

上机实验说明

一、概况介绍

《系统建模与仿真》是一门理论与实践并重的课程，不仅需要深入的理论学习，而且需要大量的上机操作练习，从而真正理解和掌握仿真基本原理和实现技术。本课程采用 AutoMod 软件进行实验教学，上机部分包括基本的软件操作练习和几个不同对象系统的建模与仿真实验。限于本课程（SA-3142100860）32 学时的设置，课内上机操作课时也很有限，需要利用课余时间完成实验作业。本学期上机安排如下表所示。

序号	日期	实验内容
1	2019/5/8	AutoMod beginning
2	2019/5/15	ASS-Conveyor
3	2019/5/22	AGV-Pathmover
4	2019/5/29	Inventory system
5	2019/6/5	机动
6	——	大作业，学期内完成

二、实验教学与学习方式

实验课教学包括集中授课与课堂自由练习两部分。集中授课主要是通过讲授和带领操作，熟悉仿真软件基本功能，对实验内容进行初步分析和建模尝试。自由练习则侧重对操作内容的重复巩固和自由探索。实验教学以 AutoMod 教育版为基础，教学内容均可在教育版范围内完成；部分有自主开拓意愿的学生可以申请使用高校网络正版软件。

软件学习与应用的关键是要用，仅仅靠上机课堂的有限学时很难达到熟练程度，还需课下通过完成实验作业和自由练习来达到熟练的目的。基于历届教学和学生学习经验及教训，提出如下几点学习建议，希望有所帮助。

- 从大轮廓到细节进行学习；
- 及时提问做好笔记；
- 养成经常查看用户手册和帮助文档的习惯；
- 同学之间经常交流和共同学习。

三、实验报告要求

每位同学应独立完成自己的实验任务。完成实验作业过程中，可以参考教材、软件使用手册、在线帮助、教师下发的阅读资料，以及任何可以得到的帮助资料，也可以和同学讨论交流，可以向助教或任课老师答疑，但最终提交的实验作业必须是自己独立完成的。

实验模型与实验报告作为实验作业应一起提交。作业以电子文档模式提交。每人提交一个文件夹，以“学号（已含班级信息）_姓名_实验名称”方式命名。文件夹中应包括模型文件夹和 word 格式的实验报告书。**特别提示：不要随意改变模型文件夹和文件的名称。**

实验报告应包括如下内容：

- (1) 实验名称、实验课日期、完成日期、班级、学号、姓名；
- (2) 实验目的、实验对象系统描述、实验内容与仿真目的（目标）；
- (3) 建模思路与模型简介；
- (4) 完成的作业内容与对思考题的回答。

说明：自选内容内容可自行选做，不属于必须完成的内容。

实验报告书参考格式：

《系统建模与仿真》实验报告

实验名称：

实验课日期： 完成日期：

完成人信息： 班级： 学号： 姓名：

- 实验目的
- 实验对象系统描述与仿真目的（目标）
- 建模思路与模型简介
- 仿真运行情况、仿真输出结果与仿真结论【有的实验有，有的实验没有】
- 作业与思考题的回答
- 其它（如心得、问题、建议等）

实验一、单服务台银行排队系统建模与仿真

1. 实验目的

通过对一个简单排队系统的建模与仿真运行，学习 AutoMod 软件的基本操作，包括基本建模元素的使用、逻辑系统的建模方法、基础语法、**仿真运行控制**和查看 AutoMod 的默认仿真输出结果。在初次学习仿真软件的同时，认识排队系统的概念与特性，体会排队系统仿真的基本原理，深刻理解离散事件系统仿真的基本概念。

2. 实验内容

(1) 对象系统描述

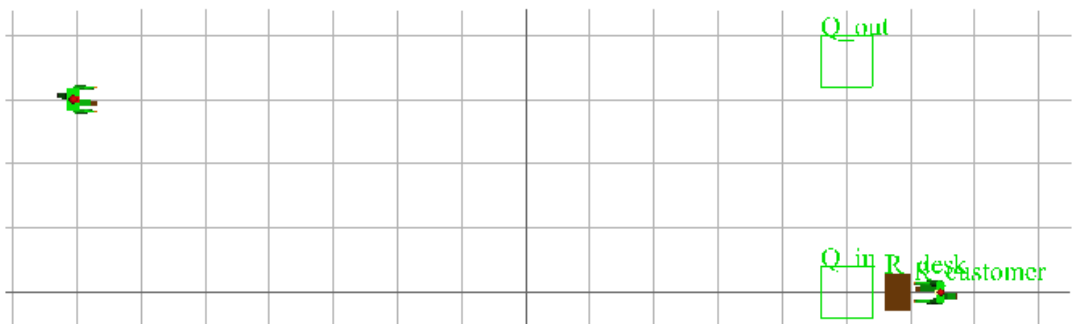
- ①只有一个服务窗口；
- ②每天 8 小时工作制；
- ③顾客接受服务原则为先到先服务；
- ④所有到达的顾客都在窗口前排队等待；
- ⑤顾客到达是随机的，每两个先后到达的顾客之间到达时间间隔是不确定的，顾客的到达时间间隔服从参数为 5（单位：min）的指数分布，第一个到达的顾客为系统默认值；
- ⑥每个顾客的服务时间也是随机的，银行工作人员服务时间服从均值为 5（单位：min）的均匀分布。
- ⑦顾客经工作人员办理业务后，离开系统。

(2) 内容

- ①完成该对象系统的 AutoMod 建模；
- ②系统中队列、服务台等实体大小和平面布置自定；
- ③调整顾客到达时间间隔和服务时间的函数分布，观察系统的变化。

3. 实验步骤

- (1) 新建一个模型（自动创建一个 process system）；
- (2) 定义服务台、队列等实体单元，按平面布置示意图布置相应的实体位置；
- (3) 定义系统参数和实体参数；
- (4) 编辑 source file 文件，编写相应代码，定义系统的流程逻辑；
- (5) 设定运行时间；
- (6) 运行模型；
- (7) 查看仿真动画和仿真输出结果；



作业与思考题

- (1) 系统是否平衡？哪些参数可以有效反映系统服务能力的平衡状况？如何有效识别不平衡系统的瓶颈？
 - (2) 将顾客到达时间间隔与服务台服务时间两个随机变量分别修改为：
 - ①都为定长值 5min；
 - ②到达时间间隔仍为 expo5，服务台服务时间的均值从 5 按照 0.1 的步长逐渐减少到 4；
 比较上述实验结果，有何发现和认识？
 - (3) 谈谈你完成本次实验的心得体会与建议。
- * 你是如何发现该系统的瓶颈的？哪些参数可以有效反映系统服务能力平衡状况？

实验一详细解析

本实验主要是熟悉 AutoMod 建模步骤；学习逻辑系统的建模方法；学习查看 AutoMod 的仿真结果。通过建立仿真模型的实际操作认识仿真的基本概念。

本实验的程序逻辑并不困难，学习的重点在于 Load、Resource、Queue 等常见实体的定义及其三维显示的设定。通过该实验能体会 AutoMod 建模的具体步骤，为以后建模良好习惯打下基础。以下是完成本系统模型所需要的项目。

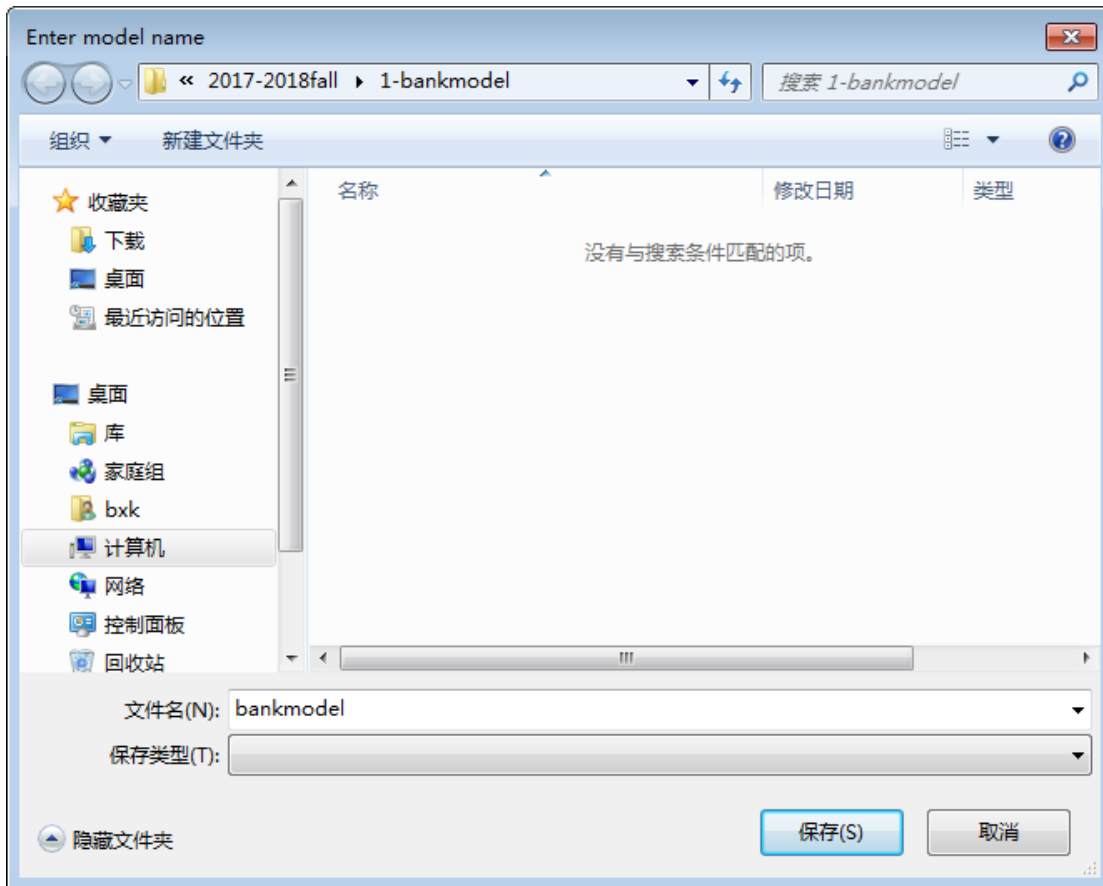
表 1-1 单服务台银行排队系统 Process 系统

Process系统		
项目	名称	备注
Process	P 1	顾客接受服务后离开
Loads	L customer	顾客，到达服从参数为5的指数分布
Resources	R worker	银行工作人员
	R desk	工作台
Queues	Q in	顾客等待服务时所在队列
	Q out	顾客离开系统时所在队列
指令	begin/end arriving procedure, send to die, wait for, move into, use Resource...	

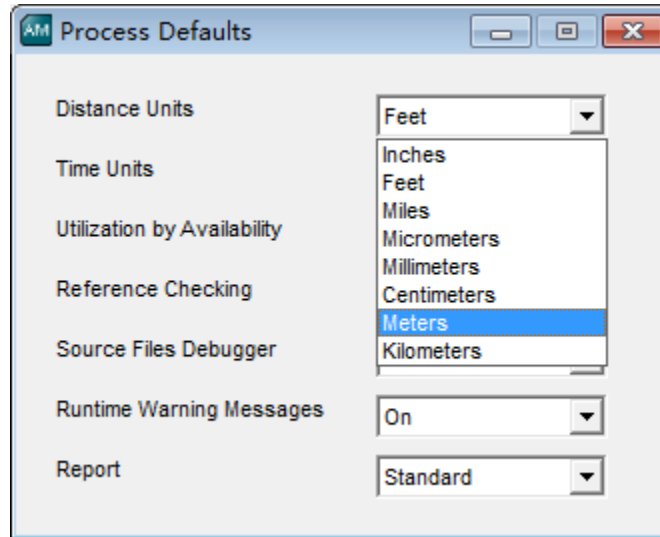
1.新建一个模型

Step1 从编辑界面的菜单选项 File 中点击 new 选项。

Step2 在工作目录下建立一个新的文件夹，如 G:\fangzhenmoxing\2017-2018fall\1-bankmodel，以后以便有关模型 1-bankmodel 的所有资料都在此目录下放置，输入模型的名字 bankmodel，点击保存即可完成模型的创建，此时在目录 G:\fangzhenmoxing\2017-2018fall\1-bankmodel 下生成一个 bankmodel.dir 的文件夹。（注：模型所在目录一定不能含中文）



Step3 点击 System 中的 Default 选项，在 Distance Units 中的选项中选择 Meters。



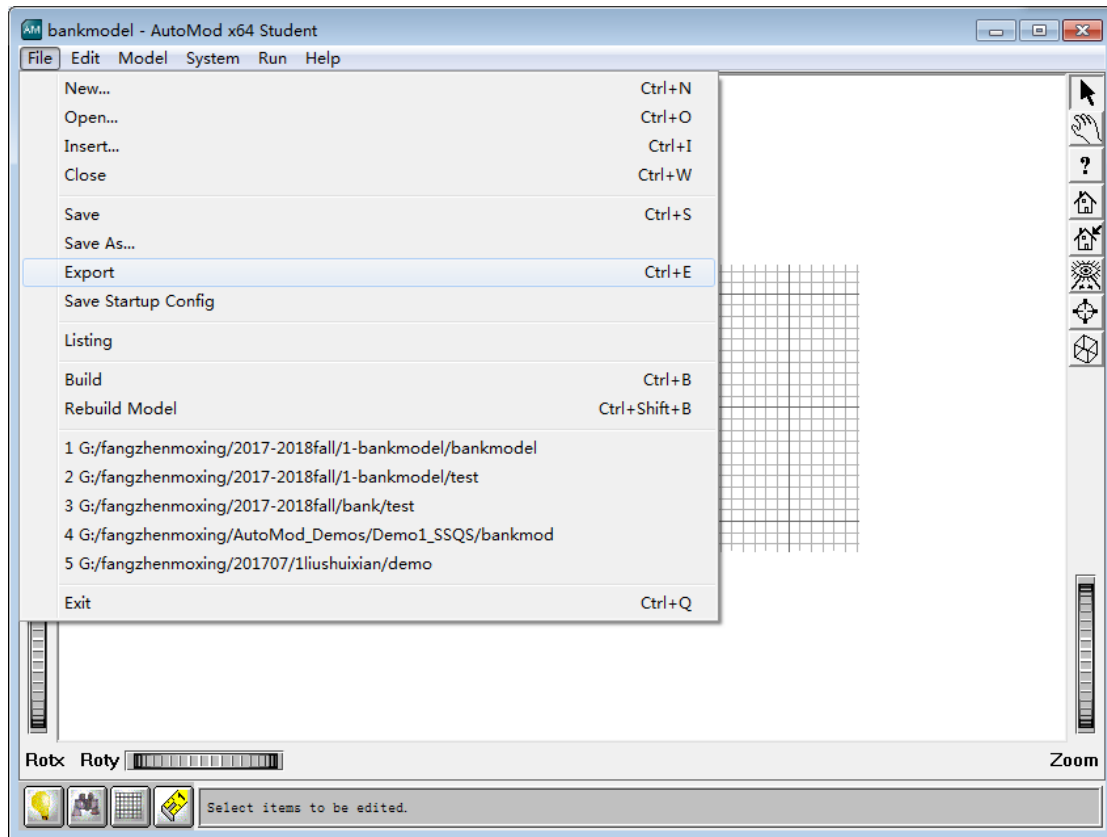
Process Defaults 视窗

Step4 模型的保存：在 AutoMod 中有两种方法可以保存一个模型。

方法一：点击菜单 **File** 下的 **Export**，生成 **arc** 文件夹，可视为模型的备份，该文件为 ASCII 码格式，相对 **dir** 文件夹占内存小，可以送到其它机器上执行。所以你在对模型作更新时要记得做 **Export** 动作，以更新文件内容。

方法二：点击菜单 **File** 下的 **Save**，此动作仅更新 **dir** 文件夹，因此你想修改模型，但又不确定改变是否要留下来时可以采取此操作。若模型运行过后不想保留这些修改，可以重新读取 **arc** 文件夹，恢复到未修改前状态。

注：一般，对模型完成确定的修改时，一定要随时 **Save** 和 **Export**。

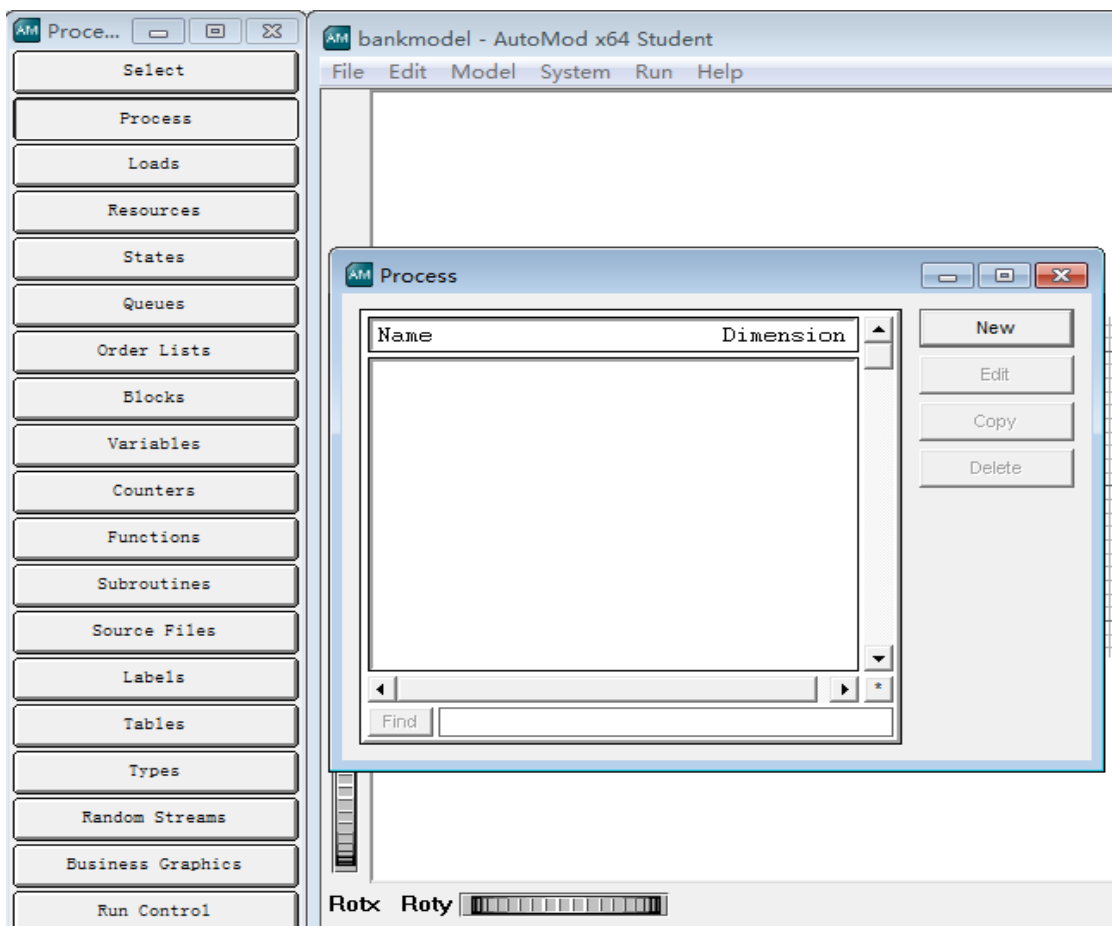


Step5 模型的关闭。模型在编辑保存完后，可点击菜单 File 下的 Exit 或编辑界面的右上角的关闭按钮退出模型。

2.定义各种实体单元

2-1 定义 Process

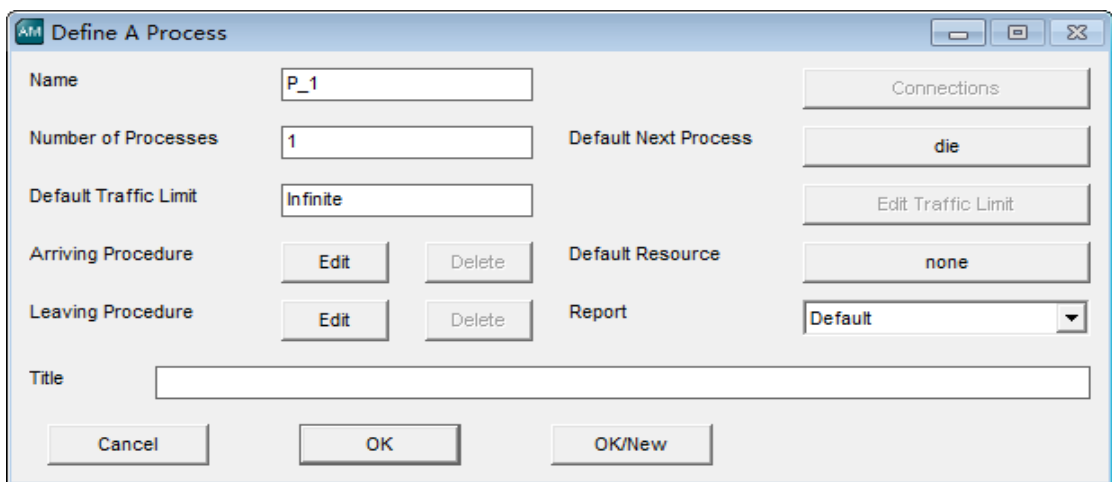
Step1 鼠标点击 Process System 的工具箱中的 Process，打开 Process 视窗。



Process 系统

Process 视窗

Step2 点击 New 选项定义一个新的 Process，即可打开 Define A Process 视窗。Name 的文本框中输入 P_1， Number of Process 的文本框中输入 1， 点击 OK 就完成了 P_1 的定义。

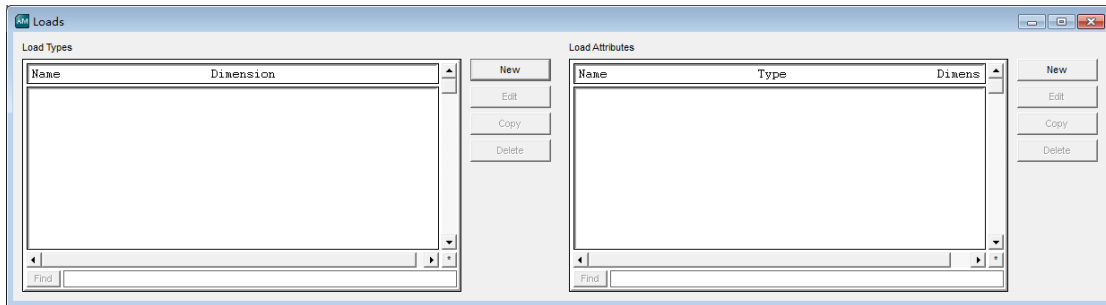


Define A Process 视窗

Step3 如果 Process 已经存在,点击 Process 视窗中的 Edit 选项,即可打开 Edit A Process 视窗，修改设置。

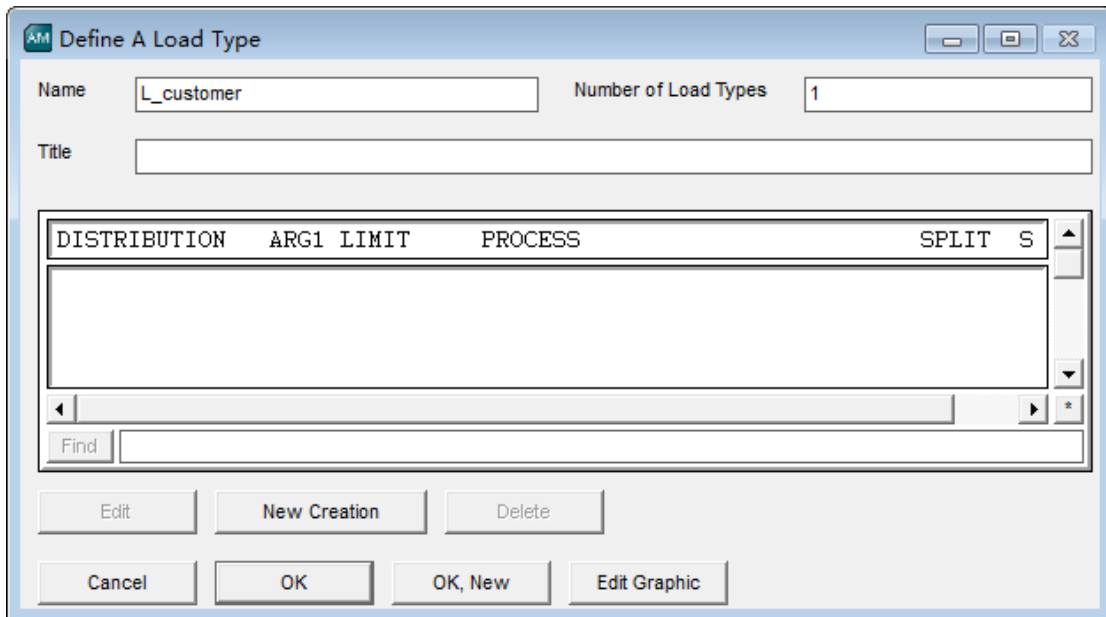
2-2 定义 Loads

Step1 点击 Process System 的工具箱中的 Loads，打开 Loads 视窗。



Loads 视窗

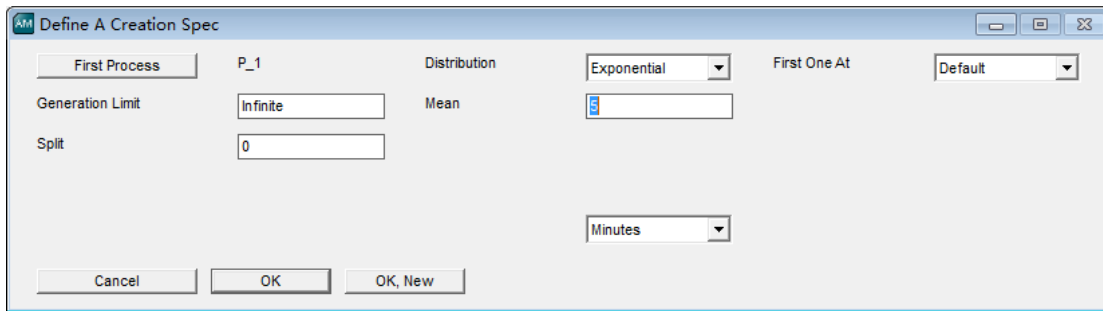
Step2 在 Loads 视窗里点击 New 选项定义一个新的 Load，即可打开 Define A Load Type 视窗。在 Name 的文本框中输入 L_customer 即可完成定义。



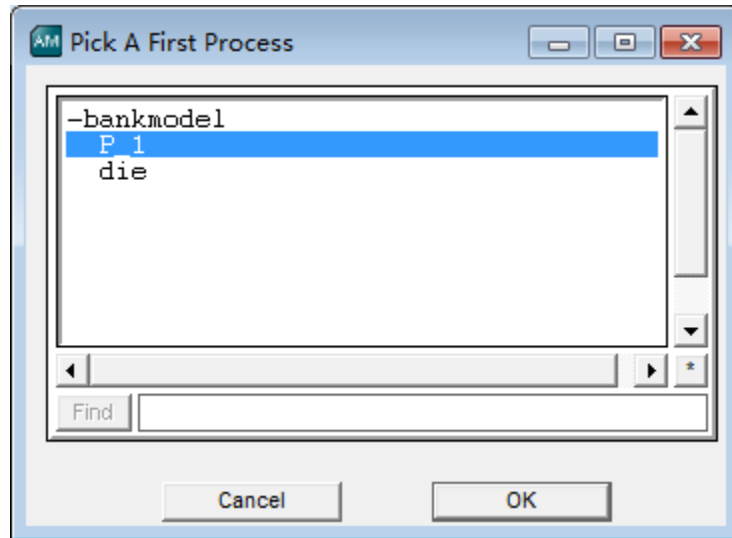
Define A Load Type 视窗

Step3 由于 L_customer 为本模型中最先进入系统的 Load，因此还需定义 L_customer 产生的频率和第一个 Load 进入的时间。点击 Define A Load Type 视窗中的 New Creation，即打开 Define A Creation Spec 视窗。顾客到达服从以参数为 5min 的 Exponential 分布，它的第一个处理的 Process 为 P_1。因此可以做出如下图中的设定。即代表第一个 顾客（First One At）在 Default 的时间进入系统，之后的 L_a 进入系统的时间分布为

（Distribution）为参数为 5 的 指数分布，并且会产生无穷多个顾客（Generation Limit=Infinite）。点击 First Process 进入 Pick A First Process 视窗，选择 P_1 作为 第一个 Process。最后点击 OK 就完成了 L_customer 的定义。



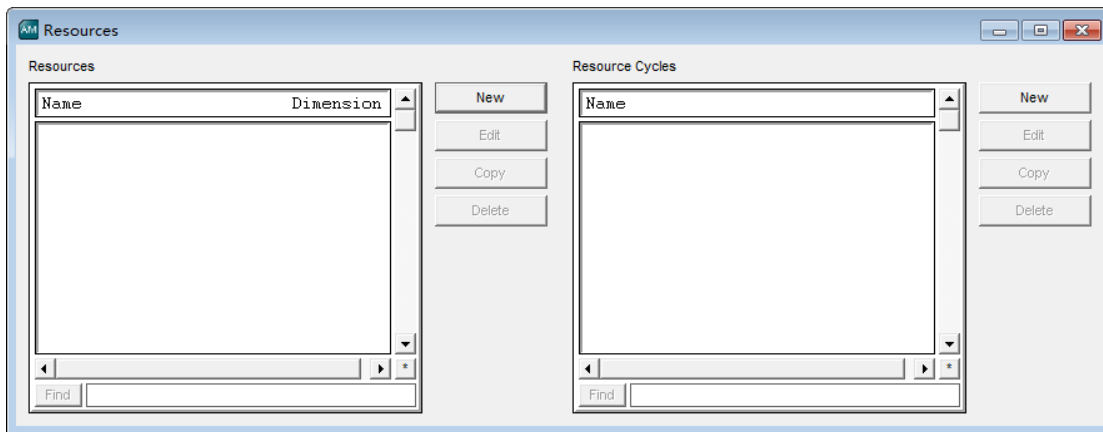
Define A Creation Spec 视窗



Pick A First Process 视窗

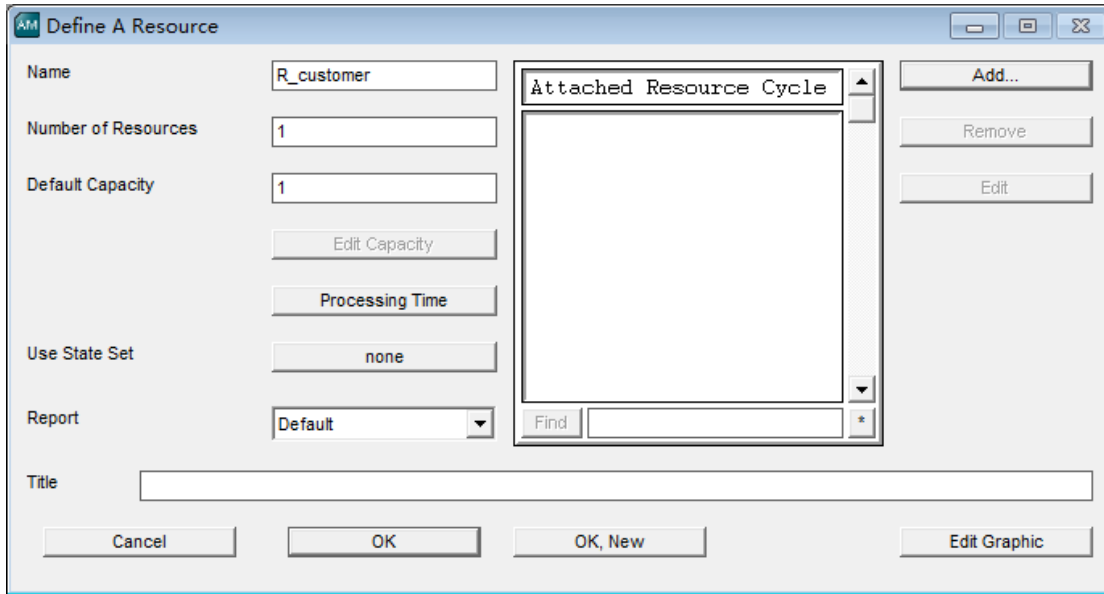
2-3 定义 Resources

Step1 鼠标点击 Process System 的工具箱中的 Resources，打开 Resources 视窗。



Resources 视窗

Step2 点击 New 选项定义一个新的 Resource，即可打开 Define A Resource 视窗。输入 Name 为 R_worker, Number of Resource 为 1, Default Capacity 为 1 即可完成资源 R_worker 的定义。设定参数的含义是有名字为 R_worker 的工作人员 1 名，它每次能服务的顾客数量为 1。

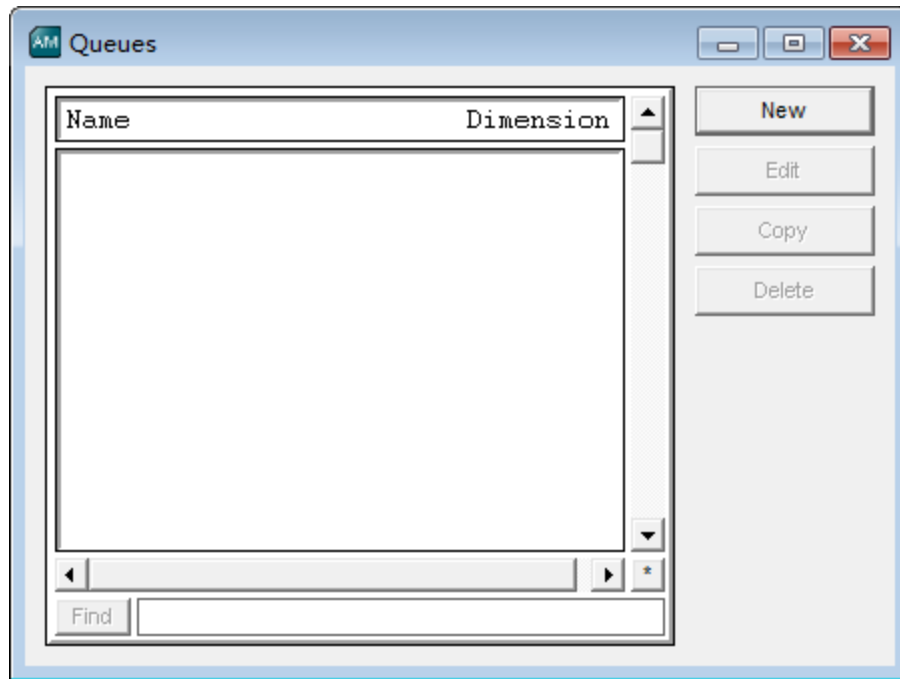


Define A Resource 视窗

Step3 按照以上步骤，定义 R_desk。

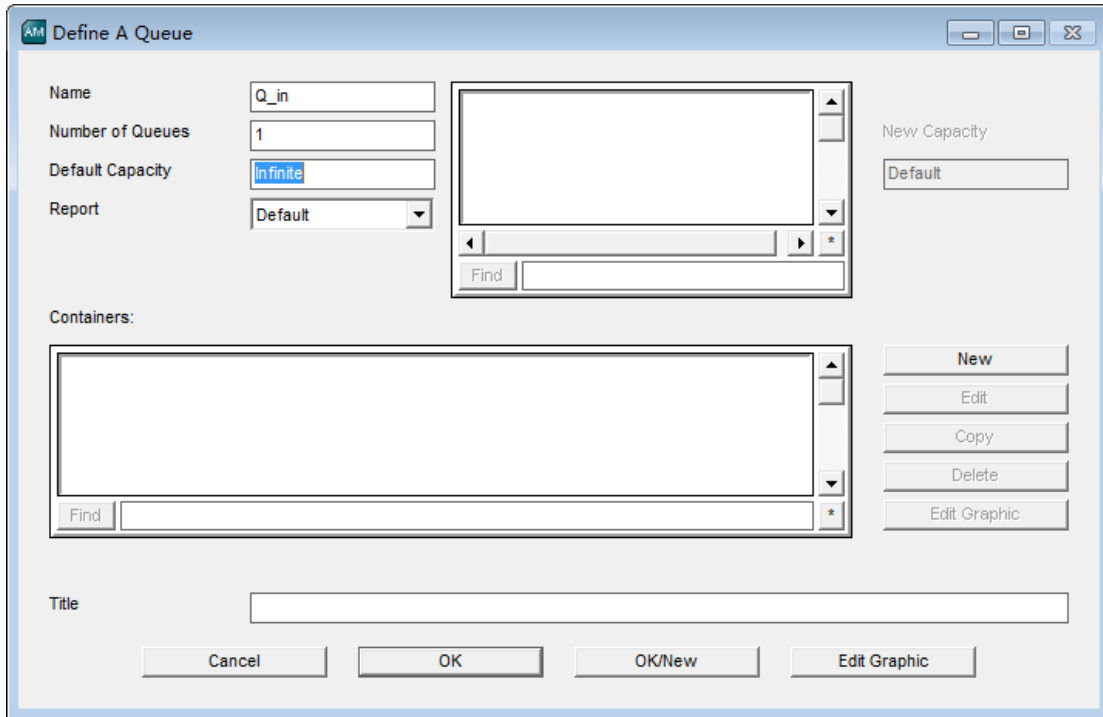
2-4 定义 Queues

Step1 鼠标点击 Process System 的工具箱中的 Queues，打开 Queues 视窗。



Queues 视窗

Step2 点击 New 选项定义一个新的 Queue, 即可打开 Define A Queue 视窗。输入 Name 为 Q_in, Number of Queues 为 1, Default Capacity 为 Infinite 即可完成队列 Q_in 的定义。设定参数的含义是有名字为 Q_in 的队列 1 列，它的容量为无限。



Define A Queue 视窗

Step3 按照以上步骤，定义 Q_out。

3.绘制模型平面图

在定义 Loads, Resources, Queues 等实体的视窗右下角都会有一个 Edit Graphic 按键，点击它就能进入相应实体三维视图的编辑界面。模型的平面布置是可以在定义实体的过程中同时完成的，无需等到所有实体定义完成后才进行。接下来以 Q_in 为例，介绍实体是如何编辑三维视图的。

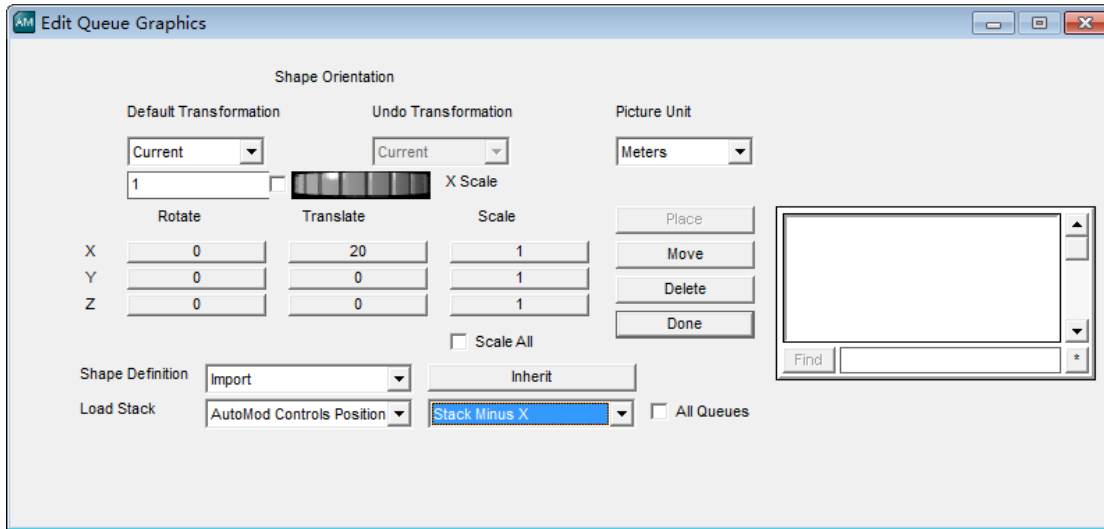
Step1 点击 Process System 工具箱中的 Queues，打开 Queues 视窗，选择所要设定的 Q_in，并点击 Edit 打开 Edit A Queue 视窗。

Step2 点击 Edit Grapics 即进入 Edit Queue Grapics 视窗。

Step3 点击 Place 后在编辑界面适当位置点放 Q_in，并在 Edit Queue Grapics 视窗中调整 Q_in 的大小（Scale）及位置（Translate），Queue 是否绕坐标轴旋转（Rotate）。

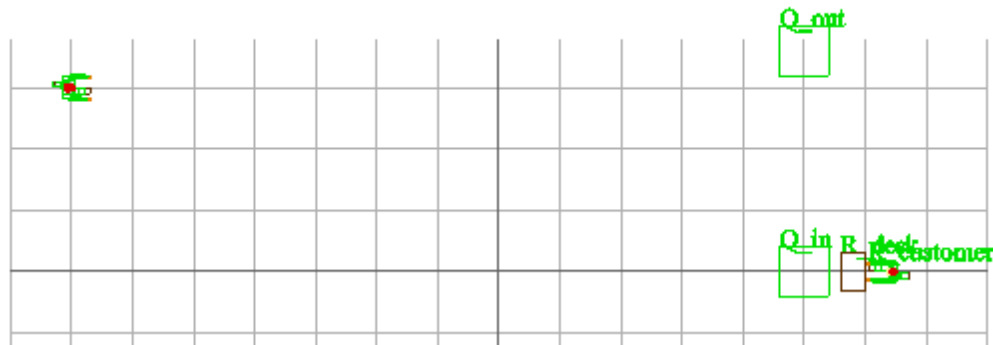
Step4 如果要改变放置的位置，可以点击 Move 或者改变 Queue 的三维坐标（即 Translate 下的数字）；删除放置的 Queue 则点击 Delete，如果放置完毕则点击 Done 退出该视窗。

Step5 选择 Q_in 的 Loads 堆叠的方式，要使 Load 沿 X 轴正方向堆叠，则点击 Load Stack 的第一个组合框选择 AutoMod Controls Position，第二个组合框选择 Stack PlusX 即可。点击第二个组合框之上按键可以选择 Queue 的颜色。



Edit Queue Graphics 视窗

Step6 按照以上步骤，定义模型中需要三维显示的 Loads, Resources, Queues 等。设定的步骤类似，可以自行练习。



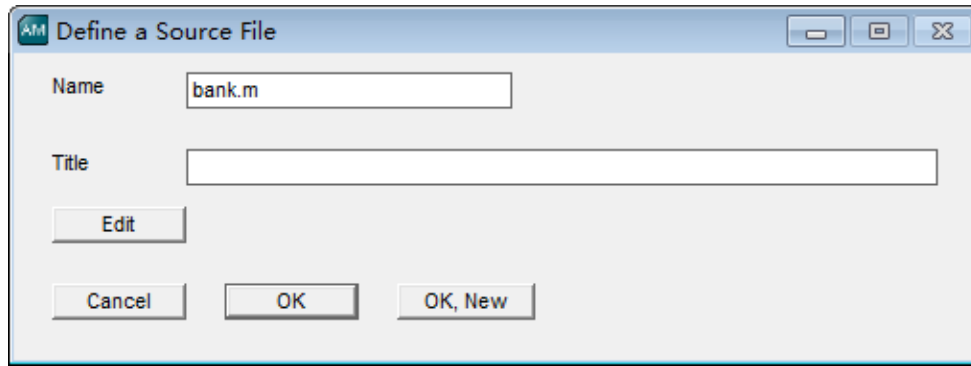
单服务台银行排队系统平面布置图

4.定义 Source File

Step1 点击 Process System 的工具箱中的 Source Files 视窗，打开 Source Files 视窗。

!!!注意：定义一个 Source File 时要注意文件的后缀名必须是.m。

Step2 点击 Source Files 视窗中的 New 选项定义一个新的 Source File，即可打开 Define A Source File 视窗。输入 Name 为 bank.m 点击 OK 就完成了 Source File 的定义。



Define a Source File 视窗

Step3 点击 Define A Source File 视窗或者 Source Files 视窗中的 Edit 键，即可开始编写仿真程序。

```

1begin P_1 arriving
2    move into Q_in
3    use R_desk for uniform 5,0 min
4    move into Q_out
5    wait for e 1 min
6    send to die
7end
8
9
10

```

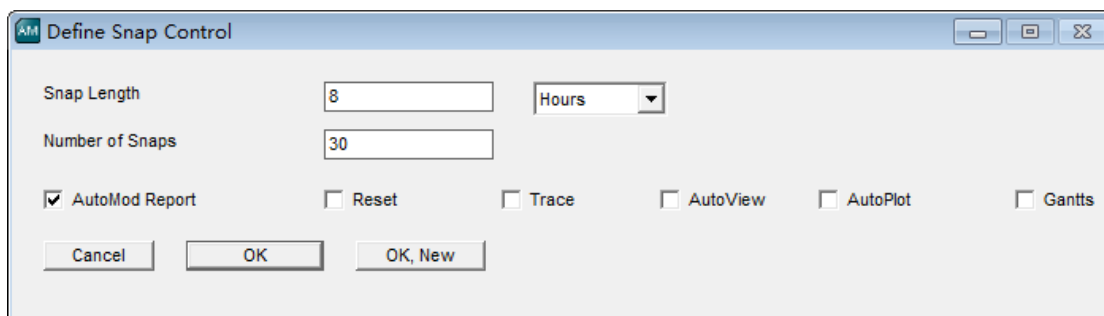
AutoMod Editor 视窗

5 设定 Run Control

本模型中的系统每天工作时间为 8 小时，连续仿真一个月（即 30 天），仿真运行完成后要求输出仿真报告。

Step1 点击 Process System 的工具箱中的 Run Control 视窗，打开 Run Control 视窗。

Step2 点击 New 定义一个新的 Run Control（运行控制），即可打开 Define Snap Control 视窗。设定 Snap Length 为 8 小时（8 Hours），Number of Snaps 为 30，点勾 AutoMod Report 选项，即可完成设定。

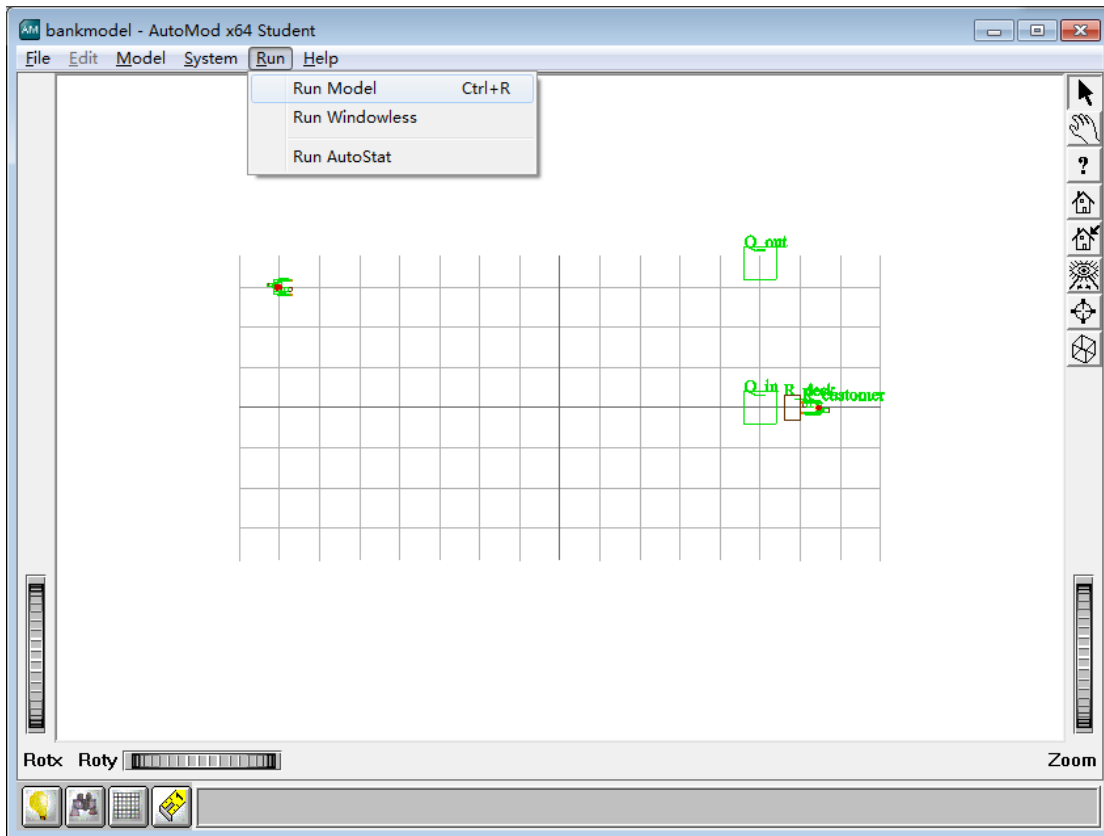


Define Snap Control 视窗

6 模型运行

当你已经编辑完一个新的模型或者载入了一个已经建好的模型后，便可以通过软件编辑界面菜单中 Run 下的 Run Model 选项运行模型。

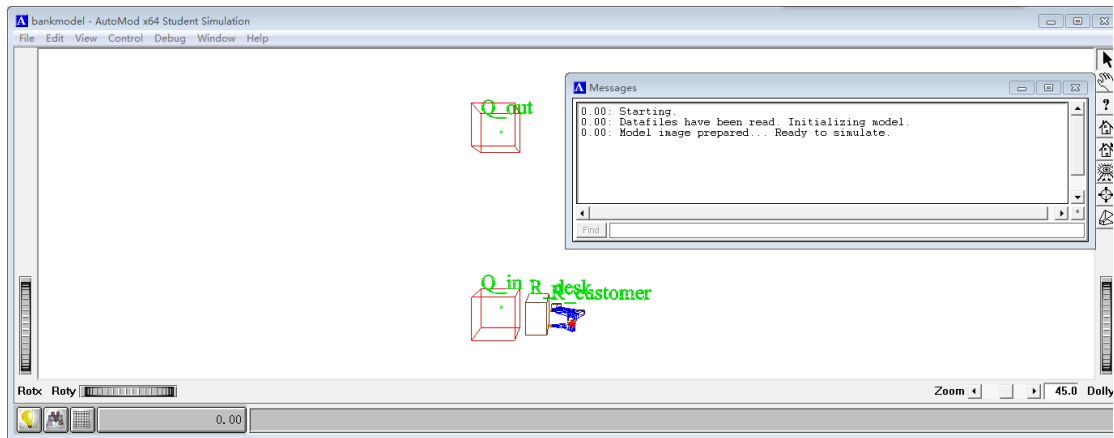
Step1 点击编辑界面菜单中 Run 下的 Run Model 选项。



在编辑界面选择 Run Model 以进入运行模式

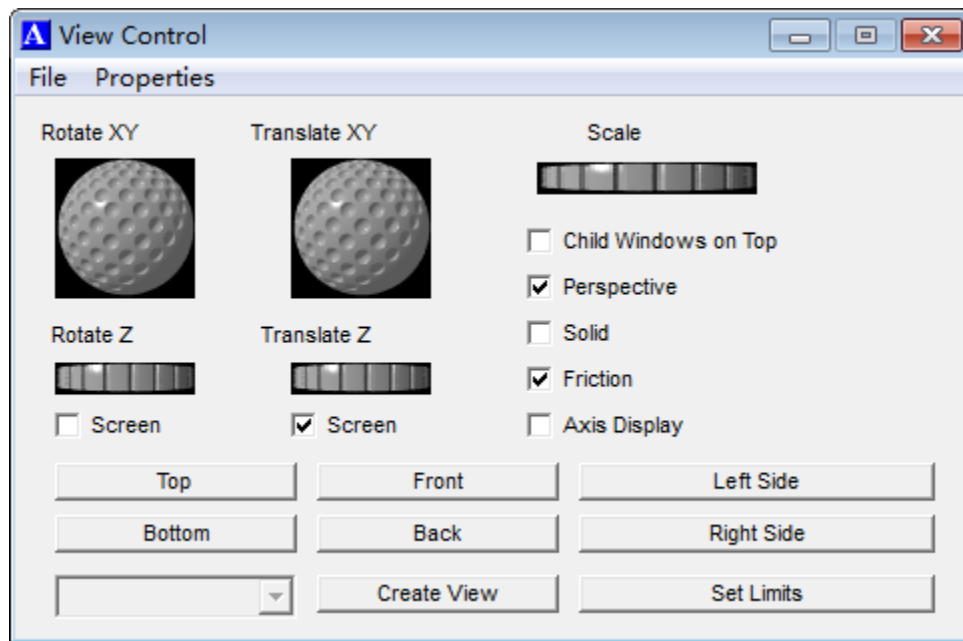
Step2 改变视野以便于观看仿真动画。编译连接完成之后即可进入运行模式的视窗，这里包含了三个部分：模拟视窗，消息视窗（Message），状态视窗（Status）。

方法一：用模拟视窗上的 3 个工具 Rotx（旋转 X 轴），Roty（旋转 Y 轴），Zoom（放缩）改变视野。

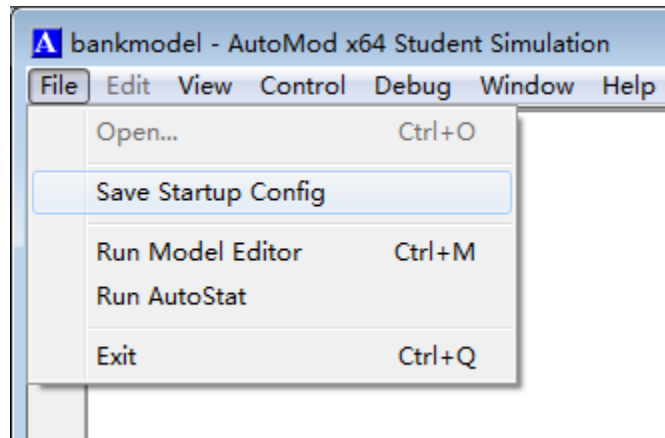


通过界面编辑改变视野

方法二：点击右下角的望远镜按钮，开启视窗来改变视野。在设置完视野后可以点击 File 菜单下的 Save Startup Config 选项保存视野，这样就不用每次都调整视野了。

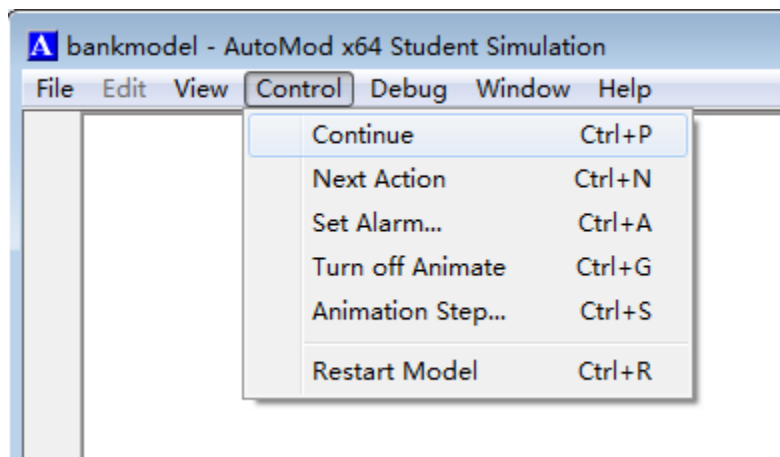


通过 View Control 视窗改变视野

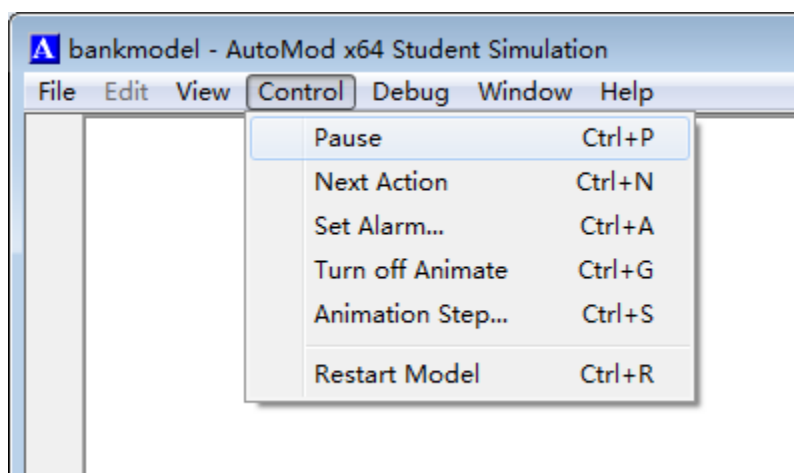


保存视野视窗

Step3 启动或暂停仿真运行。 点击模拟界面菜单的 Control 下的 Continue 即可启动仿真运行，开启之后要在关闭仿真运行时点击模拟界面菜单的 Control 下的 Pause 即可。

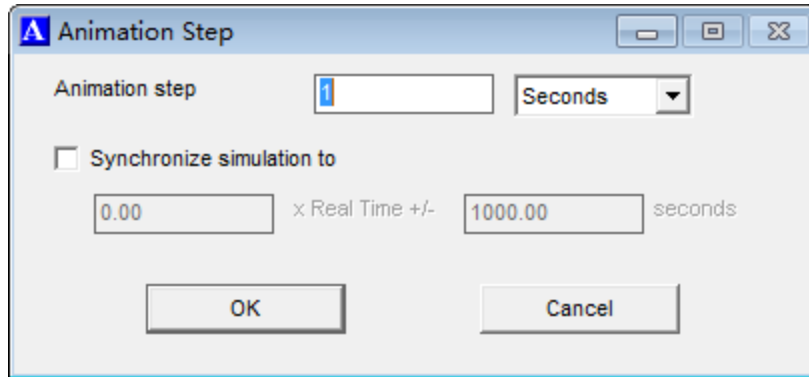


启动仿真运行



停止仿真运行

Step4 调整模型的动画速度。点击菜单选项 Control 下的 Animation Step 选项，在出现的 Animation Step 视窗中选择所要的时间及其单位。



Animation Step 视窗

也可以通过快捷键来调整速度，按键 D 减慢速度，Shift+D 加快速度。

Step5 3-D 模拟动画的开关。仿真运行开始后会根据 Animation Step 中所设定的时间来不断更新画面，如果你想减少运行的时间，可以将动画关闭。操作是点击菜单 Control 下的 Turn off Animation 选项。若要开启动画可点击菜单 Control 下的 Turn on Animation 选项。

7 查看输出报告

模型设置的 Snap Length 为 8h，一共 30 个 Snaps。

第一个 8h，输出报告如下：

```
*** AutoMod 12.5 ***
Model bankmodel
Statistics at Absolute Clock = 8:00:00.00, Relative Clock = 8:00:00.00
CPU time: Absolute: 43.062 sec, Relative: 43.062 sec

Statistics for Process System "bankmodel"

Process Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1                  92      6    3.37      --   10   0    --   1054.87

Process Traffic Limit Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1                  92      6    3.37 Infinite    10   0    --   1054.87

Queue Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
Space                92      0    0.00 Infinite     1    0    --     0.00
Q_in                 92      6    3.21 Infinite    10   0    --  1003.62
Q_out                86      0    0.16 Infinite     1    0    --    54.83

Resource Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait  State
=====
R_customer           0      0    0.00      1    0    0 0.000     0.00     0.00  Up  ----
R_desk               87      1    0.91      1    1    0 0.906    299.86   720.06  Up  ----

Random Number Streams
Name                Total
=====
stream0              178
stream_L_customer_1  93
stream_R_customer_1  0
stream_R_desk_1      0
```

图中：我们可以看到在第一个 Snap 结束之后，系统的状态。在第一个 8h 中，顾客的数量

为 93 位，其中，经过 Q_out 队列离开系统的有 86 位，目前 Q_in 队列中有 6 位，在服务台接受服务的有 1 位，一共 93 位（或经 Q_in 队列的有 93 位，在服务台接受服务的有 1 位，也是 93 位）。

```

*** AutoMod 12.5 ***
Model bankmodel
Statistics at Absolute Clock = 16:00:00.00, Relative Clock = 16:00:00.00
CPU time: Absolute: 84.039 sec, Relative: 84.039 sec

Statistics for Process System "bankmodel"

Process Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1           184      2    4.69      --   11   0   --   1469.20

Process Traffic Limit Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1           184      2    4.69 Infinite    11   0   --   1469.20

Queue Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
Space         184      0    0.00 Infinite     1    0   --     0.00
Q_in          184      2    4.51 Infinite    11   0   --   1412.81
Q_out         182      0    0.18 Infinite     2    0   --    57.01

Resource Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait  State
=====
R_customer     0      0    0.00      1    0    0  0.000     0.00     0.00  Up  ----
R_desk        183      1    0.95      1    1    0  0.953   299.93  1114.51  Up  ----

Random Number Streams
Name          Total
=====
stream0       366
stream_L_customer_1  185
stream_R_customer_1    0
stream_R_desk_1        0

```

图中：我们可以看出在第二个 Snap 结束之后，系统的状态。在前两个 8h 中，顾客的数量为 185 位，其中，经过 Q_out 离开系统的有 182 位，目前 Q_in 队列中有 2 位，在服务台接受服务的有 1 位，一共 185 位（或经 Q_in 队列的 184 位，在服务台接受服务的有 1 位，也是 185 位）。

依次，我们可以看到在所有仿真循环结束后系统的状态。

```

*** AutoMod 12.5 ***
Model bankmodel
Statistics at Absolute Clock = 10:00:00:00.00, Relative Clock = 10:00:00:00.00
CPU time: Absolute: 1323.52 sec, Relative: 1323.52 sec

Statistics for Process System "bankmodel"

Process Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1           2801     2   13.29      --  48   0   --   4100.19

Process Traffic Limit Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_1           2801     2   13.29 Infinite    48   0   --   4100.19

Queue Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
Space         2801     0    0.00 Infinite    1    0   --     0.00
Q_in          2801     2   13.10 Infinite    48   0   --  4041.64
Q_out         2799     0    0.19 Infinite    2    0   --    58.59

Resource Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait  State
=====
R_customer     0     0    0.00      1    0    0 0.000     0.00     0.00  Up  ----
R_desk        2800     1    0.97      1    1    0 0.972    299.98   3741.76  Up  ----

Random Number Streams
Name          Total
=====
stream0       5600
stream_L_customer_1 2802
stream_R_customer_1  0
stream_R_desk_1     0

```

图中：我们可以看出在第 30 个 Snap 结束之后，系统的状态。在 30 个 8h 中，顾客的数量为 2802 位，其中，经过 Q_out 离开系统的有 2799 位，目前 Q_in 队列中有 2 位，在服务台接受服务的有 1 位，一共 2802 位（或经 Q_in 队列的 2801 位，在服务台接受服务的有 1 位，也是 2802 位）。