钻床或到质检区栏。处理那些想要开始操作但还没有可以支配的操作员的场所，用红灯显示！

按照操作员的到达顺序处理他们的要求。可解决模型僵住等问题，比如钻床中的产品无法离开。为了研究这种情况，可将仿真运行一段较长的时间。我们将在以后再来学习如何预防这种僵局。

可在实验向导中建立Operator（操作员）的利用率，这会超过80%（见Busy状况）。这是否与你自己的计算结果一致？

总之，看上去这个模型可以很好地处理流动量，但是Operator（操作员）将会很辛苦，即使（到目前为止）走动的时间还没有算在内。‘走动’将会在下个章节中讲到……

9.2 在模型的维度内移动

Operator（操作员）可以以三种方式移动

- a. 重新定位而不花时间(系统默认)
- b. 走动
- c. 通过网络走动

当走动时间和运输时间在问题中可以忽略时，我们使用第一个选项。因为两地之间有一定距离，我们引进了运输时间，就不必考虑具体路线，所以我们这时就使用第二个选项。若Operator（操作员）在网络通道及连接点之间移动，我们就选用第三个选项。可用于例如在仓库过道接收命令或叉车和自动导向车等输送车辆，这些都在第10章Transporters（输送员）中进行模拟。

从选项一到选项三显然增加了移动行为的复杂性。所以，作为仿真模型的设计人员，始终要注意输送时间及运输网络是否与问题的框架相关联！

在对Operator（操作员）的移动进行更详细地研究之前，我们要先了解一些Enterprise Dynamics中距离与维度使用的信息。除了例4中的输送机，这本使用指南里的样本模型都还没有涉及到物理维度的使用。那么，若地点，大小及两地之间的距离对模型确实有影响，又该怎么解决呢？

图9-3中，除了一些其它原子，我们可以看到网格——在原子后面总是可见，我们随后将会处理。网格的作用相当于坐标系，每个方格大小为1*1m。左上角的黑色方格相当于坐标（0，0），这就是说Enterprise Dynamics把东北角作为一个原子的位置的参考点。通过在Enterprise Dynamics中移动光标来感受一下这个坐标系统！

在图9-3中，我们看到在一个三角形中有三个点，N1-1，N1-2 和N1-3。
查证N1-2点的坐标为（5，6）。

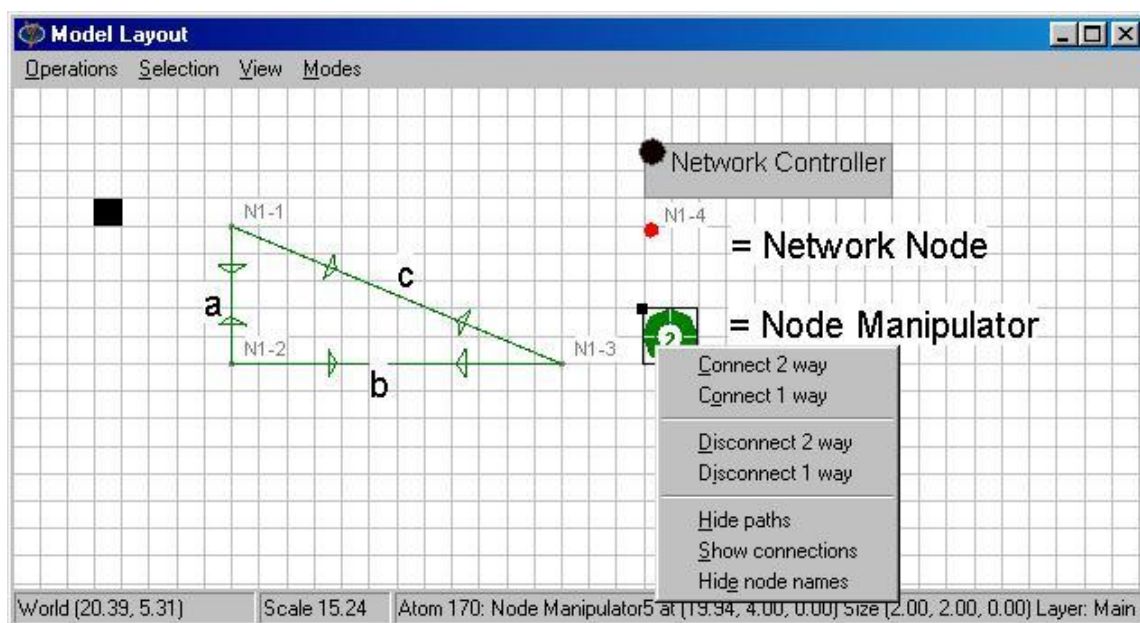


图 9-3 坐标系，距离及几个网络原子

N1-1与N1-2 之间的距离是5米，所以，如果一位Operator（操作员）在这两点之间以默认速度1m/s移动，将需要5秒钟。

根据Pythagoras的距离理论来算c的距离: $c^2 = a^2 + b^2$ 。得出 $c = 13$!

9.3 走动（自由运动）

若对Operator（操作员）来说，距离显得较重要，我们将默认设置`relocate without time expense（重新定位而不化时间）`改成`walk（走动）`。

右键点击Operator（操作员）选择(如图9-2) Edit Operator，就会出现输入窗口(如图9-4).

通过点击Walk选项，Operator（操作员）就会以用户自定义的速度（默认速度为1m/s）在目前位置和下一个位置之间移动。所应用的距离公式是Pythagoras公式，也就是说，两点之间的距离是直线距离，不考虑两点之间的任何物体。这就是所谓的‘自由范围’运动/自由运动。

注意!

原子大小不一。起决定作用的坐标系总是位于原子左上角！

Operator（操作员）原子的其他领栏将在稍后进行详细讨论。

图 9-4 Operator的输入窗口

(续) 例 5: : 红和蓝的Operator (操作员)

现在调整operator1.mod以使Operator (操作员) 按以上所述行走, 或直接打开operator2.mod。运行模型并注意:

1. Operator (操作员) 在屏幕上倾斜地从一台钻床移动到质检点, 反之亦然。
2. Operator (操作员) 在Server (服务器) 上的直观显示与之前模型中的有所区别。

计算并确认在模型中上述的重新定位所需时间不超过21秒(准确地说, 是 $\sqrt{436}$ 秒)。

Operator (操作员) 的利用率也会上升: 在一个小时内, 平均花在钻床上的时间占了40分钟, 质检时间占了12分钟, 还有1到2分钟的走动时间。这可在实验向导中确认……

9.4 网络一般概况

在前面几节中我们已经看到了直线内的自由范围运动。但是由于墙壁或其他障碍物，不可能所有的都是自由范围运动，因此更多的是通过映射路径的运动。

因此，在Enterprise Dynamics中引进了网络这一概念。网络就是很多点和线的集合。可以将Operator（操作员）或Transporter（运输员）连接在这样一个网络上，一旦连接，他们将只能通过这个网络移动。这些点形成网络中的交叉点，线代表路径。这些路径既可以是单向的，也可以是双向的。

为了创建并嵌入一个网络，我们需要执行以下几个步骤：

- a. 将一系列**Network Nodes（网络节点）**及一个**Node Manipulator（节点操控器）**拖入模型分布中。这两个原子都能在Transport目录下找到，并在图9-3中显示。
- b. 连接两个选定的点可通过
 - i. 将Node Manipulator（节点操控器）置于第一个点上
 - ii. 右击（Node Manipulator完全变绿）
 - iii. 将Node Manipulator（节点操控器）置于第二个点上
 - iv. 再右击在两点之间将出现一条绿色的连接，这根据用户在Node Manipulator（节点操控器）上所做选择，既可是单向的，也可是双向的。
可对你想要创建的所有连接重复此过程。
3. 用同样的方法，将需要用到**Operator（操作员）**的所有原子在网络中与（相邻的）点连接起来。
注意：右击位于原子左上角的Node Manipulator（节点操控器）将会用蓝色的直线显示连接。
4. 通过右击将Operator（操作员）与网络相连接，通过Operator（操作员）上的光标，选择Connect to network（连接到网络）（如图9-2）。
5. 将一个**Network Controller（网络控制器）**拖入模型分布中，右击选择Optimize Network（优化网络）选项。

用粗体显示的名称是新出现的原子，都能在Transport目录下找到。Network Node（网络节点）就不需要再解释了。图9-3显示了Node Manipulator（节点操控器）的所有选项。若要删除连接，可重复步骤2；但是通过Disconnect（断开连接）选项。

若选择Connect 2（连接2）的方式或Connect 1（连接1）的方式，Node Manipulator（节点操控器）会变绿，分别用图例2或1显示。若选择Disconnect 2（断开连接2）的方式或Disconnect 1（断开连接1）的方式，Node Manipulator（节点操控器）会变红，分别用图例2或1显示。

图9-5包含了Network Controller（网络控制器）的所有选项。为了熟悉这些原子，可通过做练习3及重复上述步骤，并参考这些原子的Help（帮助）文件！

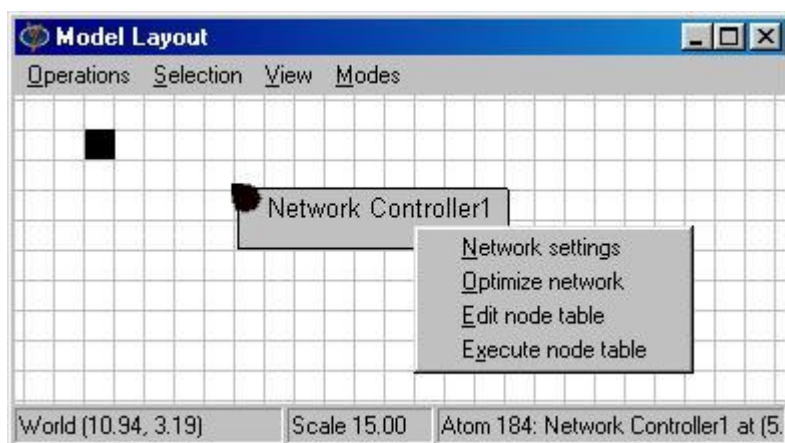


图 9-5 Network Controller（网络控制器）的选项

3 练习

用一个由Source（源），Server（服务器），Conveyor（输送机）及Sink（接收器）组成的新模型进行实验，并通过网络连接Source（源）和Server（服务器），Sink（接收器）置于Conveyor（输送机）后。试着添加一些单向的路径，再将这些连接删除，等等……确保最短的路径不同于返回的路径。通过网络连接Server（服务器）与Conveyor（输送机）。是否出现了蓝色的连接线？添加一个带有Operator（操作员）的Team（团队），如上所述，将Operator（操作员）与网络相连接。最后，通过Network Controller（网络控制器）优化网络。

Operator（操作员）是否很好地在不同路径之间上下移动？

Important重要!

通过Network Controller（网络控制器）优化网络，网络用户总是能选择最短的路径（内部网络应用了Dijkstra运算法则）。这就是为什么在网络中每做一步更改都需要执行这一步骤。

9.5 在网络中行走

（续）例 5：红和蓝的Operator（操作员）

如图9-6，在原来的模型中添加4个点，5个连接。蓝色的线代表网络与需要Operator（操作员）的原子之间的连接。Operator（操作员）的速度仍然是1m/s。

执行章节9-4中的所有步骤并合理地应用到练习3中。

1. 确认Operator（操作员）在网络中行走并且选择两个地点之间最短的路线。注意Operator（操作员）会将自己定位于通向发出命令的原子的交叉点。
2. 确认从质检点走到一台钻床需要23秒，而在两台钻床之间走动需要8秒。这也就是没有计算在蓝色连接上的走动时间！

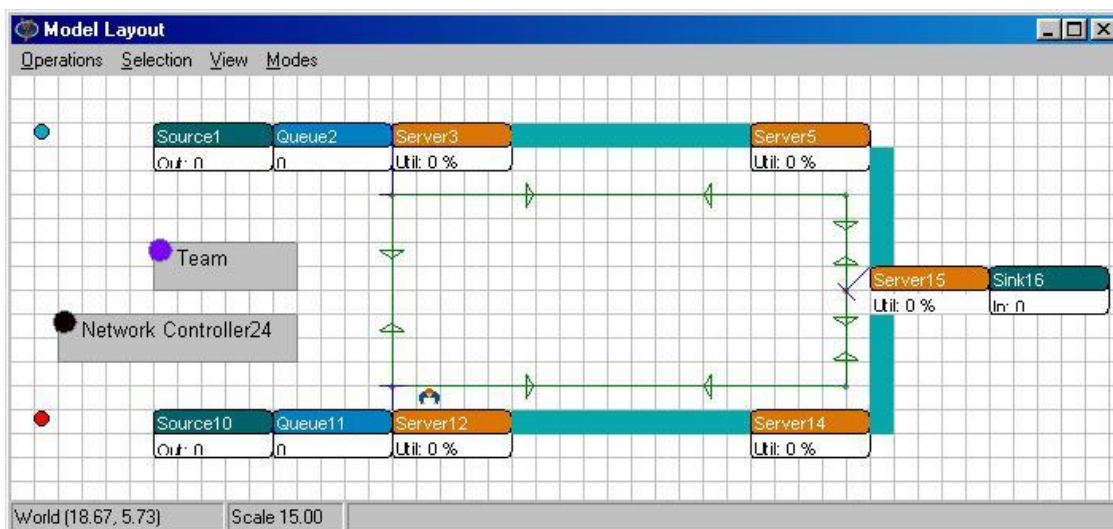


图 9-6 网络中的Operator（操作员）

在operator3.mod下可找到此模型。

9.6 优先权的分配

如果你对以上三个模型中的一个进行长时间的运行，就会观察到输送机偶尔堵塞，使Operator（操作员）不能往输送机上递送产品，其原因是输送机上产品已满。这就是我们所说的**僵局**：由于一项操作无法完成而导致系统暂停。

这种情况可能会在这里出现：由于默认的设置，Operator（操作员）会根据先进先出原则处理所收到的所有命令。假设在钻床和质检点前都有序列，Operator（操作员）在第100个时间单位时完成辅助质检工作，而钻床1和2上都有一个产品‘等待着’Operator（操作员）：这一点可明显地在视觉上显示出来，因为原子呈现红色。让我们进一步假设命令在第90个单位时间到达钻床1，在第95个单位时间到达钻床2。

那么在第100个单位时间时，会以什么顺序执行命令？首先，质检命令完成（Operator可离开），然后一个新的产品进来等待质检，当Operator（操作员）查看他的命令表时，发现有三条可见的命令，根据先进先出原则，他将选择最前面[等待时间最长]的命令。确保命令表上有三条可见命令，而不是两条。这在稍后的优先法则上将会非常重要！

所以，若Operator（操作员）要在第100个单位时间时选择时，他的命令表将显示：

命令	来源	命令到达时间
1	钻床 1	90
2	钻床 2	95
3	质检	100

图 9-7 没有优先权时Operator的命令顺序

根据先进先出原则，现在Operator（操作员）先去钻床1，然后去钻床2，然后回到质检点。很明显，在繁忙时段Operator（操作员）需要重复来回走动，所以会导致系统僵局！

现在我们把优先权分配给质检任务。可参考章节9-1中的调用语句。质检时，将Trigger on Entry（开始进入）上的语句：

calloperators(atombyname([Team],model),1)

改成：calloperators(atombyname([Team],model),1,2)

现在来自质检的命令享有优先权2，而来自钻床的命令仍然（默认）享有优先权1。也可在最后的语句上加一个‘1’，但是这里没有必要。如果图9-7的命令按优先权来排列，那么命令到达时，将会出现一张不同的命令表：

命令	来源	命令到达时间	优先权
1	质检	100	2
2	钻床 1	90	1
3	钻床 2	95	1

图 9-8 有优先权时Operator的命令顺序

实际上，这就代表质检这一行的命令会被优先处理。为什么？

若想在实际运行中看到这一现象，请先运行一个没有优先权的操作员模型，当看到模型将要陷入僵局时，停止运行，然后调整质检的优先权，再接着运行。

有优先权的模型可在operator1 with priority.mod等文件名下找到。

9.7 更多关于Operator（操作员）的选项

还有更多关于Operators（操作员）的选项，假设一项任务要求有两位Operators

（操作员），如一起抬一块很重的金属板。一位Operator（操作员）也可以在一个产品上停留更长时间。所有这些情况都会在任务中涉及到。

4 任务

在operator1.mod中增加一位Operator（操作员）并按以下要求操作：

1. 质检时需同时有两位Operators（操作员）
2. 钻孔过程会一直需要一位(任意的) Operator（操作员），但他将一直陪同产品直到产品到达数控机床
3. 与2中相同，只是在这里红Operator（操作员）只处理产品红，而蓝Operator（操作员）只处理产品蓝

在operator4.mod中可找到所做的这三点更改。

10 输送员

在实际操作中，产品的远距离输送会发生在叉车，AGV（自动导向车），前伸式电动堆高机或其它运输工具中。在Enterprise Dynamics中，这些运输工具用Transporter（输送员）或Advanced Transporter（高级输送员）来模拟。

Advanced Transporter（高级输送员）或者可以在方格上自由运动，或者被限定在网络的通道中。这马上让我们想起了上一章提到的Operator（操作员）。事实上，我们用了同样的概念来解释Advanced Transporter（高级输送员）在方格上的自由运动或限定在网络通道中的运动。因此，我们建议你先看完第9章！

但是Advanced Transporter（高级输送员）与Operator（操作员）之间有明显的区别。Operator（操作员）只是自己从一个地方移动到另一个地方，而Advanced Transporter（高级输送员）要输送一个或多个产品，需要在一个地方装货，在另一个地方卸货，有时会赶不上其它人，有时又赶超别人。总之，Advanced Transporter（高级输送员）要复杂得多。因此，在Transport（输送）目录下有两个Transporter（输送员），即Transporter（输送员）和Advanced Transporter（高级输送员），但是只有后者能在网络通道上行使。

在章节10-1中，我们列出了两个Transporter（输送员）的作用，接着在章节10-2中举了一个含有Transporter（输送员）例子。在章节10-3中首先我们将对Advanced Transporter（高级输送员）重复这个例子，然后通过增加一个网络对主体进行扩展。章节10-4详细描述了原理及与Dispatcher（分派员），the Advanced Transporter（高级输送员）和Destinator（指定员）之间的连接。在章节10-5中，我们讨论了Advanced Transporter（高级输送员）的装货和卸货策略，而章节10-6描述了几个Advanced Transporter（高级输送员）的同时使用。

10.1 Transporter(s)（输送员）的作用

Transporters（输送员）由以下作用：

1. 将一个产品从货物聚集地移到目的地。
2. 基于货物聚集地与目的地之间的距离和指定速率（无网络连接的自由运动），可自动确定两地之间的行使时间。（无网络的自由行动）。
3. 需要增加速度时提高加速度，需要停下时减速，并保持自动遵守正确的行使时间。
4. 在采集及递送产品时增加装载及卸载次数。

Advanced Transporter（高级输送员）除以上作用外，还：

5. 能够同时从产品聚集地移动多个产品到目的地；
6. 能够通过网络通道移动，若在一个连接上设定最多可通多少输送工具；
7. 能将产品向上输送，也就是像仓库中一样，将货物放到一定高度的货物架上；
8. 能够设定(x,y,z) 坐标值，指定产品从哪里采集或堆积到什么地方。

10.2 使用Transporter（输送员）

例 6 采集与递送

与例5中的情况相同，这里我们也有两条平行的生产线，产品蓝或红，经过机器加工后，堆在工厂地板上等待叉车。叉车将产品输送到一个距离较远的输送机上，并离开系统。叉车与输送机之间的直线距离大约为15m。两台加工机器之间的距离大约为10m，产品蓝和红经2分钟的平均中间到达时间到达机器。机器需要20秒的平均生产时间。到达时间与生产时间呈指数分布。所有缓冲器的标准容量是10，输送机的标准速度是1m/s。系统分布可见图10-1。

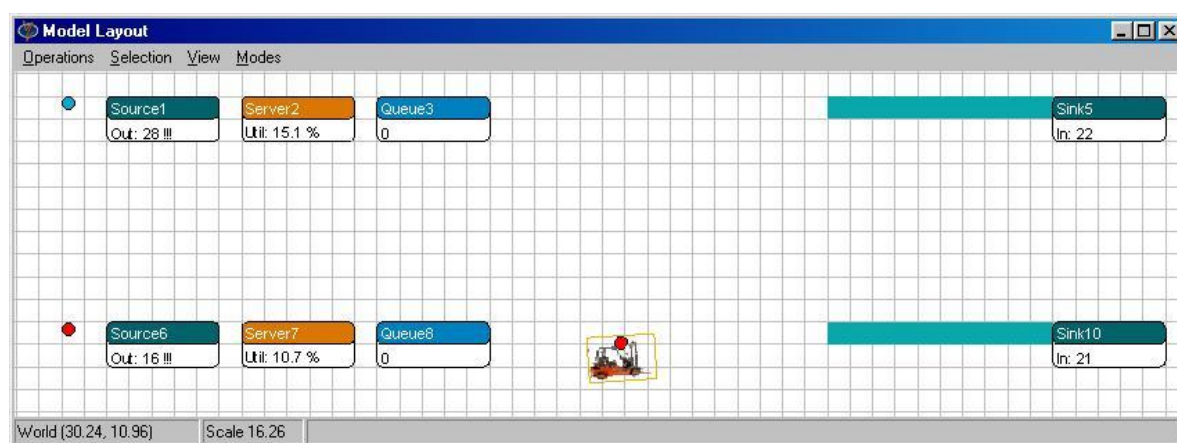


图 10-1 生产线分布

叉车速度为1m/s，在方格上自由运动。首先，我们假定两台输送机上接收到的产品数量是相等的。

问在前面的问题：

1. 理论上叉车的使用率大概有多高？
2. 若采集和放置产品各须5秒钟，叉车的使用率会有多高？
3. 若这两个问题的前提是：产品只被送到最近的输送机，即产品蓝送到较高的输送机，产品红送到较低的输送机，答案又会是什么？

希望这些问题不会太难回答！让我们通过建立一个有Transporter（输送员）的模型来回答这些问题。

从Transport目录下将Transporter（不是Advanced Transporter）拖入模型中。将Transporter的输入通道1或2分别与第一条或第二条生产线上位于机器后面的缓冲器的输出通道连接起来。将Transporter的输出通道1或2分别与第一条或第二条生产线上的输送机的输入通道连接起来。

双击Transporter会打开如图10-2的输入窗口。

默认速度是1 m/s，改变Send To（送到）语句以使产品从两台输送机上被任意选出。与Input Strategy（输入策略）相同，选择Largest Queue（在屏幕上可见）或'the Longest Waiting'。

Input Strategy（输入策略）与Send To（送到）语句在第五章讲到！

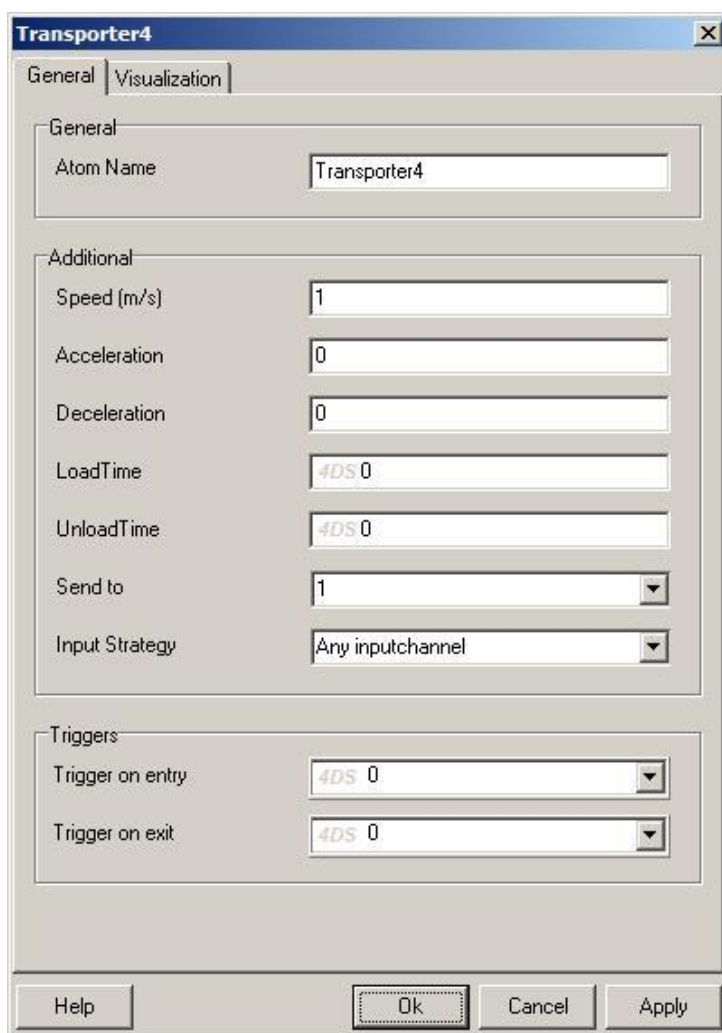


图 10-2 Transporter的输入窗口

现在用你自己的模型或用transporter1.mod进行几次仿真运行。在实验向导中定义58小时的仿真运行，输出变量Status将显示在Transporter上。得出的利用率是否与你自己的计算结果一致？

然后在Load Time（装货时间）和Unload Time（卸货时间）增加5秒。再次运行实验，叉车的利用率将会从55%增加到71%。

为了回答第3个问题，需要调整模型。通过使用一个叫‘Destination（目的地）’的标签，很容易就能做到。对产品蓝，标签的值设为1，而对产品红，标签值设为2，并用到Transporter（输送员）的Send to（送到）语句上来确定正确的输送机。

用你自己的模型或transporter2.mod再进行几次运行。与之前的模型相比较，利用率将会下降几个百分点。不论是对有卸货时间的变量，还是对没有卸货时间的变量，结果都是一样。请解释原因！

10.3 在网络中的（高级）Transporter（输送员）

在章节10-1中我们说过Transporter（输送员）能做的事，Advanced Transporter（高级输送员）都能做到。因此，我们对例6进行扩展，首先在模型1中用Advanced Transporter（高级输送员）代替Transporter（输送员），并将结果进行比较。显然，这些结果都相同！

事实上Advanced Transporter（高级输送员）经常会用到另外两个原子：Dispatcher（分派员）和 Destinator（指定员）。这两个原子都能在Transport目录下找到。Dispatcher（分派员）通过Advanced Transporter（高级输送员）控制产品的采集过程（分派=发送），Destinator（指定员）将产品指定到目的地。图10-3在模型分布中描述了这三个原子。对每个原子使用不同图标就能清晰地区分Advanced Transporter（高级输送员）与Transporter（输送员）！

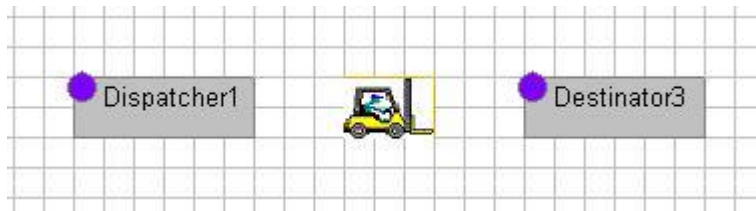


图 10-3 Dispatcher（分派员），Advanced Transporter（高级输送员）及Destinator（指定员）

Dispatcher（分派员）的输入通道与采集场所的输出通道相连，而输出通道又与一个或多个Advanced Transporter（高级输送员）相连。采集过程的推理方法写在Dispatcher（分派员）上。

Destinator（指定员）的输入通道与一个或多个Advanced Transporter(高级输送员)的输出通道相连，而输出通道又与目的地的输入通道相连。除了调整通道，在Destinator（指定员）上基本上不做什么改变。

因此Advanced Transporter（高级输送员）位于中间，还包含了关系到接下来的路线，速度，装货及卸货等参数。

现在可以把Transporter从transporter1.mod中移走，并将上述的三个原子拖进模型中。用适当的方式连接起通道。为了将Advanced Transporter与Transporter进行比较，你还需从产品红和产品蓝的序列中选择最小的序列。

因此，在Dispatcher（分派员）的Sort Tasks By上选择：

3. 最小容量 --> 容器中容量最小的原子最先分派

在General Parameters下点击Advanced Transporter（高级输送员），在Send To（送到）上选择：

5. 按百分比：（50%的产品送到通道1，剩余的送到通道2）

或者选择选项4，可任意选择输出通道。

这一模型可在transporter3.mod的文件名下找到，并如图10-4中所描绘。

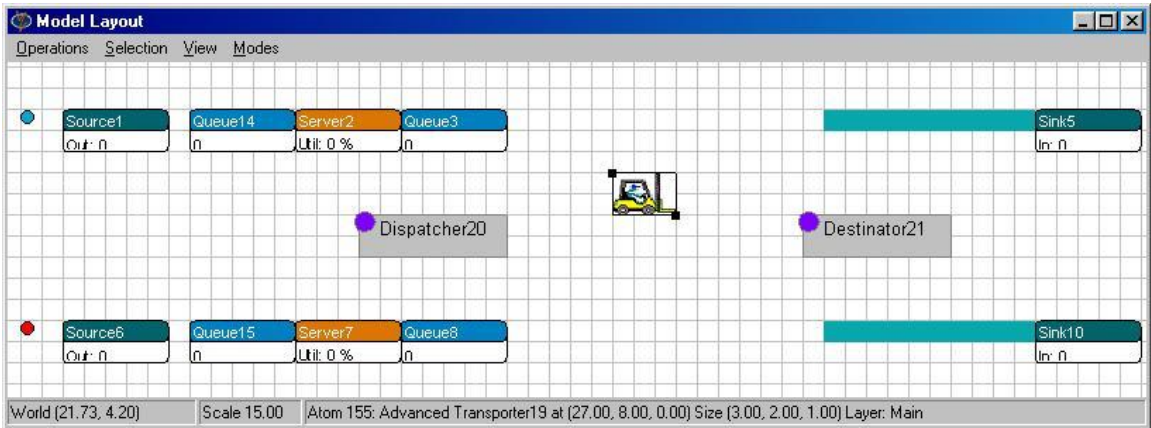


图 10-4 没有网络的Advanced Transporter（高级输送员）

当运行此模型时，我们可通过实验向导发现，Advanced Transporter（高级输送员）的利用率也是55%。

以下的一些行为只属于Advanced Transporter（高级输送员），由于可直接通过上下文看出，我们就不在前面加‘高级’两个字了。

（续）例 6

生产线与输送机被安排在工厂的两个不同的大厅。如图10-5所示，两个大厅通过一条路径相连，在每个大厅内也是通过映射路径移动的。

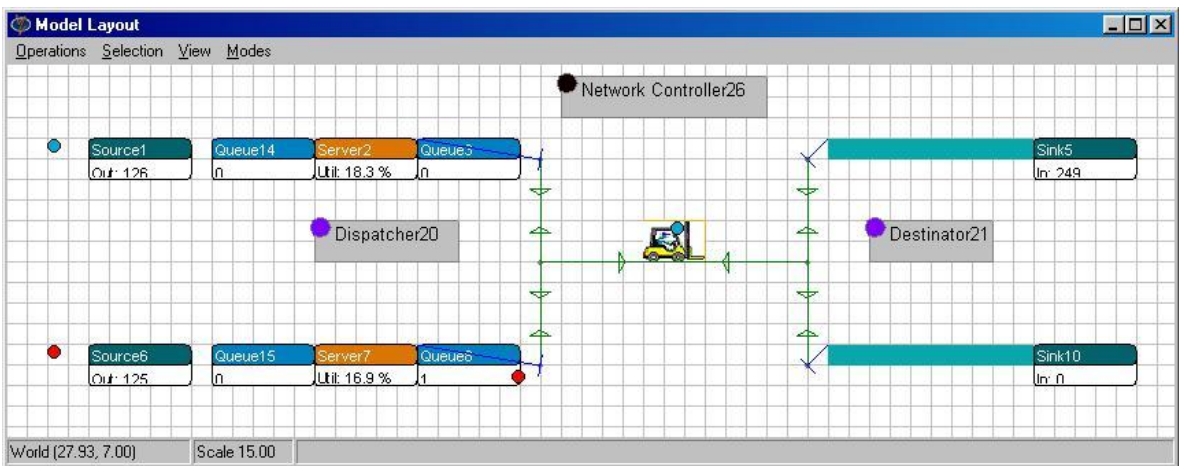


图 10-5 在网络中的Advanced Transporter（高级输送员）

在两条生产线结尾的采集处的两点之间建立了连接，在通向输送机的目的地的两点之间也建立了连接，两个大厅之间的路径的两端也建立了连接。所有的连接都是双向的。

问题及任务

1. 基于transporter3.mod的模型，通过6个点添加以上网络。
2. 此时Transporter（输送员）的利用率多高？
3. 当分别增加5秒的装货及卸货时间时，利用率又是多高？

重要！

始终要通过章节9中的步骤来建立网络，首先要将Transporter连接到网络，然后再优化网络！

首先，检查模型是否按屏幕上所示工作。Transporter（输送员）是否通过沿着连接移动？输送时间是否正确？若不能保证这些正确，就使用transporter4.mod的模型！

如果不出现错误，问题2的答案将是77%的利用率。因为一次上下移动的周期为46秒，平均每分钟有一个产品进入缓冲器：46/60四舍五入就是77%。若装货及卸货分别需要5秒钟时间，周期时间需要再增加10秒：那么利用率就是56/90，即93%。

10.4 连接Dispatcher（分派员），Advanced Transporter（高级输送员）及Destinator（指定员）

例6已向我们展示了这些Transporter（输送员）的很多功能。对于‘一般’Transporter（输送员）功能，我们或多或少已经都讲到了。我们只跳过了加速和减速这一段，但是很容易就能把这些加进模型中——两个Transporter（输送员）都可以——并在屏幕上观察到。另外，我们也展示了Advanced Transporter（高级输送员）在网络中的移动。但是，这个原子的功能还有很多！

首先我们对Dispatcher（分派员）和Destinator（指定员）进行仔细描述。这两者都能与Advanced Transporter（高级输送员）结合起来使用。

10.4.1 Dispatcher（分派员）

Dispatcher（分派员）为与其输入通道相连的原子分配输送任务（没有产品！）。这些任务被分配到与其输出通道相连的Advanced Transporter（高级输送员）上：这里Advanced Transporter（高级输送员）可以不止一个！

一共有9条默认的Dispatch To（分派到）法则：1. Specific channel（具体通道） --> always send to channel 1（始终送到通道1）。

这是与Dispatcher（分派员）的第一条输出通道相连的Transporter（输送员）。

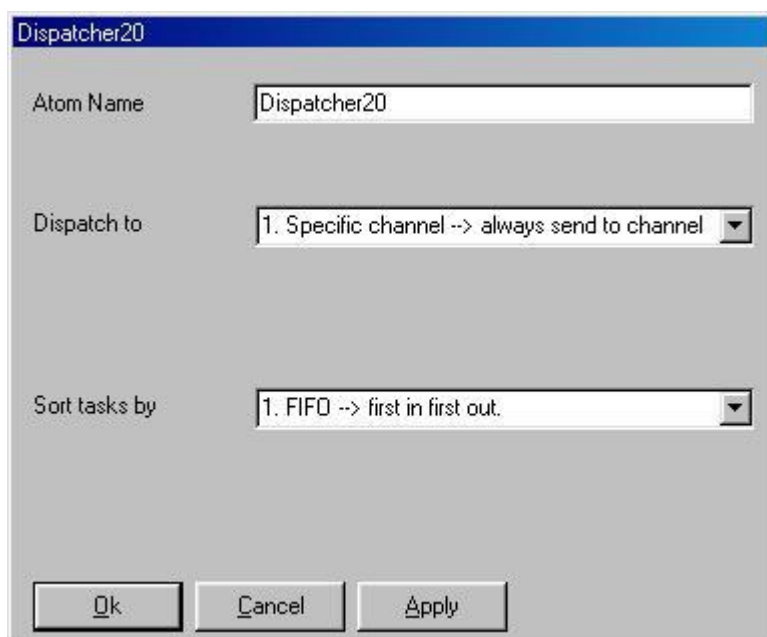


图 10-6 Dispatcher（分派员）的输入窗口

这些输送任务可根据10条预定义的分类法则进行分类('Sort tasks by')，默认设置为先进先出。对所有的预定义分配及分类法则进行了解！

10.4.2 Destinator（指定员）

Destinator（指定员）的输入通道与Advanced Transporters（高级输送员）的一条或多条输出通道相连接。Destinator（指定员）的输出通道与所有可能的目的地相连接。由于所有的递送逻辑都显示在Advanced Transporter（高级输送员）上，Destinator（指定员）窗口上看不到什么（如图10-7）。

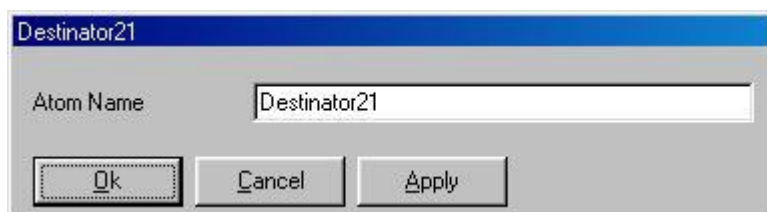


图 10-7 Destinator（指定员）

10.4.3 Advanced Transporter（高级输送员）

Advanced Transporter（高级输送员）的输入窗口中有5张不同的标签表，可设置以下参数：

- General（一般性）参数
- Speed（速度）参数
- Load（装货）参数
- Offset（偏置）参数
- Link to network（连接到网络）

我们将在这里讨论General（一般性）参数及Load（装货）参数。Speed（速度）参数与连接到网络都不需要过多解释。

Offset（偏置）参数将会与Warehouse（仓库）原子一起，在下面的章节中讲到。

General（一般性）参数（如图10-8）中没有增加新的功能。但是，必须要知道，Send To（送到）语句是通过Destinator（指定员）的输出通道执行的。

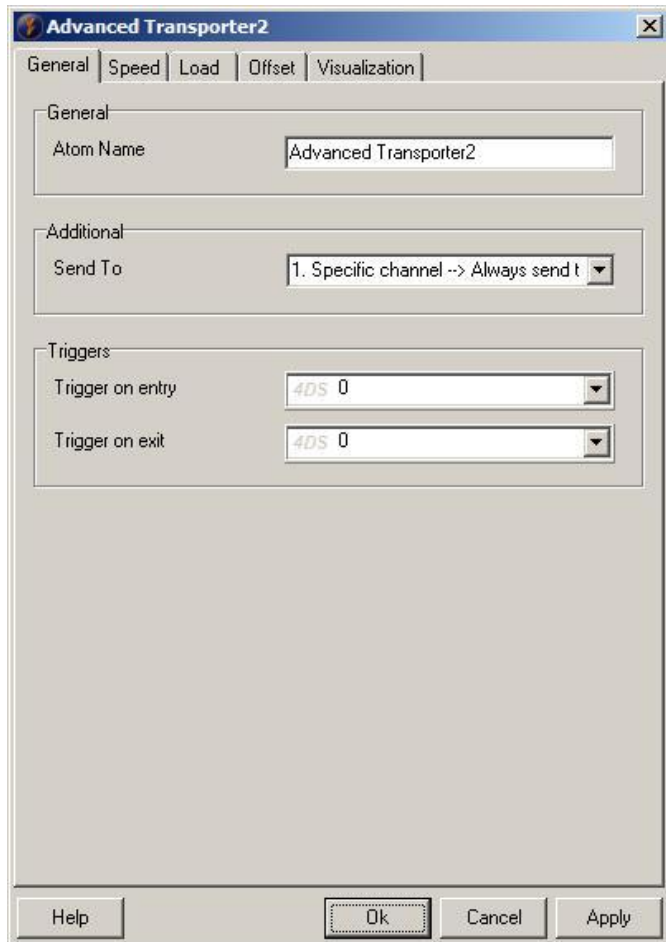


图 10-8 Advanced Transporter（高级输送员）的General（一般性）参数的输入窗口

Load（装货）参数(如图10-9)中提供了很多新的选项。首先，可以调整装货量，只有达到设定的装货量，Transporter（输送员）才会开始输送产品！

若没有设置装货限定，就会按照输送任务的到达顺序装货，也就是说，Transporter（输送员）将在不同装货地之间来回跑动。这里有6个设置选项：

- No Restriction（没有限定）（默认）
- Samename（相同名字）
- Same（相同）
- Same label text（相同标签）
- Samemother（相同来源）
- Same container（相同容器）

我们需要对其中几个选项进行解释。

只有在第一个产品装好后才能激活所限定的装货量。譬如，设定装货量为5，Transporter（输送员）先输送序列10中一个叫做‘radio（收音机）’的产品。

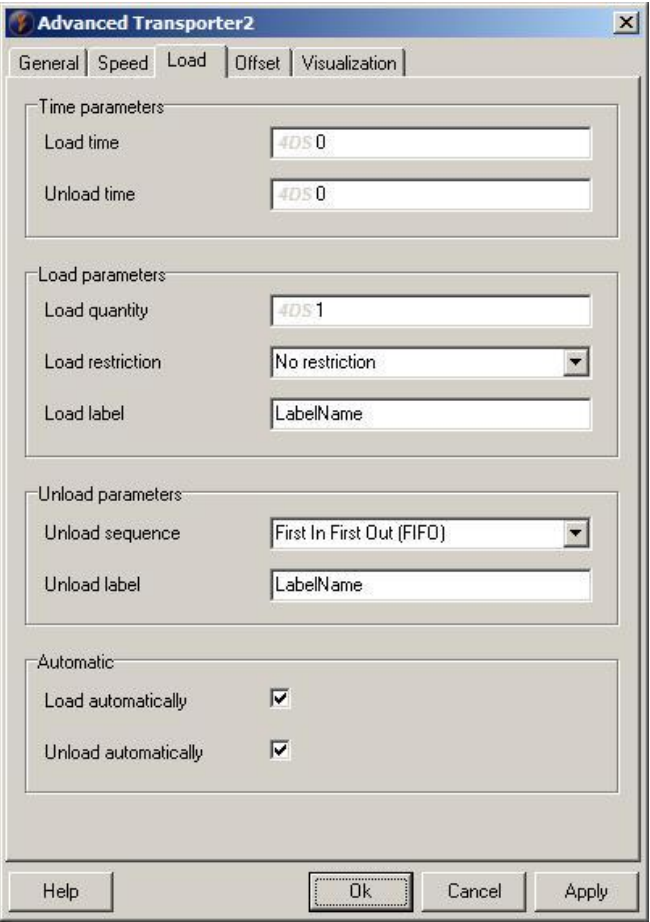


图 10-9 Advanced Transporter（高级输送员）的Load（装货）参数

那么装货限定：

相同名字	就代表着	在所有采集点等待并采集另外4个收音机；
相同容器	就代表着	在序列10中等待并采集另外4个收音机。

在以上两种情况中，必须保证所采集的收音机会出现。所以要注意系统僵局！

10.5 Advanced Transporter（高级输送员）的装货及卸货策略

我们把下面的选项引入我们的问题中：

1. 采集并输送两个产品到输送机中；
2. 采集并输送两个 *相同* 的产品到输送机中；

3. 采集并输送两个相同的产品到一台具体的输送机中：蓝产品到第一台输送机，红产品到第二台输送机。

选项1操作起来很容易：只需将装货量改为2。确认现在Transporter（输送员）不仅输送红、蓝产品，还输送其他产品，根据产品在缓冲器内的到达顺序。这就是说，若一个红产品后面跟着一个蓝产品，Transporter（输送员）只有在红产品到达缓冲器时才会走过去处理红产品。随后，Transporter（输送员）走到蓝产品那里，也许中间需要等待。然后，再到达输送机。注意，最后采集的产品会在Advanced Transporter（高级输送员）的图标中出现。

卸货过程也有很多变量：红和蓝可能被放在同一台输送机上，也可能放在不同输送机上。这还仅仅是选项1，但是思考一下为什么会出现这种情况呢？

关键就在于Send to（送到）语句：一个产品到第一台输送机的机率是50%，到第二台输送机的机率也是50%。显然，在缓冲器内部对每个产品有一种引力，使其到‘首选’的输送机上去。选项1可在transporter5a.mod模型中可见。

对于选项2，我们只需将装货限定从选项1中改成Same Container（相同容器）。这一过程可在模型transporter5b.mod中可见。但是还是会发生无组织地发送。

对于选项3，我们需要回到Destination（目的地）标签，如同章节10-2一样：蓝产品的目的地标签值设为1，红产品的目的地标签值设为2。然后，我们从选项2上把Transporter（输送员）的Send to（送到）语句改成：

7. By label value (direct)（直接通过标签值）：the channel number is written directly on the label named destination of the 1st atom in the queue. If the label value is 0, then send to channel 1.（在序列第一个原子的目的地标签上直接写明通道号码。如果标签值为0，就是送到通道1）

这一方法可在transporter5c.mod模型中找到。

你也可以自己思考一些问题，试着想出解决方法……

10.6 多个Transporters（输送员）的使用

若在一个网络中有几个不同功能的（高级）Transporters（输送员），将会如何工作？如果他们不能赶超别人，又将如何工作？我们再次通过例子来处理这些问题。

（再续）例 6

由于顾客需求量大，两条生产线上的流量都增加了一倍。一个Transporter（输送员）不够用了，这时需要增加一个Transporter（输送员）。两个输送员一次只输送一个产品到任意一台输送机上，而第二个输送员的速度是2m/a，显然比第一个输送员快。若两个输送员同时去采集一个产品，产品将会被快的输送员采集到。

问题及任务

- 7 基于transporter4.mod的模型，在网络中按要求增加另一个输送员，并调整产品供应。别忘了优化网络！
- 8 在自己的模型中观察两个输送员的行为，若发现问题，可打开transporter6.mod的模型。输送员1能否超过输送员2，并不引起任何问题？
- 9 为了再检查一遍，可再次求出两个输送员的平均利用率

在transporter6.mod模型中，我们在网络中增加了另一个高级输送员。为了便于区别，我们将第二个输送员的图标改成‘一般’输送员。当然这个更改不会改变新的高级输送员的功能！

图 10-10 显示了有两个输送员的部分分布。

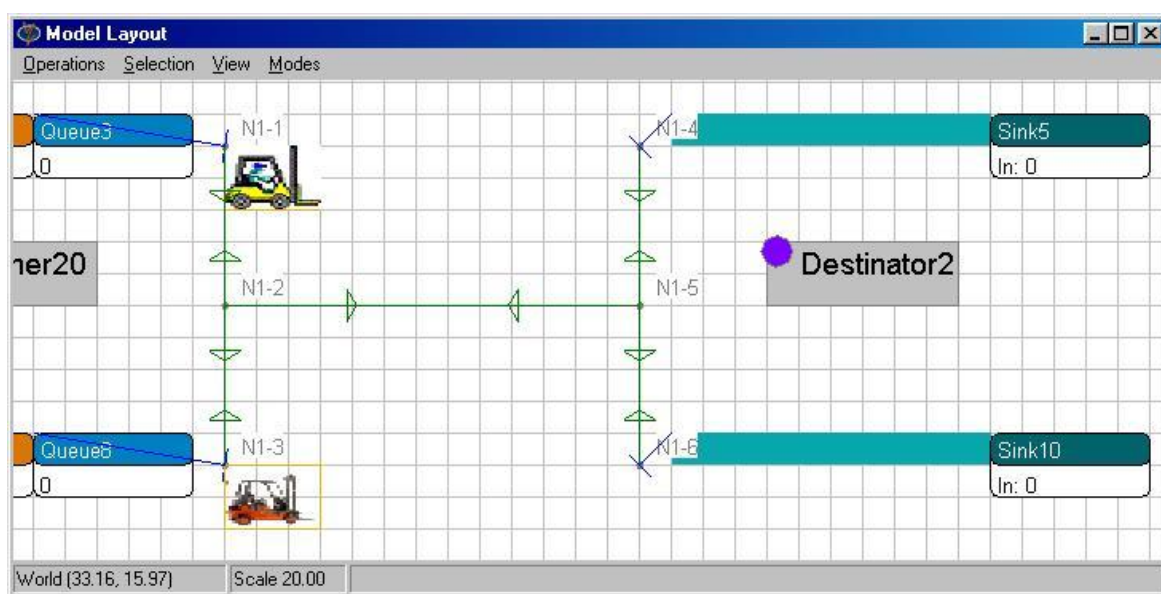


图 10-10 一个网络中有两个输送员

通过Node Manipulator（节点操控器）及Show Nodes（显示节点）选项，可见节点的名称。节点N1-1和N1-3分别被选为黄和红输送员的起点位置。当模型重启时，可看到所做调整。

当两个输送员同时去采集一个产品时，产品被分到快的输送员，这可通过在

Dispatcher（分派员）上将Dispatch to（分派到）改为：

3. A free Transporter (LTF)（空闲的输送员） --> look for a free Transporter. Check the Last Transporter connected to the dispatcher First.（寻找空闲的输送员。将最后一个输送员最先连接到分派员）

这是因为快的输送员与Dispatcher（分派员）的第二条输出通道相连。可通过其他Dispatch To（分派到）法则看看还有哪些分派可能性。

问题9可在实验向导中回答。因为优先权给了输送员2，我们看到输送员整体的利用率为48%（Status Idle 0.52），而输送员1的利用率升高到56%（Status Idle 0.44）。

由于装货的平均分配，理论上结果为57.5%：每分钟2条命令，其中一条由输送员1执行（46秒），另一条由输送员2执行（23秒），而每次装货所需平均时间为34.5秒，而不是60秒。

解释为什么这时的利用率57.5%比给慢输送员优先权时的利用率56%还要高！

（再续）例 6

出于安全考虑，不管往哪个方向，只有一个输送员能使用中间的通道。另一个输送员只能等各自的通道空闲了以后才能输送。所以在这里是不能赶超对方的。

问题及任务

11. 将上面的情况在由两个输送员的网络中操作。

为了回答问题11，我们先来看看如何在部分网络实现容量限定：

在图10-11中，我们看到两个输送员在N1-2到N1-5的连接上移动，而现在必须将这种情况排除。

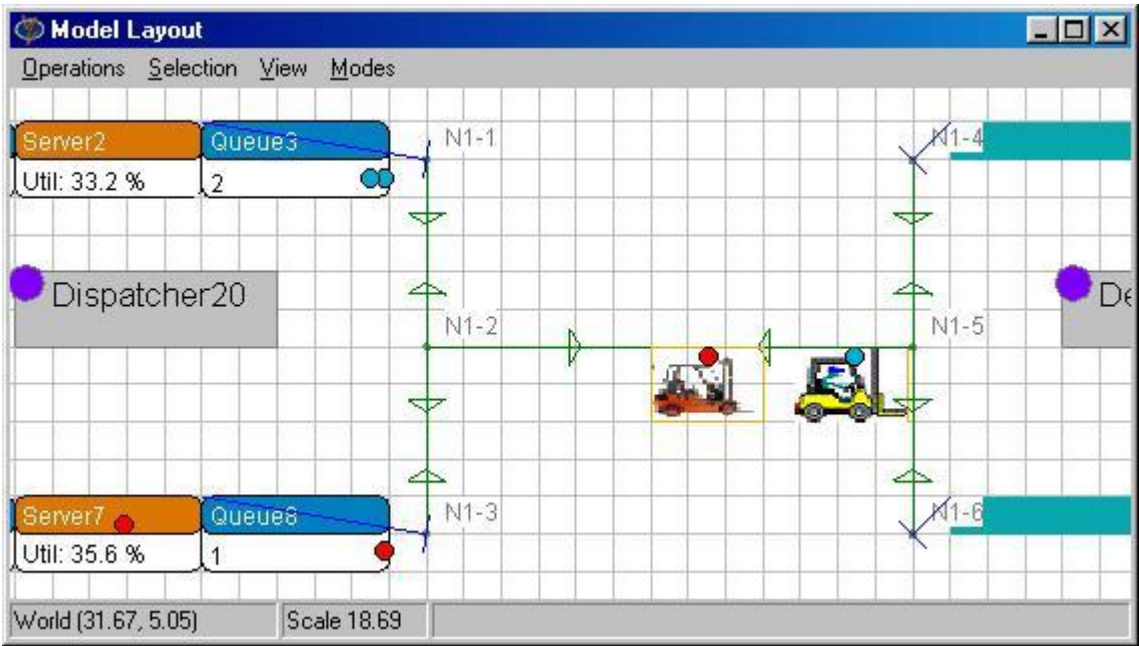


图 10-11 在N1-2到N1-5的连接上有两个输送员

另外，已选中点N1-2，这可从点N1-2有下方的黑色边看出。双击这个点可打开图10-12的表。

Direction	Capacity	Content
N1-1	1000000	0
N1-3	1000000	0
N1-5	1	0

图 10-12 N1-2的网络表

这张表包含了N1-2到N1-1， N1-3及N1-5的三个连接。 *Capacity*（容量）显示了所指连接中允许同时通过的输送工具的数量。默认设置是1000000，或：无限的。*Content*（内容）显示了那一时刻所指连接中正在移动的输送工具。从N1-2到N1-5 content中的2辆输送工具即是图10-11中的快照。

让模型运行一段时间，观察图10-12右手边那一栏的数字一直不停地变化。在网络中的其他点上也可观察此现象。

通过将N1-2到N1-5的容量改成1，问题11就变得简单了。同样在N1-5到N1-2上重复此操作！

现在观察输送员在从N1-2到N1-5的路程上的举止。车辆是否有在等待？注意当达到容量极限时连接会变成红色！

这一模型可在transporter7.mod的文件名下找到。

现在我们将不再解释网络中或没有网络的输送员的应用了。

很明显，我们现在已经能够在仓库环境中建立一些更复杂的模型了。

到现在为止，还没有讲到的就是仓库中的产品架，产品可从这里被采集或被放置。将来，我们会在第11章中与Warehouse（仓库）原子一同讲到。