

实验四、经典报童问题的建模与仿真

1. 实验目的

进一步熟悉 AutoMod 软件，学习其对库存问题的建模与仿真分析方法。通过基于 Excel 和基于 AutoMod 的仿真建模比较，理解多种仿真建模方法的差异和相似之处。

通过对经典报童问题的建模与仿真，加深对 AutoMod 建模的认识和理解。并将 AutoMod 建模与 Excel 模型结果进行对比，观察收益最大时的订货量分别是多少。

2. 实验内容

(1) 对象系统描述：

报童问题描述的是单周期存储中的订购决策问题，即报童每天售出的报纸份数是一个离散随机变量，如果报纸的进货量低于需求量，那么报纸全部售出，但会给报童带来缺货损失（即：不能满足客户需求带来的损失，以及因不能满足客户需求而失去一部分客户带来的损失），为简化计算，在本模型中，我们忽略缺货损失；如果进货量高于需求量，那么报纸一部分剩余，报童将以较低价格退回报社，减少了盈利。所以，在实际考虑中，应该使进货量尽可能接近需求量，使报童获得更大的盈利。

本模型将以以下案例为基础进行建模：

报童每天以 6 美分/份的价格购进一批报纸，以 11.99 美分/份的价格卖出，顾客需求服从参数为 (70,10) 的均匀分布，每天未卖出的报纸将以每份 3 美分的价格退回报社，报童问题模型将得到报童每天最佳的进货量以获得最大收益。

(2) 实验内容

- ① 完成该报童模型建模。
- ② 对进货量进行仿真分析和决策。
- ③ 对比 AutoMod 模型与 Excel 模型，分析异同。

3. 建模设计

以下是其中一种建模方法，以供参考：

进货量、销售量、退回量分别以队列形式来表示：每天所进报纸均在队列中等待，每天结束之后，进货量-销售量为此时队列中报纸份数，同时销售出去的报纸进入顾客队列，未销售出去的报纸进入回收队列，等待退回报社。

4. 实验步骤

- (1) 新建一个模型（自动创建一个 process system）；
- (2) 定义 Loads、Queues 等实体单元；
- (3) 定义系统各个变量；
- (4) 编辑 source file 文件，编写相应代码，定义系统的流程逻辑；
- (5) 根据系统特征设定运行时间；
- (6) 运行模型；
- (7) 查看仿真动画和仿真输出结果，分析报童问题的收益。

作业与思考题

- ① 如何判断最佳进货量？
- ② 你是如何设定运行时间的？为什么？
- ③ 用 AutoMod 软件建立的报童仿真模型与用 Excel 建立的报童仿真模型有什么异同？

自选内容

将报童模型中的缺货损失分别设为 0、单价、二倍单价，分别运行模型，看看对结果有什么影响？

实验四详细解析

本实验学习的重点在于对各个变量的设置，以及通过改变进货量观察收益的变化以得到报童问题的最佳进货量和最大收益。以下是完成本系统模型所需要的项目设置。

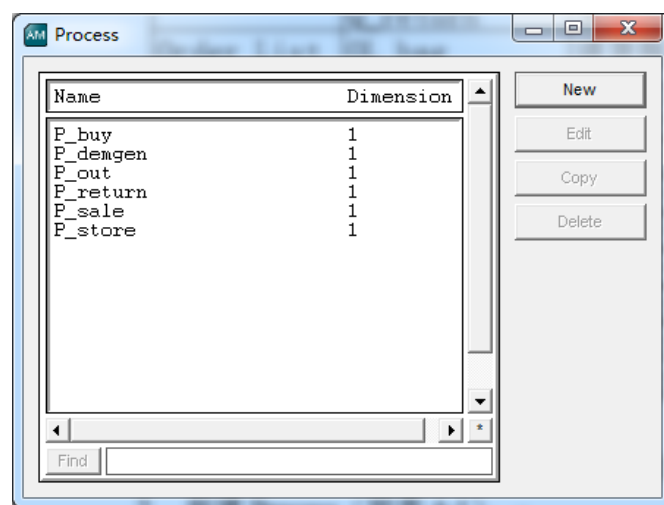
表 4-1 经典报童问题建模与仿真项目设置

Process系统		
项目	名称	备注
Process	P buy	产生进货报纸
	P demgen	产生需求量, Uniform (70,10)
	P out	销售出去的报纸离开系统
	P return	未卖出的报纸进行回收
	P sale	根据需求销售报纸
	P store	所进报纸进入队列储存
Loads	L dailydrive	每天七点进货进行销售的驱动型Load
	L newspaper	报纸
Queues	Q bag	进货报纸的储存区
	Q customer	销售给顾客的报纸
	Q return	未卖出的报纸进行回收
Order List	OL bag	进货报纸的逻辑缓存区
Variables	V CSoutputfile	FilePtr型变量, 将结果写到输出文件
	V buy	整型变量, 进货量
	V cost	Real型变量, 进价, 6美分
	V dem	整型变量, 需求量
	V left	整型变量, 剩余量
	V oos	整型变量, 缺货量
	V price	Real型变量, 售价, 11.99美分
	V profit	Real型变量, 收益
	V salvage	Real型变量, 未卖出报纸的回收价格, 3美分

1. 新建一个模型（calendar selling）。

修改单元格长度为米，并保存和输出模型。

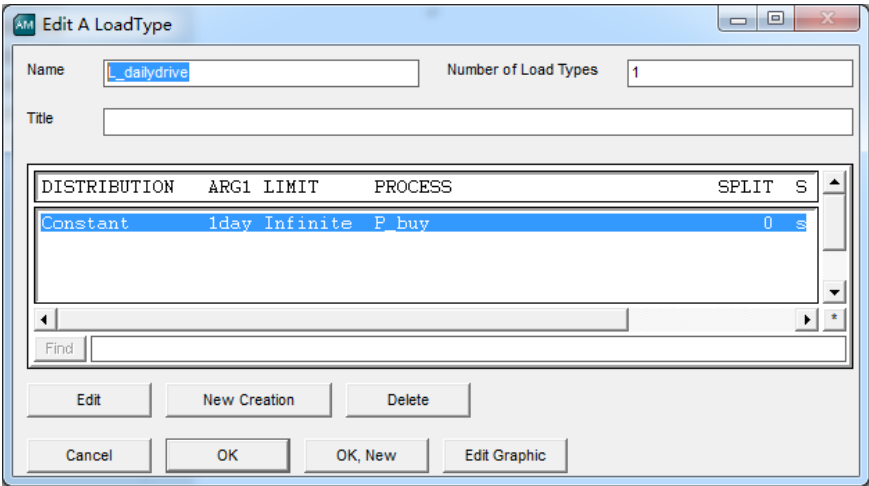
2. 新建 Process（按表 4-1）。



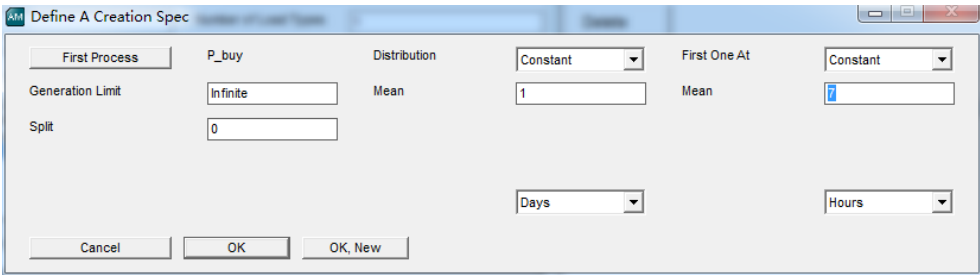
Process 视窗

3. 新建 Loads。

Step3-1 新建 L_dailydrive。



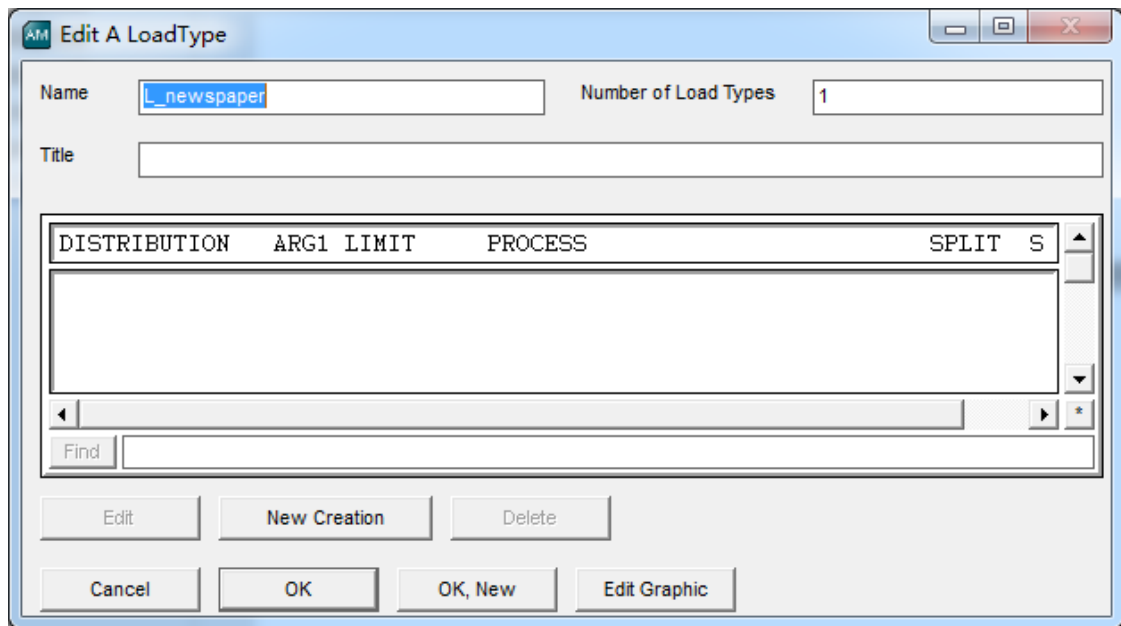
Edit A LoadType 视窗



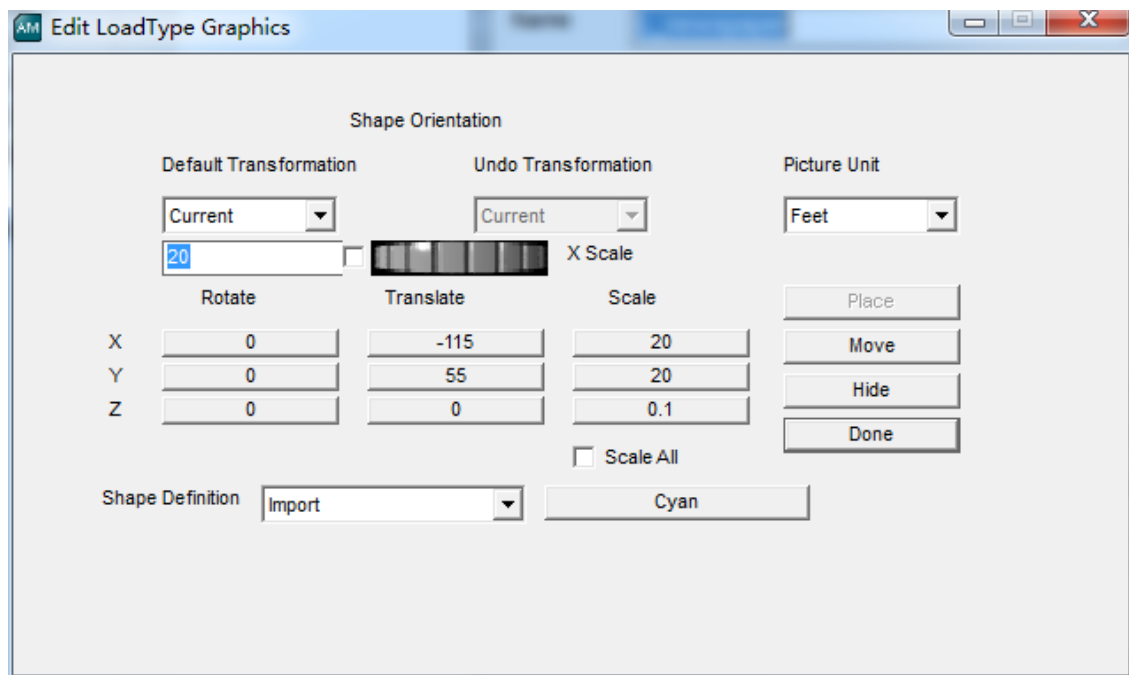
Define A Creation Spec 视窗

注：所设 L_dailydrive 并未放置到系统中，因为它只是一个驱动型 Load，驱动程序的运行。

Step3-2 新建 L_newspaper。



Edit A LoadType 视窗

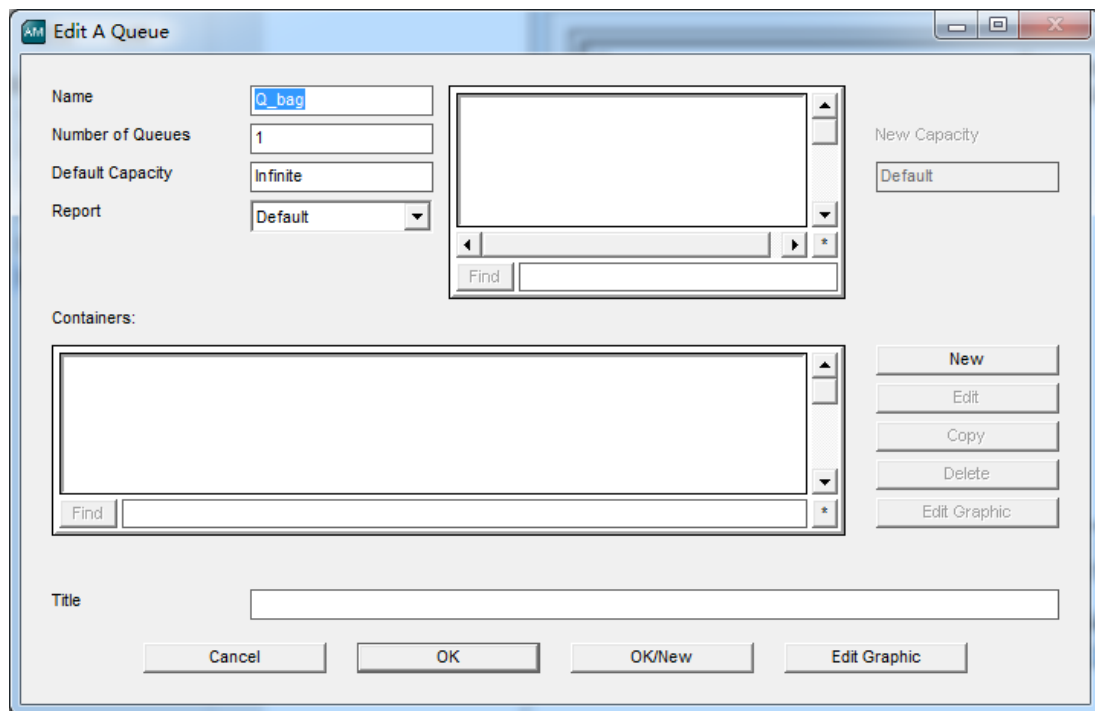


Edit LoadType Graphics 视窗

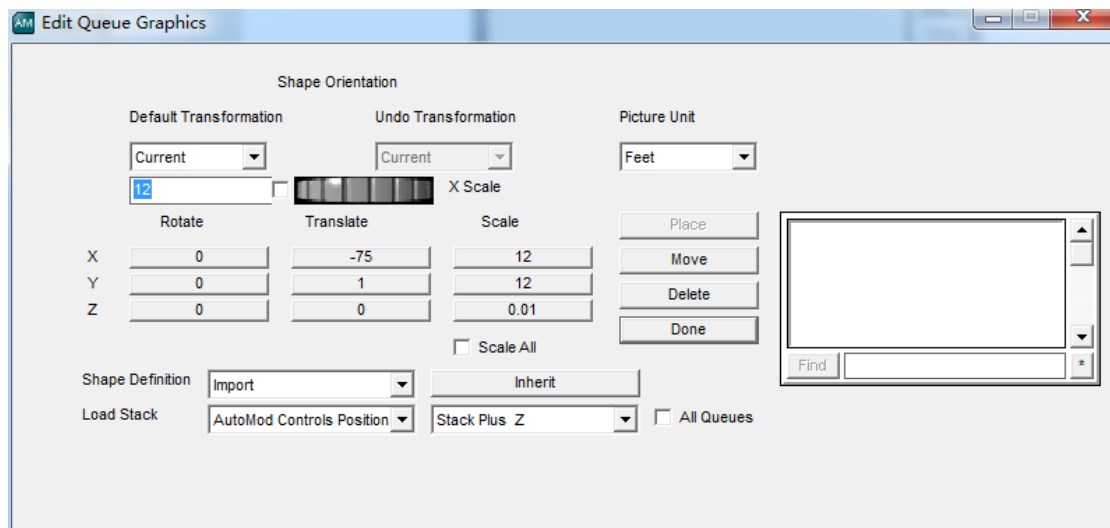
注：所建 L_newspaper 并没有设置 Creation，但要放置在系统中。Load 在代码中让其产生。

4. 新建队列。

Step4-1 新建 Q_bag。

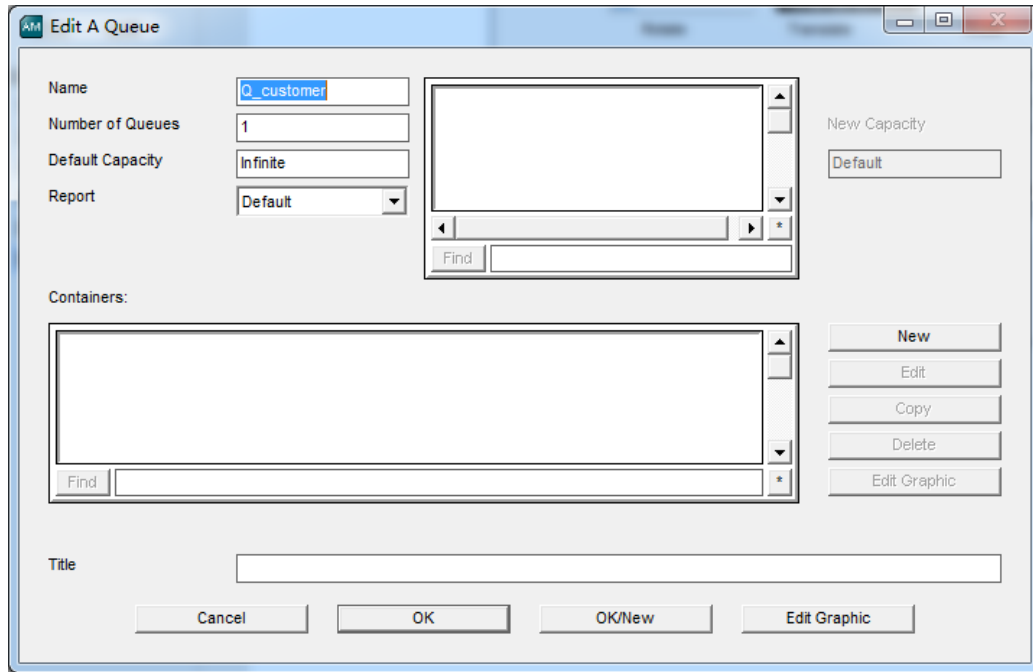


Edit A Queue 视窗

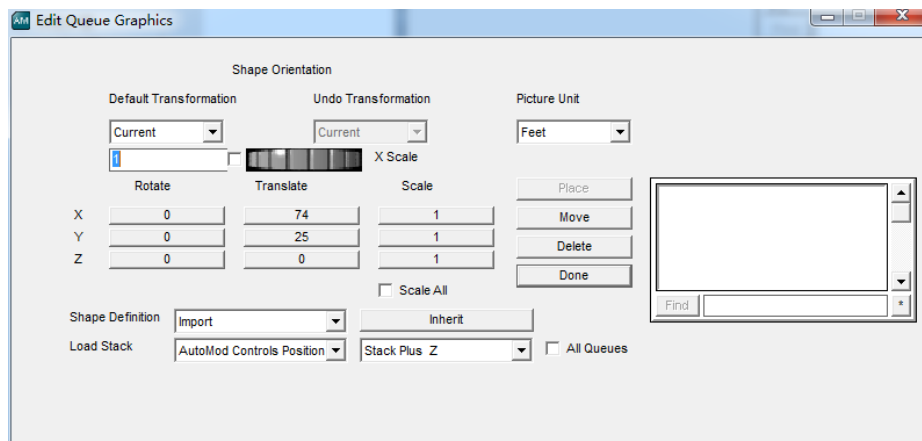


Edit Queue Graphics 视窗

Step4-2 新建 Q_customer。

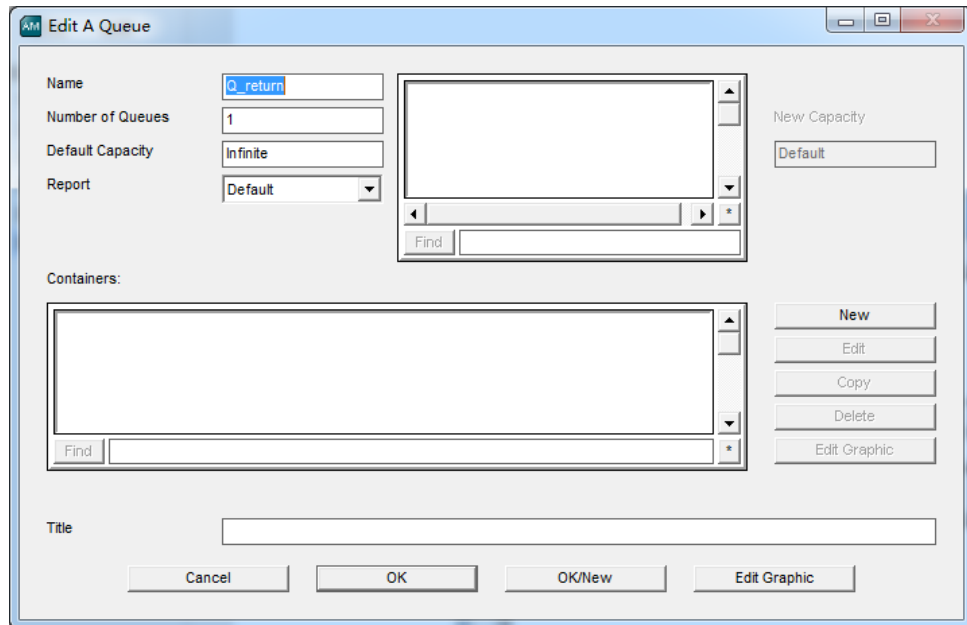


Edit A Queue 视窗

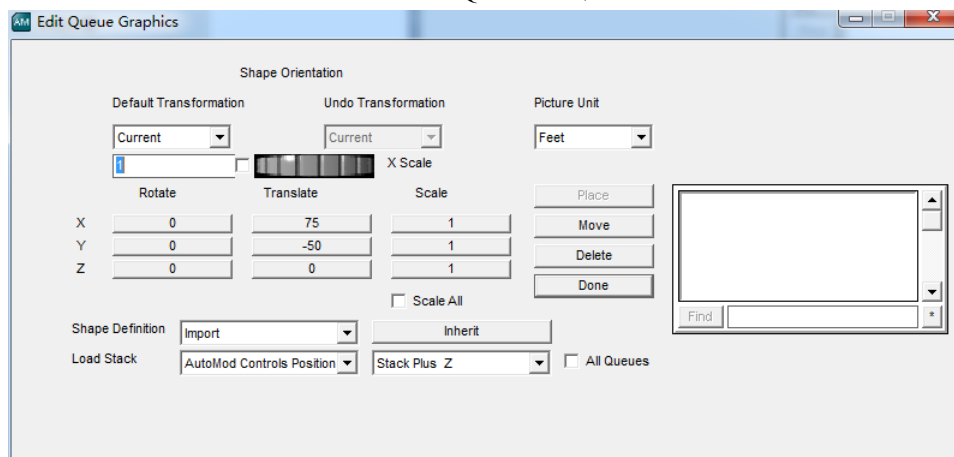


Edit Queue Graphics 视窗

Step4-3 新建 Q_return。



Edit A Queue 视窗



Edit Queue Graphics 视窗

5. 新建 Order List。

AMM Edit An Order List

Name Number of Order Lists

Sort by

with value first

If tie then use as tie breaker

Report

Title

Edit An Order List 视窗

6. 新建 Variables。

AMM Variables

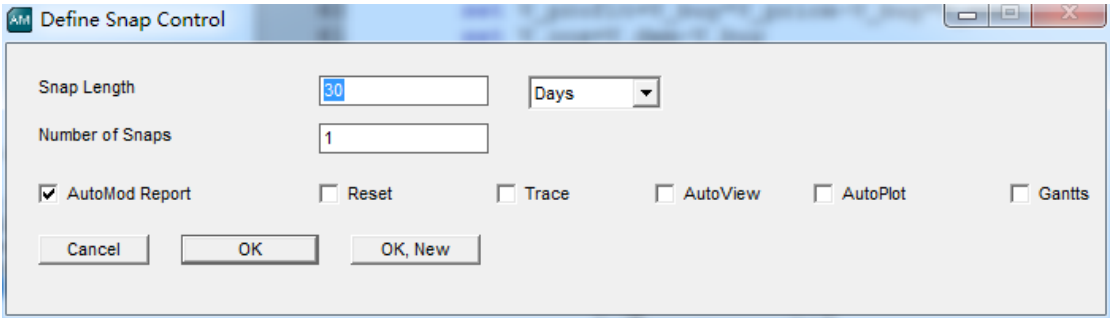
Name	Type	Dimension
V_CSoutputfile	FilePtr	1
V_buy	Integer	1
V_cost	Real	1
V_dem	Integer	1
V_left	Integer	1
V_oos	Integer	1
V_price	Real	1
V_profit	Real	1
V_salvage	Real	1

Variables 视窗

7. 新建 Source File 文件。注：文件后缀一定要加.m。

```
1begin model initialization function
2    set V_buy=90
3    set V_cost=6
4    set V_price=11.99
5    set V_salvage=3
6    open "CSresult.txt" for writing save result as V_CSoutputfile
7    return true
8end
9
10begin P_buy arriving
11    create V_buy loads of loadtype L_newspaper to P_store
12    send to P_demgen
13end
14
15begin P_store arriving
16    move into Q_bag
17    wait to be ordered on OL_bag
18end
19
20
21begin P_demgen arriving
22    set V_dem = normal 70,10
23    wait for 11 hr
24    send to P_sale
25end
26
27begin P_sale arriving
28
29    if V_buy >= V_dem then
30        begin
31            order V_dem loads from OL_bag to P_out
32            set V_profit=V_dem*V_price-V_buy*V_cost+(V_buy-V_dem)*V_salvage
33            set V_left=V_buy-V_dem
34            order V_left loads from OL_bag to P_return
35            print V_profit to V_CSoutputfile
36        end
37    if V_buy < V_dem then
38        begin
39            order V_buy loads from OL_bag to P_out
40            set V_profit=V_buy*V_price-V_buy*V_cost
41            set V_oos=V_dem-V_buy
42            print V_profit to V_CSoutputfile
43        end
44    end
45end
46
47begin P_out arriving
48    move into Q_customer
49    wait for 4 hr
50    send to die
51end
52
53begin P_return arriving
54    move into Q_return
55    wait for 4 hr
56    send to die
57end
58
```

8. 设 Run Control。



Define Snap Control 视窗

9. 运行模型。



10. 查看输出报告和输出文件。

Step10-1 查看输出报告。

```

*** AutoMod 12.6 ***
Model calendarselling
Statistics at Absolute Clock = 30:00:00:00.00, Relative Clock = 30:00:00:00.00
CPU time: Absolute: 7.719 sec, Relative: 7.719 sec

Statistics for Process System "calendarselling"

Process Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
P_buy         30      0    0.00      --    1    0    --      0.00
P_sale        30      0    0.00      --    1    0    --      0.00
P_demgen      30      0    0.46      --    1    0    --    39600.00
P_store      2700    0   41.25      --   90    0    --    39600.00
P_out        2033    0   11.29      --   85    0    --    14400.00
P_return      667     0    3.71      --   43    0    --    14400.00

Process Traffic Limit Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
P_buy         30      0    0.00 Infinite    1    0    --      0.00
P_sale        30      0    0.00 Infinite    1    0    --      0.00
P_demgen      30      0    0.46 Infinite    1    0    --    39600.00
P_store      2700    0   41.25 Infinite   90    0    --    39600.00
P_out        2033    0   11.29 Infinite   85    0    --    14400.00
P_return      667     0    3.71 Infinite   43    0    --    14400.00

Queue Statistics
Name          Total    Cur  Average Capacity  Max  Min  Util   Av_Time  Av_Wait
=====
Space        2730    0    0.46 Infinite    2    0    --     435.16
Q_bag        2700    0   41.25 Infinite   90    0    --    39600.00
Q_customer   2033    0   11.29 Infinite   85    0    --    14400.00
Q_return     667     0    3.71 Infinite   43    0    --    14400.00

Order List Statistics
Name          Total    Cur  Average  Max  Min  Av_Time
=====
OL_bag        2700    0   41.25   90    0  39600.00

Order List Backordered Statistics

Name          Total    Cur  Average  Max  Min  Av_Time
=====
OL_bag        2700    0   41.25   90    0  39600.00

Order List Backordered Statistics
Name          Total    Cur  Average  Max  Min  Fulfilled
=====
OL_bag         0      0    0.00     0    0         0

Random Number Streams
Name          Total
=====
stream0       40
stream_L_dailydrive_1 0

```

从报告中可以看出，30 天一共进了 2700 份报纸。

Step10-2 查看输出文件 CSresult.txt。

从输出文件 CSresult 中，可以看到这 30 天每天的收益。

通过改变进货量，我们可以得到不同进货量下，30 天的收益，通过比较，我们可以得到最优进货量。

```
323.340000
287.380000
440.210000
305.360000
368.290000
215.460000
152.530000
332.330000
386.270000
305.360000
188.490000
494.150000
251.420000
287.380000
431.220000
377.280000
287.380000
404.250000
260.410000
404.250000
224.450000
368.290000
476.170000
404.250000
395.260000
422.230000
278.390000
359.300000
350.310000
395.260000
```