

# 实验五、生产-搬运-库存系统综合实验

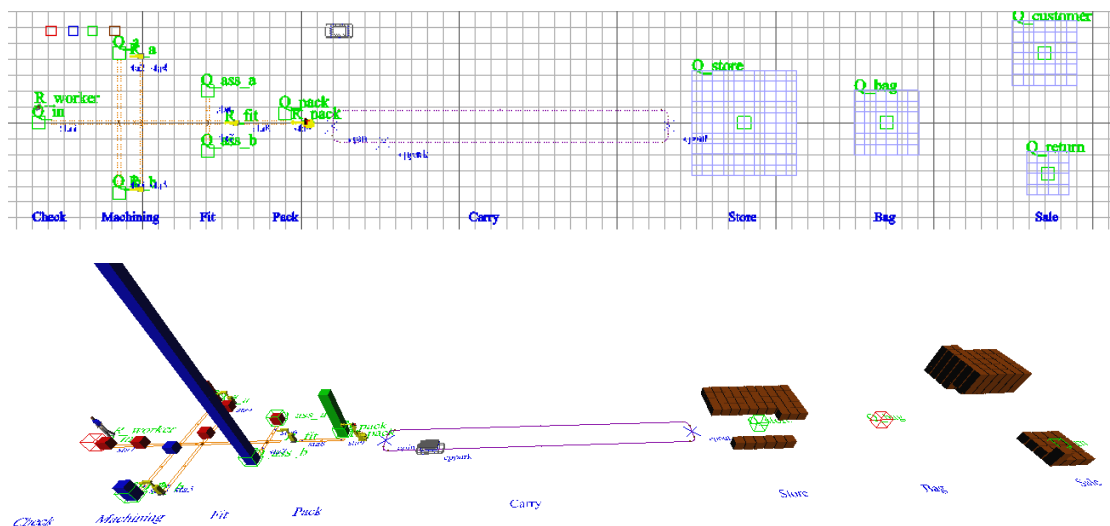
## 1 . 实验目的

通过应用所学的 Conveyor 子系统、Pathmover 子系统和库存系统完成生产-搬运-库存系统的仿真建模，加深对 AutoMod 建模的理解。

## 2 . 实验内容

### (1) 对象系统描述

有一个生产-搬运-库存系统，其布局如下：



该系统的流程及相关参数如下：

- 两种工件 L\_a、L\_b 分别以指数分布（5min）和均匀分布（5，2）min 的时间间隔到达车间，在检验处（队列 Q\_in）共同等待检验；
- 操作工人在检验处进行人工检验，每件检验用时服从指数（1min）分布。不合格的工件废弃；合格的送往后续加工工序，合格率为 95%；
- 合格的 L\_a 送往机器 R\_a 加工，如需等待，则在 Q\_a 队列中等待；合格的 L\_b 送往机器 R\_b 加工，如需等待，则在 Q\_b 队列中等待；
- L\_a 在机器 R\_a 上的加工时间服从指数分布（4）min；L\_b 在机器 R\_b 上的加工时间服从均匀分布（5，2）min；
- 1 个 L\_a 和 1 个 L\_b 在装配工序上装配成 L\_product，装配工时服从正态分布（5，

- 1) min, 装配等待时,  $L_a$  和  $L_b$  分别在各自的队列  $Q_{ass\_a}$ 、 $Q_{ass\_b}$  中等待;
- 产品  $L_{product}$  输送到打包机进行包装, 每箱装 10 个产品, 打包时间服从正态分布  $(4, 0.5)$  min, 然后循径系统由小车搬运至仓库储存。
  - 零售商每周向生产商订货, 订货量是一个离散随机变量, 订价为 6 元, 售价为 11.99 元, 顾客需求服从参数为  $(150, 10)$  的均匀分布, 未卖出的货物以每件 3 元退回生产商。

## (2) 实验内容

- ① 完成该系统建模。
- ② 对进货量进行仿真分析和决策。

## 3. 实验步骤

- (1) 新建一个模型 (自动创建一个 process system);
- (2) 定义小车、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) 新建 Conveyor 子系统和 Pathmover 子系统, 完成路径和站点创建。
- (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,

## 实验六详细解析

通过应用所学的 Conveyor 子系统、Pathmover 子系统和库存系统完成生产-搬运-库存系统的仿真建模, 加深对 AutoMod 建模的理解。下表是完成本系统模型所需项目设置。

项目设置

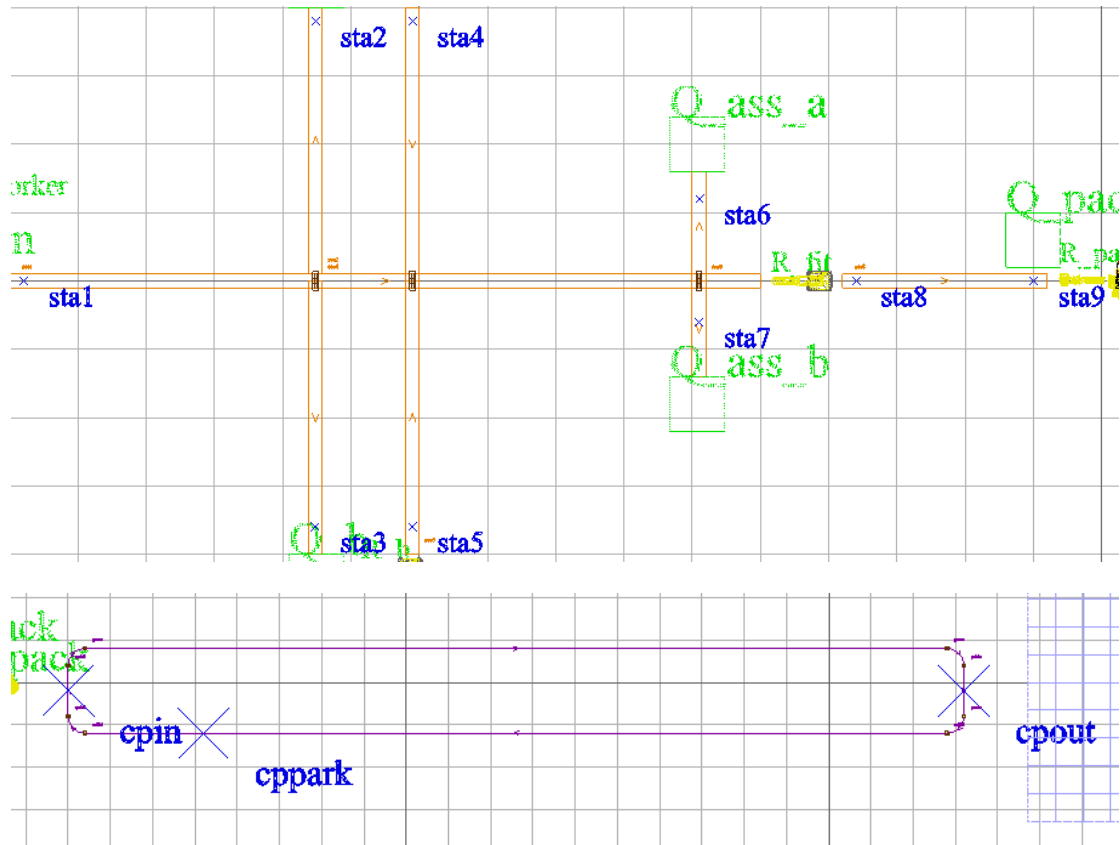
Process系统		
项目	名称	备注
Process	P in	两种工件进入系统
	P match	两种工件进行再加工
	P assfit	准备装配
	P fit	装配
	P pack	打包
	P move	小车搬运
	P store	储存
	P buy	零售商进货
	P bag	进货进入队列存储
	P demgen	产生需求量, Uniform (150,10)
	P sale	根据需求销售
	P out	销售出去的离开系统
	P return	未销售出去的返回给生产商
Loads	L a	工件a, e 5 min
	L b	工件b, uniform (5,2) min
	L demgen	需求驱动型load
	L fitdummy	装配驱动型load
	L pack	打包后的产品
	L packdummy	打包驱动型load
	L product	产品
Resources	R a	工件a加工
	R b	工件b加工
	R fit	装配
	R pack	打包
	R worker	检查人员
Queues	Q a	工件a加工等待队列
	Q b	工件b加工等待队列
	Q ass a	工件a等待装配队列
	Q ass b	工件b等待装配队列
	Q bag	进货进入队列存储
	Q customer	销售给顾客部分进入队列
	Q in	工件a, b进入系统等待队列
	Q pack	打包后等待队列
	Q return	未销售出去的等待队列
	Q store	打包后产品储存队列
Order List	OL ass a	工件a等待装配缓存区
	OL ass b	工件b等待装配缓存区
	OL bag	进货缓存区
	OL pack	打包后产品缓存区
	OL store	打包后产品储存缓存区
Variables	V CSoutputfile	输出文件
	V buy	进货量, 整型
	V cost	进价, real型
	V dem	需求量, 整型
	V left	剩余量, 整型
	V oos	缺货量, 整型
	V price	售价, real型
	V profit	利润, real型
	V salvage	回收价, real型

## 1. 新建一个模型（zonghe）

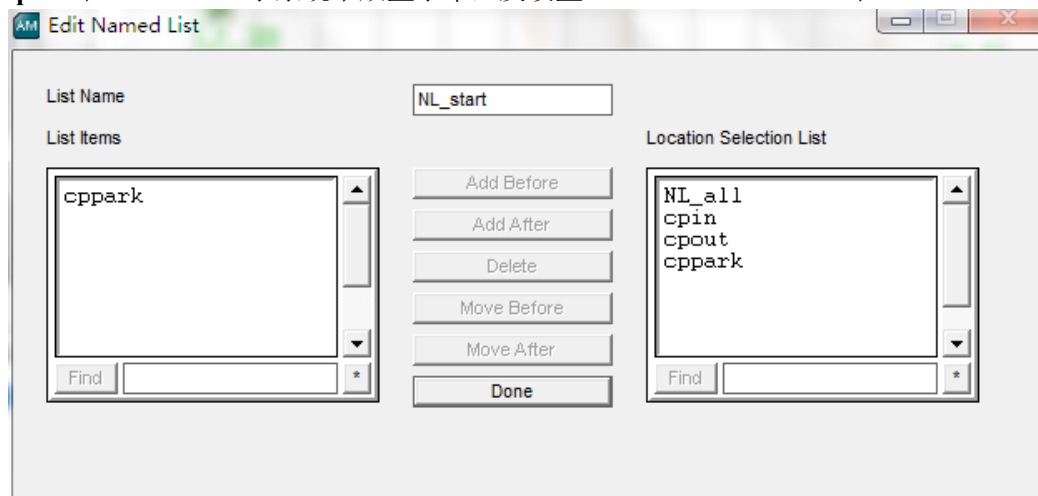
修改单元格长度为米，并保存和输出模型。

## 2. 新建 Conveyor 子系统和 Pathmover 子系统

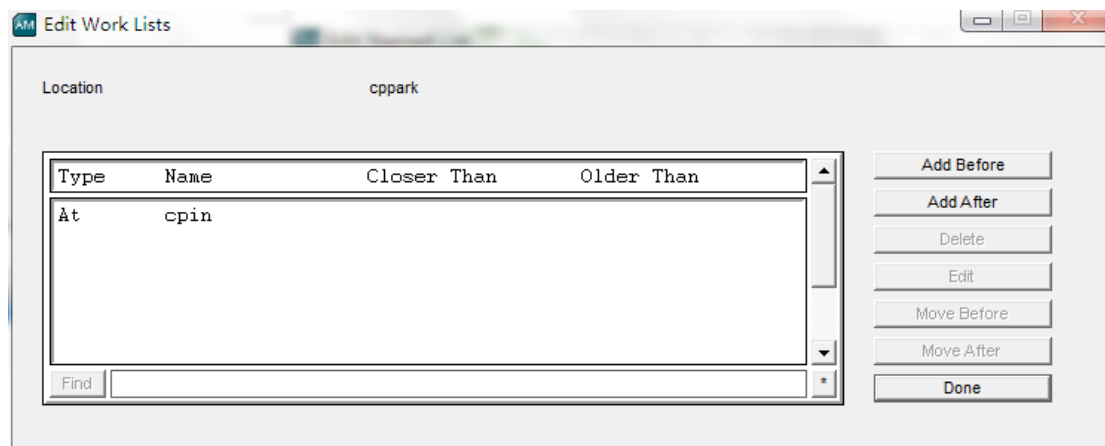
**Step2-1** 点击 System-New，新建 Conveyor 子系统（conv）和 Pathmover 子系统（pm）。并设置路径和站点如下图。



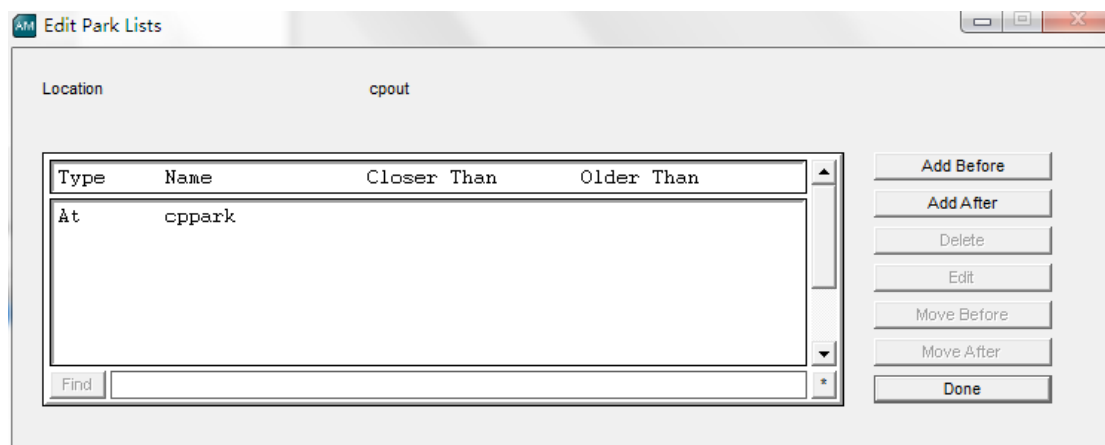
**Step2-2** 在 Pathmover 子系统下放置小车，及设置 Name List、Work List 和 Park List。



Edit Named List 视窗



Edit Work Lists 视窗



Edit Park Lists 视窗

### 3. 新建 Process

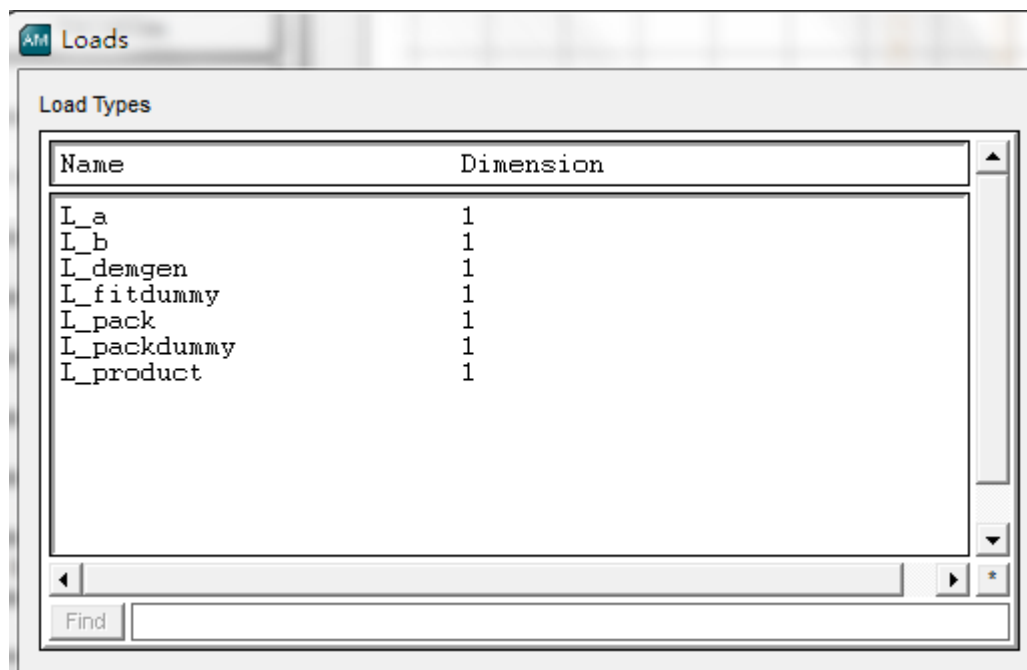
点击 System-Open, 打开 Process 编辑界面。点击 Process-New, 新建 Process 如下。



Process 视窗

#### 4. 新建 load

点击 Loads-New，新建各种 Load 如下。

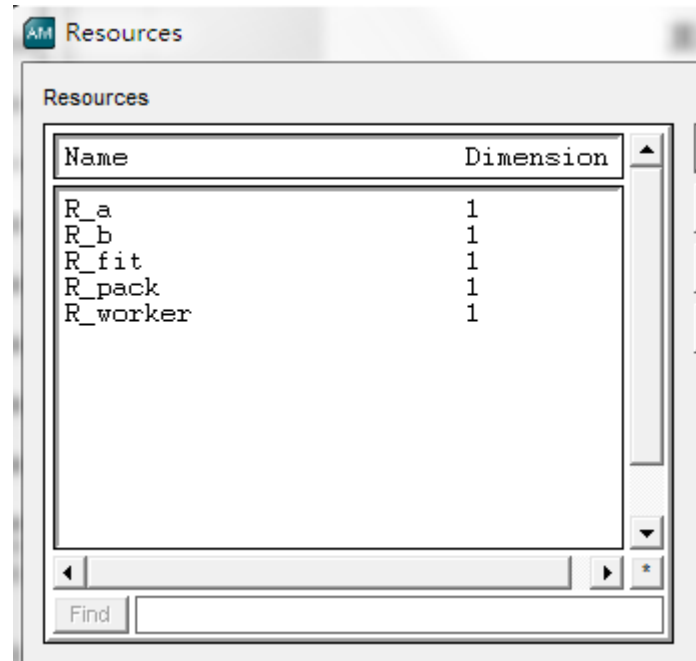


Loads 视窗

其中，只有 L\_a, L\_b, L\_demgen 需要设置 creation，只有 L\_a, L\_b, L\_pack, L\_product 需要设 Place。

## 5. 新建 Resources

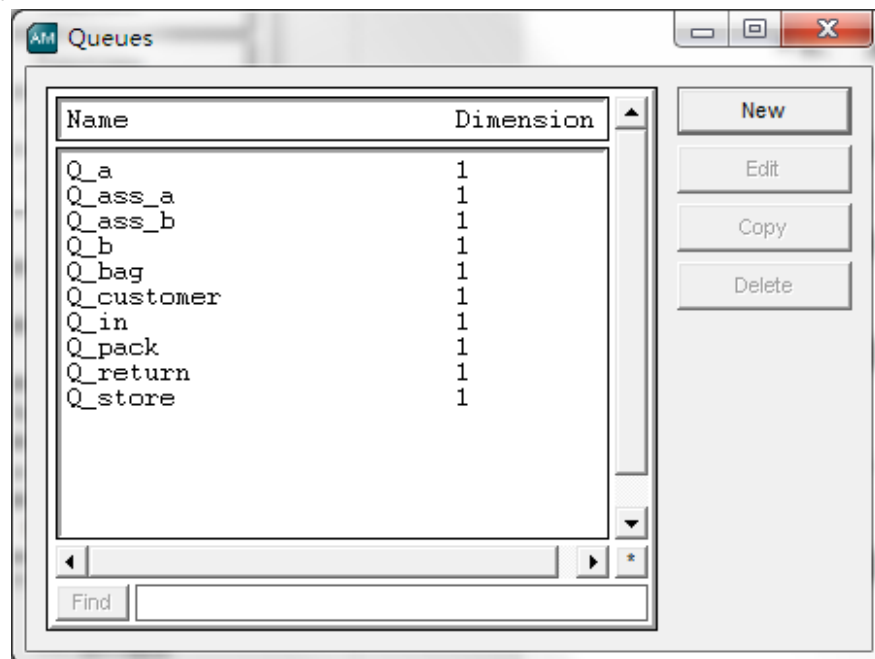
点击 Resources-New，新建 Resources 如下。



Resources 视窗

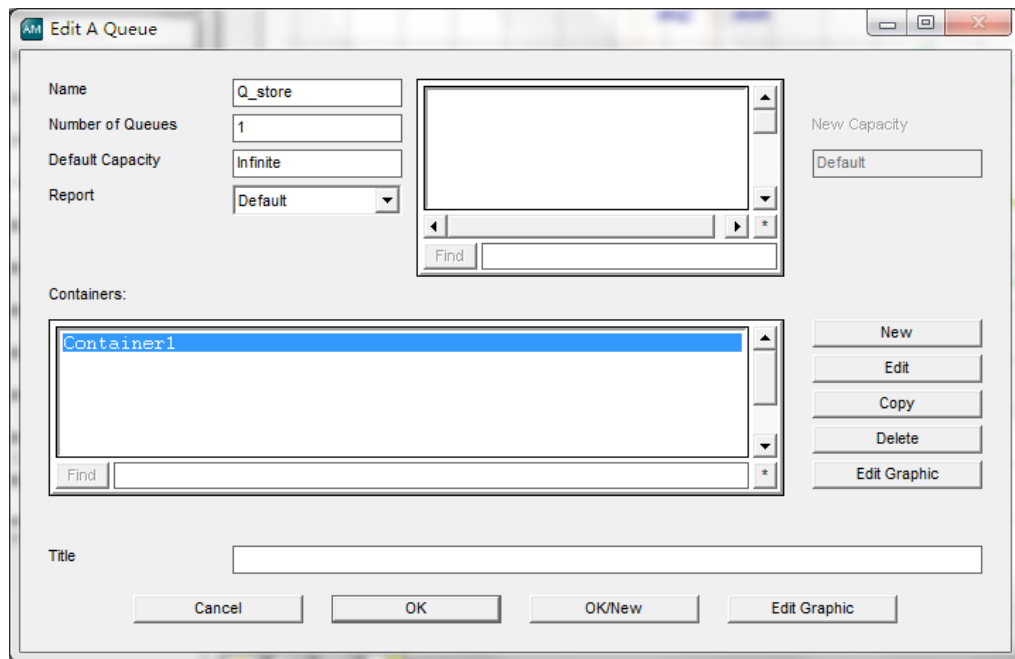
## 6. 新建 Queues

点击 Queues-New，新建队列如下。

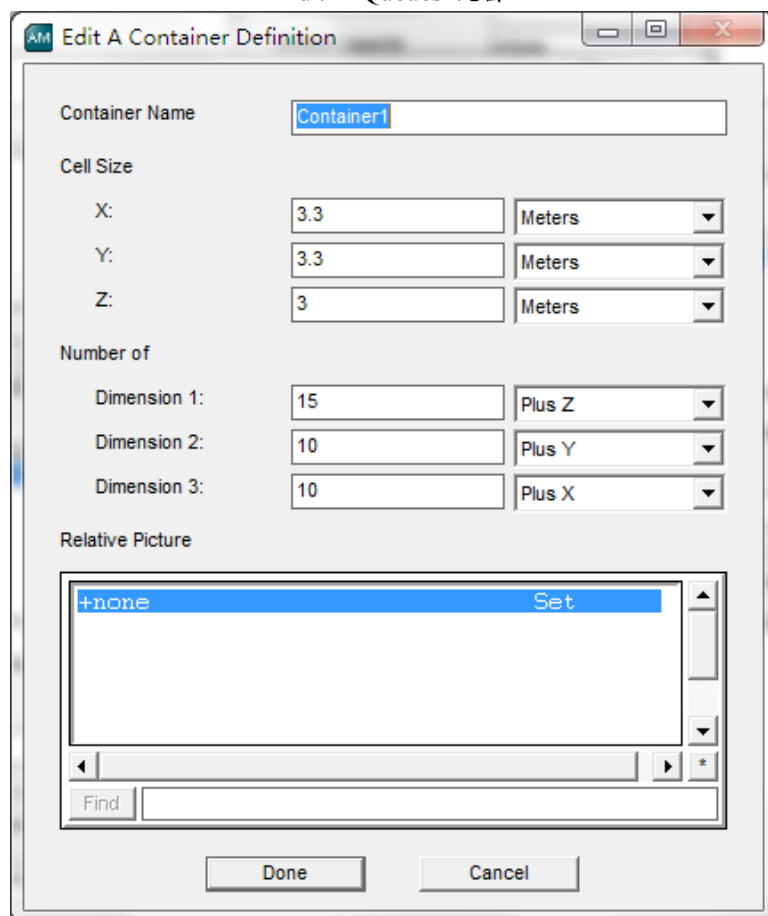


Queues 视窗

其中，Q\_store, Q\_bag, Q\_customer, Q\_return 可设置 Container 以便使库存更加形象。



Edit A Queues 视窗



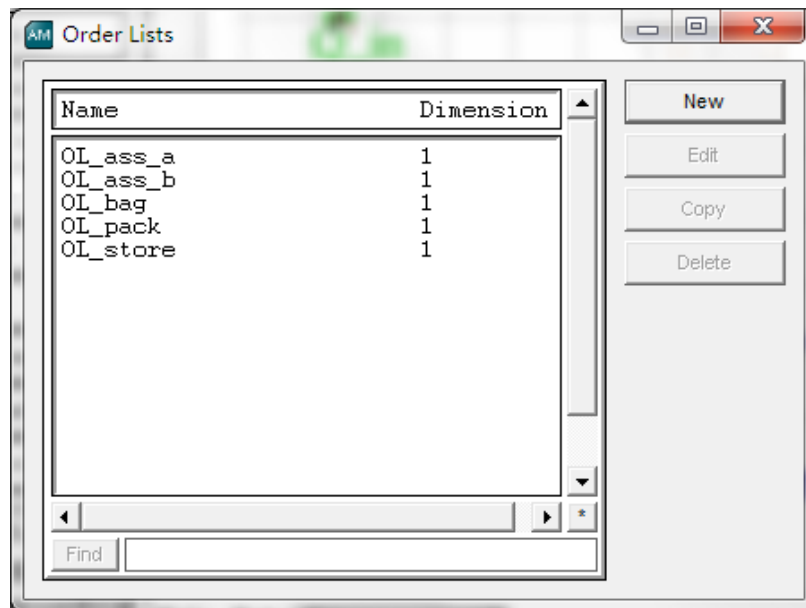
Edit A Container Definition 视窗

其中, cell size 可根据自己所设 load 大小来设置。留出余量以使 load 堆叠时有稍许间隙。同理, 另外 Q\_bag, Q\_customer, Q\_return 的设置与此类似。



## 7. 新建 Order List

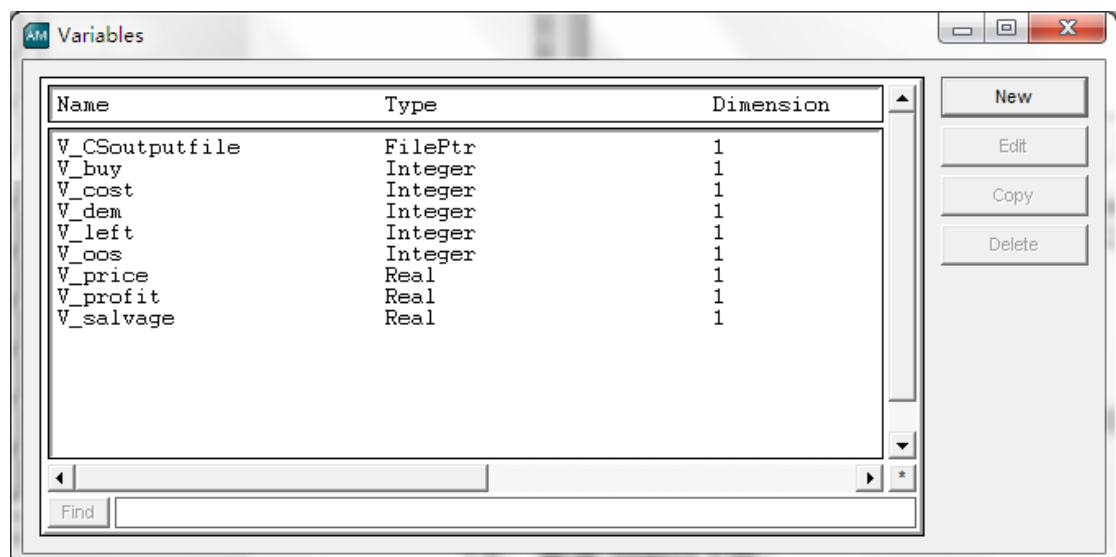
点击 Order List-New，新建 Order List 如下。



Order Lists 视窗

## 8. 新建 variables

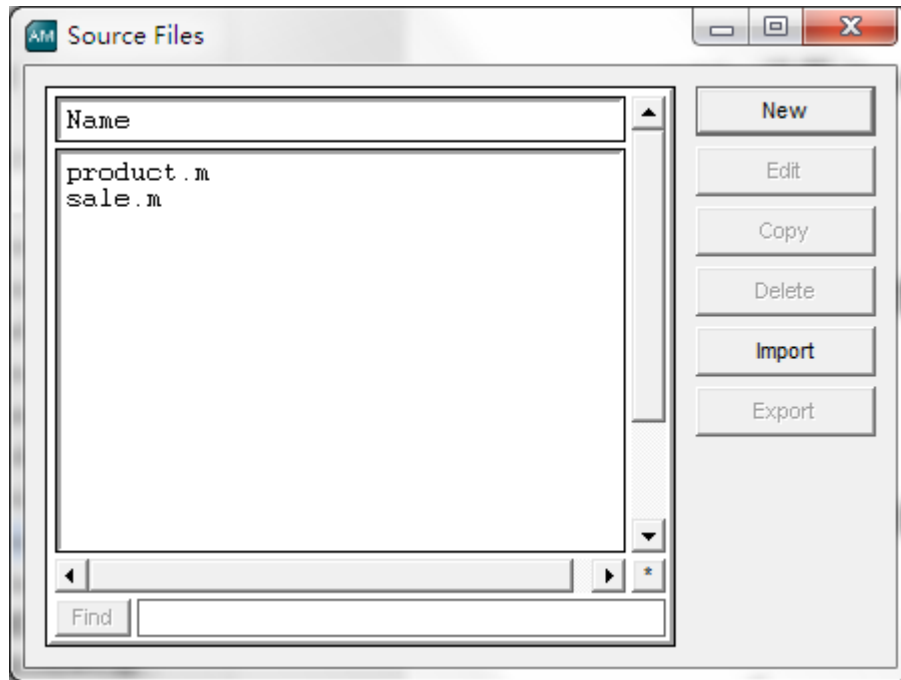
点击 Variables-New，新建 Variables 如下，注意变量类型。



Variables 视窗

## 9. 新建 Source File

点击 Source File-New，新建 Source File 如下。



Source Files 视窗

product.m 代码文件如下。

```
1begin P_in arriving
2    move into Q_in
3    use R_worker for e 1 min
4    send to oneof (95:P_match,5:die)
5end
6
7begin P_match arriving
8    move into conv.sta1
9    if load type = L_a then
10        begin
11            travel to conv.sta2
12            move into Q_a
13            use R_a for e 4 min
14            move into conv.sta4
15            travel to conv.sta6
16            move into Q_ass_a
17            wait to be ordered on OL_ass_a
18        end
19    if load type = L_b then
20        begin
21            travel to conv.sta3
22            move into Q_b
23            use R_b for u 5,2 min
24            move into conv.sta5
25            travel to conv.sta7
26            move into Q_ass_b
27            wait to be ordered on OL_ass_b
28        end
29end
30
```

```

30
31begin P_assfit arriving
32    if OL_ass_a current loads > 1 and OL_ass_b current loads > 1 then
33        create 1 load of L_fitdummy to P_fit
34end
35
36
37begin P_fit arriving
38    order 1 load from OL_ass_a to die
39    order 1 load from OL_ass_b to die
40    wait for n 5,1 min
41    create 1 load of load type L_product to P_pack
42    send to die
43end
44
45begin P_pack arriving
46    move into conv.sta8
47    travel to conv.sta9
48    move into Q_pack
49    if OL_pack current loads > 8 then
50        begin
51            create 1 load of load type L_packdummy to P_move
52        end
53        wait to be ordered on OL_pack
54end
55
56begin P_move arriving
57    order 10 loads from OL_pack to die
58    wait for n 4,0.5 min
59    create 1 load of load type L_pack to P_store
60    send to die
61end
62
63begin P_store arriving
64    move into pm.cpin
65    travel to pm.cpout
66    move into Q_store
67    wait to be ordered on OL_store
68end
69

```

sale.m 代码文件如下。

```

1begin model initialization function
2    set V_buy=150
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    order V_buy loads from OL_store to P_bag
12    send to P_demgen
13end
14
15begin P_bag 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 150,10
23    wait for 11 hr
24    send to P_sale
25end
26

```

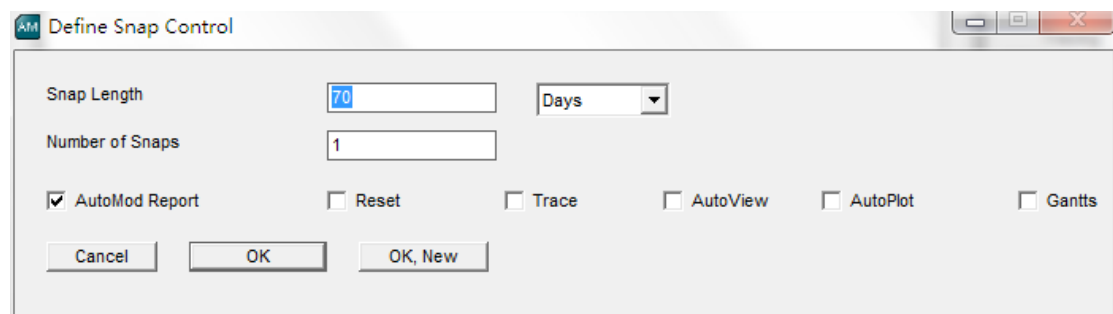
```

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
45end
46
47begin P_out arriving
48  move into Q_customer
49  wait for 5 days
50  send to die
51end
52
53begin P_return arriving
54  move into Q_return
55  wait for 5 days
56  move into Q_store
57end
58

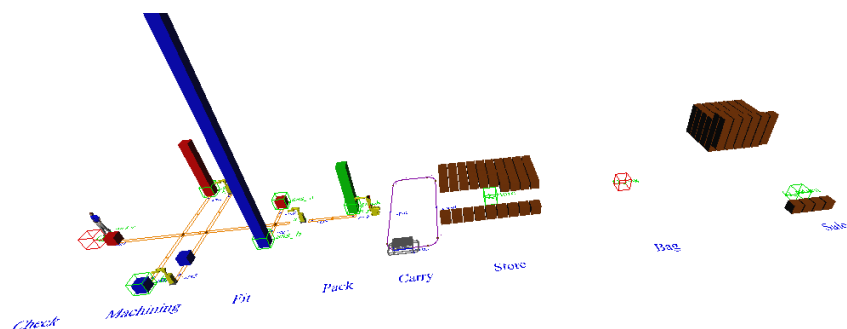
```

## 10. 设置 Run Control

点击 Run Control-New，新建 Run Control 如下。



## 11. 运行模型



## 12. 查看输出结果和输出报告

### Step12-1 查看 Report 文件:

```
*** AutoMod 12.6 ***
Model zonghe
Statistics at Absolute Clock = 70:00:00:00.00, Relative Clock = 70:00:00:00.00
CPU time: Absolute: 10.104 sec, Relative: 10.104 sec

Statistics for Process System "zonghe"

Process Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_in                 40401    0    0.58      --    10    0    --      87.35
P_match              38413    81    57.21     --   127    0    --     9007.16
P_fit                19166     1     0.95     --     7    0    --      299.74
P_pack               19165     5     4.64     --    12    0    --     1464.02
P_store              1916    559   277.89     --   559    0    --    877175.69
P_assfit             100800     0     0.00     --     1    0    --         0.00
P_move               1916     0     0.08     --     1    0    --      241.70
P_sale               10      0     0.00     --     1    0    --         0.00
P_out                1325     0    94.64     --   150    0    --    432000.00
P_buy                10      0     0.00     --     1    0    --         0.00
P_bag                1357     0     8.89     --   150    0    --     39600.00
P_demgen             10      0     0.07     --     1    0    --     39600.00
P_return             32      0     2.29     --    17    0    --    432000.00

Process Traffic Limit Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
P_in                 40401    0    0.58 Infinite    10    0    --      87.35
P_match              38413    81    57.21 Infinite   127    0    --     9007.16
P_fit                19166     1     0.95 Infinite     7    0    --      299.74
P_pack               19165     5     4.64 Infinite    12    0    --     1464.02
P_store              1916    559   277.89 Infinite   559    0    --    877175.69
P_assfit             100800     0     0.00 Infinite     1    0    --         0.00
P_move               1916     0     0.08 Infinite     1    0    --      241.70
P_sale               10      0     0.00 Infinite     1    0    --         0.00
P_out                1325     0    94.64 Infinite   150    0    --    432000.00
P_buy                10      0     0.00 Infinite     1    0    --         0.00
P_bag                1357     0     8.89 Infinite   150    0    --     39600.00
P_demgen             10      0     0.07 Infinite     1    0    --     39600.00
P_return             32      0     2.29 Infinite    17    0    --    432000.00

Queue Statistics
Name                Total    Cur  Average Capacity  Max  Min  Util  Av_Time  Av_Wait
=====
Space               183374     1     1.10 Infinite     7    0    --      36.44
Q_in                40401     0     0.59 Infinite    10    0    --      87.59
Q_a                 19244     1     3.10 Infinite    26    0    --     972.75
Q_b                 19169     2     2.42 Infinite    11    0    --     764.62
Q_ass_a             19242    76    28.59 Infinite   108    0    --     8985.72
Q_ass_b             19167     1    21.32 Infinite   113    0    --     6725.97
Q_pack              19165     5     4.50 Infinite    10    0    --     1420.63
Q_store             1916    559   277.86 Infinite   559    0    --    862675.94
Q_bag               1357     0     8.89 Infinite   150    0    --     39600.00
Q_customer           1325     0    94.64 Infinite   150    0    --    432000.00
Q_return            32      0     2.29 Infinite    17    0    --    432000.00

Order List Statistics
Name                Total    Cur  Average  Max  Min  Av_Time
=====
OL_ass_a            19242    76    28.59   108    0   8985.72
OL_ass_b            19167     1    21.32   113    0   6725.97
OL_pack             19165     5     4.50    10    0   1420.63
OL_store            1916    559   277.85   559    0  877068.89
OL_bag              1357     0     8.89   150    0   39600.00

Order List Backordered Statistics
Name                Total    Cur  Average  Max  Min  Fulfilled
=====
OL_ass_a             0      0     0.00     0    0         0
OL_ass_b             0      0     0.00     0    0         0
OL_pack              0      0     0.00     0    0         0
OL_store             0      0     0.00     0    0         0
OL_bag               0      0     0.00     0    0         0
```

## Order List Backordered Statistics

Name	Total	Cur	Average	Max	Min	Fulfilled
OL_ass_a	0	0	0.00	0	0	0
OL_ass_b	0	0	0.00	0	0	0
OL_pack	0	0	0.00	0	0	0
OL_store	0	0	0.00	0	0	0
OL_bag	0	0	0.00	0	0	0

## Resource Statistics

Name	Total	Cur	Average	Capacity	Max	Min	Util	Av_Time	Av_Wait	State
R_worker	40401	0	0.40	1	1	0	0.401	59.96	27.39	Up ----
R_a	19244	1	0.77	1	1	0	0.766	240.76	731.84	Up ----
R_b	19168	1	0.95	1	1	0	0.952	300.35	464.29	Up ----
R_fit	0	0	0.00	1	0	0	0.000	0.00	0.00	Up ----
R_pack	0	0	0.00	1	0	0	0.000	0.00	0.00	Up ----

## Random Number Streams

Name	Total
stream0	146141
stream_L_a_1	20236
stream_L_b_1	20167
stream_R_worker_1	0
stream_R_a_1	0
stream_R_b_1	0
stream_R_fit_1	0
stream_R_pack_1	0
stream_L_demgen_1	0
stream_L_fitdummy_1	0

Statistics for Conveyor System "zonghe.conv"

## Section Types Statistics

Name	Total	Cur	Average	Capacity	Max	Min	Util	Av_Time	Av_Wait
DefaultSection	95988	0	1.09	Infinite	8	0	--	68.99	

Statistics for Conveyor System "zonghe.conv"

## Section Statistics

Name	Total	Cur	Average	Capacity	Max	Min	Util	Av_Time	Av_Wait
sec1	76823	1	1.05	Infinite	7	0	--	83.00	
sec2	19244	0	0.19	Infinite	4	0	--	58.33	
sec3	19243	0	0.22	Infinite	5	0	--	69.11	
sec4	19169	0	0.17	Infinite	3	0	--	55.00	
sec5	19167	0	0.21	Infinite	1	0	--	65.44	
sec6	19242	0	0.05	Infinite	1	0	--	15.00	
sec7	19167	0	0.02	Infinite	1	0	--	5.00	
sec8	19165	0	0.14	Infinite	3	0	--	43.33	

## Station Statistics

Name	Total	Cur	Average	Capacity	Max	Min	Util	Av_Time	Av_Wait
sta1	38413	0	0.00	Infinite	2	0	--	0.26	
sta2	19244	0	0.43	Infinite	6	0	--	135.28	
sta3	19169	0	0.42	Infinite	3	0	--	131.70	
sta4	19243	0	0.00	Infinite	2	0	--	0.15	
sta5	19167	0	0.00	Infinite	1	0	--	0.00	
sta6	19243	1	0.49	Infinite	6	0	--	154.11	
sta7	19167	0	0.45	Infinite	1	0	--	140.44	
sta8	19165	0	0.00	Infinite	1	0	--	0.06	
sta9	19165	0	0.14	Infinite	3	0	--	43.33	

Station Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
conv:DefaultStation	191976	1	1.92 Infinite	10	0	--	60.58		

Photoeye Type Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
DefaultPhotoeye	0	0	0.00 Infinite	0	0	--	0.00		

Motor Type Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
DefaultMotor	1	1	1.00 Infinite	1	0	--	6048000.00		

Motor Statistics				
Name	Status	Total Down	%_Up	Av_Time Up
M_sec1	Up	0	1.00	6048000.00

Statistics for Path Mover System "zonghe.pm"

Guide Path Type Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
DefaultGuidePath	15329	1	1.00 Infinite	2	0	--	396.14		

Guide Path Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
path9	1916	0	0.02 Infinite	1	0	--	50.00		
path10	1916	0	0.00 Infinite	1	0	--	10.67		
path11	1916	0	0.02 Infinite	1	0	--	50.00		
path12	1917	1	0.96 Infinite	1	0	--	3032.14		
path13	1916	0	0.00 Infinite	1	0	--	6.38		
path14	1916	0	0.00 Infinite	1	0	--	6.38		
path15	1916	0	0.00 Infinite	1	0	--	6.38		
path16	1916	0	0.00 Infinite	1	0	--	5.80		

Control Point Type Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
DefaultControlPoint	5749	1	1.00 Infinite	2	0	--	1052.01		

Control Point Statistics									
Name	Total	Cur	Average Capacity	Max	Min	Util	Av_Time	Av_Wait	
cpin	1916	0	0.01 Infinite	1	0	--	40.05		
cpout	1916	0	0.02 Infinite	1	0	--	66.76		
cppark	1917	1	0.97 Infinite	1	0	--	3048.18		

Vehicle Statistics									
Delivering			Retrieving			Going To Park			Parking
Percent	Average	Percent	Average	Percent	Average	Percent	Average	Percent	
Of Total	Trips	Time/	Of Total	Trips	Time/	Of Total	Trips	Time/	Of Total
Time	Made	Trip	Time	Made	Trip	Time	Made	Trip	Time
DefVehicle:									
0.021	1916	66.76	0.013	1916	40.05	0.010	1916	30.05	0.957
All DefVehicle:									
0.021	1916	66.76	0.013	1916	40.05	0.010	1916	30.05	0.957

The Average Capacity Lost Due To Congestion is 0.0 Percent  
Of Total Capacity Which is 0.0 Vehicles

Vehicle Segment Statistics						
Vehicle	Total	Cur	Average Capacity		Max	Min
=====						
DefVehicle:						
DefVehicle(1)	1	1	1.00	Infinite	1	0
Current Status Of Vehicles						
Vehicle	Status	Time In Current Cur Status Location		Destination Location	Load On or Claimed By Veh	
=====						
DefVehicle(1)	Idle	1185.11	cppark	cppark	Empty	
Current Status Summary						
Type	Average Time In Cur Status	Maximum Time In Cur Status	Minimum Time In Cur Status			
=====						
DefVehicle	1185.11	1185.11	1185.11			

从报告中，我们可以看出各个 Queues、Process 等中的 load 的数量。

## Step12-2 查看输出文件

```

817.590000
745.670000
898.500000
889.510000
898.500000
826.580000
844.560000
898.500000
898.500000
898.500000

```

通过改变进货量，我们可以得到最佳进货量及最大利润。