7th International Conference
″Research and Development in Mechanical Industry″
RaDMI 2007
16 - 20. September 2007, Belgrade, Serbia

BUSSINES SIMULATING PROCES FOR THE PRODUCTION SURROUND, USING QUEUEING SYSTEM SIMULATION WITH WINQSB

Radomir Brzaković[1], Zoran Marjanović[2], Željko Radulović[3]
[1] Zastava automobili, PJ IS, Kragujevac, SERBIA, E-mail: brzijax@yahoo.com
[2] Zastava automobili, Institute for automobile , Kragujevac, SERBIA
[3] Zastava automobili, PJ IS, Kragujevac, SERBIA, E-mail: rzeljko2004@yahoo.com

Summary : *The objective of this work is an attempt to resolve aproblem of increasing production in one factory which field of the production is production electric switchers. Also, the objective of this work is to improve the quantity of the final products, in order to increase the profit for the company. Using the tools necessary for performing a simulation model, it is implemented a QUEUEING SYSTEM SIMULATION WITH WINQSB.*

Keywords : *bussines simulation,proces,winqsb,queue*

1. INTRODUCTION

1.1. Description of the problem area and background

In this project I will try to resolve problem of increasing production in one factory which field of the production is production electric switchers.
The switchers which are predicted in this company are consist of following parts:

- Plastic Body
- Plastic part for finger pressing
- Metal part - which gives user possibility to put harder switcher on the wall.
- Electric part - which gives switcher connection with the electric installation.

In order to resolve this problem I will imagine one simplified model of these production lines, which will help me to experiment with this system.
The simplified technological process has following fazes and parts:

- Machine1- we are cutting and giving form to the plastic part for the body of the switcher.
- Machine2- we are cutting and giving form to the plastic part for the active pressing part, which is produced, from the different part of the material.
- Machine4- we are giving form to the metal part of the switcher.
- Machine6- we are cutting and giv~ng form to the electric part of the switcher.
- Machine3- after both plastic parts get wished form they are connected on the machine3.
- Machine5 - after metals parts get form and plastic parts are connected, we are putting all in the same piece.
- Machine7-Will connected electrical part with the rest of the switcher.
- Machine8 - is used for packing the final product for delivery.

1.2. The model and simulation software selected

My goal, is to improve the quantity of the final products, in order to increase the profit for the company. Because it is not so expensive and I have the tools necessary for performing a simulation model, I decide to use

QUEUEING SYSTEM SIMULATION WITH WINQSB.

Queuing simulation is an example of discrete event simulation.

- Discrete event simulation models can be used to delve into the fine details of complex systems with many interactions. These models are most commonly used to create detailed operational systems representing demands among activities requiring scarce resources over time.
- Once built, a detailed discrete event simulation model is often far more convenient and risk-free to manipulate than the real-world system itself.
- Because of the dynamic nature of discrete event simulation models, we must keep track of the current value of simulated time as the simulation proceeds, and we also need a mechanism to advance simulated time from one value to another. In principal there are two approaches, the *next-event time advance* and the *fixed-increment time advance.*
    - With the *next-event time advance* approach, the simulation clock is initialized to zero and the times of occurrence of future events are determined. The simulation clock is then advanced to the time of occurrence of the most imminent (first) of these future events, at which point the state of the system is updated to account for the fact that an event has occurred. This process of advancing the simulation clock from one event to another is continued until eventually some pre-specified stopping condition is satisfied. This approach is used by all major simulation languages and by most people coding their model in a general-purpose language (in fact the fixed-increment time advance to be discussed below is a special case of this one).
    - With the *fixed-increment time advance*, the simulation clock is advanced in increments of exactly Δt time units for some appropriate choice of Δt. After each update of the clock, a check is made to determine whether any event has occurred during the previous interval of length Δt. If one or more events is scheduled to occur during this interval, these events will be considered to occur at the *end* of the interval and the system state is updated accordingly. The primary purpose of this approach appears to be for systems where it can reasonably be assumed that all events actually occur at one of the times $n\Delta t$ (n=0,1,2,) for an appropriately chosen Δt.
- There is a logical organization for the components of a discrete event simulation model:
    - The simulation starts at time 0 with the main program invoking the initialization routine, where the simulation clock is set to zero, the system state, the statistical counters and the event list are initialized.
    - After control has been returned to the main program, it evokes the time routine to determine which type of event is most imminent.
    - If an event of type *I* is the next to occur, the simulation clock is advanced to the time that event type *I* will occur and control is returned to the main program.
    - Then the main program invokes event routine i, where typically three types of activities occur:

1. The system state is updated to account for the fact that an event of type *i* has occurred;
2. Information about system performance is gathered by updating the statistical counters;
3. The times of occurrence of future events are generated and this information is added to the event list.

    - After all processing has been completed, either in event routine *i* or in the main program, a check is made to determine whether the simulation should now be terminated. In this case the report generator is invoked from the main program to compute estimates of the desired measures of performance and to produce a report. Otherwise control is passed back to the main program.

- Recall from queuing theory that in essence all queuing systems can be broken down into individual sub-systems consisting of entities queuing for some activity (as shown in Figure 1).

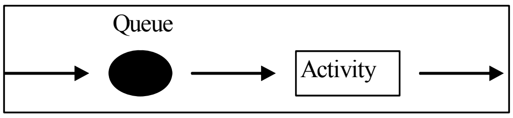


Figure 1:

- Typically we can talk of this individual sub-system as dealing with *customers* queuing for **service**. To analyse this sub-system we need information relating to:
    - **Arrival process:**
        - how customers arrive e.g. singly or in groups (batch or bulk arrivals)
        - how the arrivals are distributed in time (e.g. what is the probability distribution of time between successive arrivals (the ***interarrival time distribution***))
        - Whether there is a finite population of customers or (effectively) an infinite number
    - **Service mechanism:**
        - a description of the resources needed for service to begin

- how long the service will take (the ***service time distribution***)
- the number of servers available
- whether the servers are in series (each server has a separate queue) or in parallel (one queue for all servers)
- whether pre-emption is allowed (a server can stop processing a customer to deal with another "emergency" customer)

- **Queue characteristics:**
  - how, from the set of customers waiting for service, do we choose the one to be served next (e.g. FIFO (first-in first-out) also known as FCFS (first-come first served); LIFO (last-in first-out); randomly) (this is often called the *queue discipline*)
  - Do we have:
    - Balking (customers deciding not to join the queue if it is too long)
    - Reneging (customers leave the queue if they have waited too long for service)
    - Jockeying (customers switch between queues if they think they will get served faster by so doing)
    - a queue of finite capacity or (effectively) of infinite capacity

• Note here that integral to queuing situations is the idea of uncertainty in, for example, interracial times and service times. This means that probability and statistics are needed to analyse queuing situations.

• Whilst queuing theory can be used to analyse simple queuing systems more complex queuing systems are typically analysed using **simulation** (more accurately called **discrete-event simulation**).

2. QUEUEING SYSTEM FOR THIS PROBLEM

The queuing system for this problem is the one presented in Figure 2, where I have three parts X, Y, Z and J that are being assembled together.

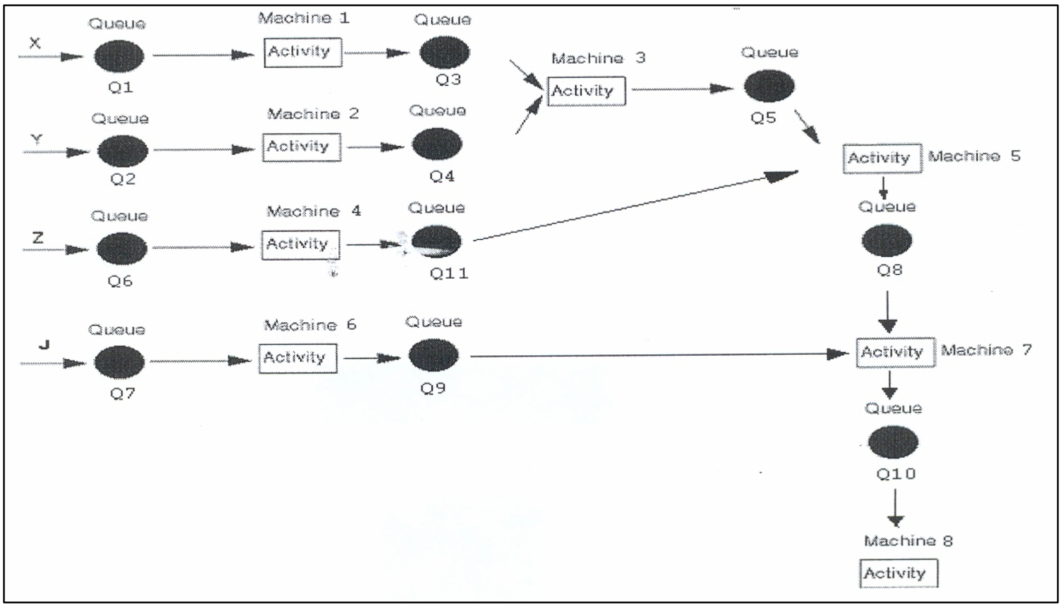


Figure 2:

• Before assembly this three parts, machines must undergo some processing's:

- X goes to machine1;
- Y goes to machine2;
- Z goes to machine4;
- J goes to machine6;
- X and Y go to machine3, where they are assembled;
- X, Y and Z go to machine5, where they are assembled;

- X, Y, Z and J go to machine7, where the final product is finalized;
- X, Y, Z and J, like a final product, go to machine 8 for delivery.

- X, Y, Z and J are the customers flowing through the system and the machines are servicing these customers (in this case, X represents plastic part of the switcher-the body, Y represents the another plastic part - the active part (for the pressing), Z represents the metal part - on the switcher and J represents the electric part - on the switcher)
- For the customers (X, Y, Z, J) we need to specify an interarrival time distribution.
  Here we have assumed that:
  - X parts have interarrival times which are uniformly distributed between 0.2 and 1.0 hours;
  - Y parts have interarrival times which are uniformly distributed between 0.6 and 1.0 hours;
  - Z parts have interarrival times that are normally distributed with mean 0.6 hours and_standard deviation 0.2 hours;
  - J parts have interarrival times that are normally distributed with mean 0.6 hours and_standard deviation 0.01 hours;

For the servers (Machines 1 to 8) I need to specify the service time distribution - for the parts that they process. Here:

- Machine1 processes X parts with a service time that is normally distributed with mean 0.1 hours and standard deviation 0.01 hours.
- Machine2 processes Y parts with a service time that is normally distributed with mean 0.6 hours and standard deviation 0.02 hours.
- Machine3 is the machine that assembles parts X and Y together so above I have specified it processes both parts with the same service time distribution (here a constant processing time of 0.2 hours).
- Machine4 processes Z parts with a service time that is normally distributed with mean 0.45 hours and standard deviation 0.02 hours.
- Machine5 is the machine that assembles parts X, Y and Z together so above I have specified it processes both parts with the same service time distribution (here a constant processing time of 0.2 hours).
- Machine6 processes J parts with a service time that is normally distributed with mean 0.2 hours and standard deviation 0.02 hours.
- Machine 7 is the machine that assembles parts X, Y, Z and J together so above I have specified it processes both parts with the same service time distribution (here a constant processing time of 0.2 hours).
- Machine 8 processes the assembled part and, I have specified the processing time for both parts as the same - here normally distributed with mean 0.8 hours and standard deviation 0.33 hours.

For all queues I have specified the queue discipline as FIFO (first-in first-out) meaning that parts are processed in the order in which they arrive in the queues. I have defined queue capacities. Once a queue is full then the preceding activity cannot pass on a part (customer) until there is space available in the queue:

- Capacity of Ql: 30 units
- Capacity of Q2: 30 units
- Capacity of Q3: 50 units
- Capacity of Q4: 50 units
- Capacity of Q5: 30 units
- Capacity of Q6: 10 units
- Capacity of Q7: 10 units
- Capacity of Q8: 100 units
- Capacity of Q9: 100 units
- Capacity of Q10: 150 units
- Capacity of Q11: 150 units

2.1. Analysing the system using the simulation module in the WINQSB/QSS

The objectives of the simulation are:

- To obtain useful insight into the behaviour of the manufacturing system
- To identify the bottleneck point in the production process which generates very long queue or very crowded machines
- To explore some policies in order to increase production of the manufacturer system

2.1.1. WINQSB input

• The initial package input screen is shown in Figure 3. I have 23 components as there are 4 items being processed (X, Y, Z and J), 11 queues and 8 activities (so 4+11+8=23 components).

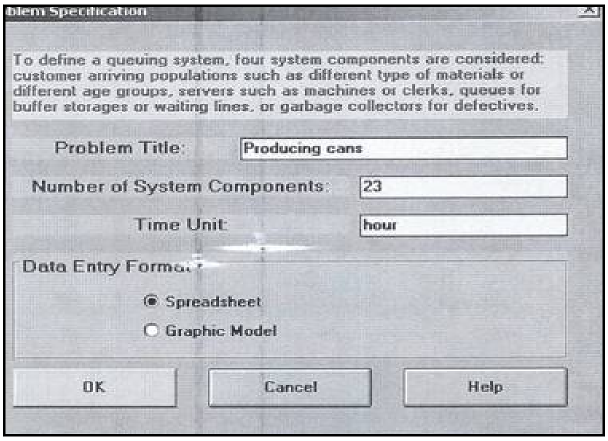


Figure 3:

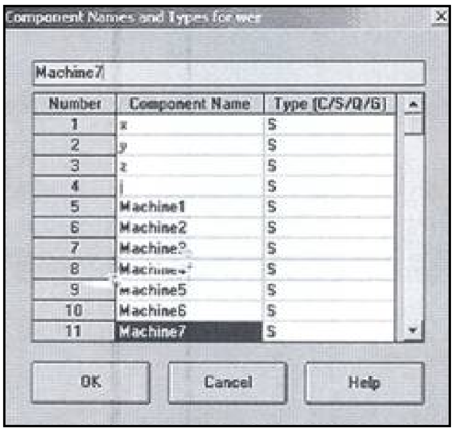
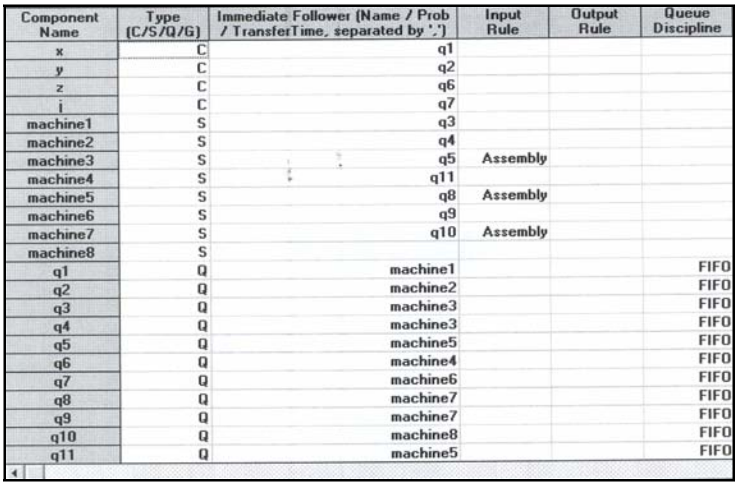


Figure 4:



Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline
x	C	q1			
y	C	q2			
z	C	q6			
j	C	q7			
machine1	S	q3			
machine2	S	q4			
machine3	S	q5	Assembly		
machine4	S	q11			
machine5	S	q8	Assembly		
machine6	S	q9			
machine7	S	q10	Assembly		
machine8	S				
q1	Q	machine1			FIFO
q2	Q	machine2			FIFO
q3	Q	machine3			FIFO
q4	Q	machine3			FIFO
q5	Q	machine5			FIFO
q6	Q	machine4			FIFO
q7	Q	machine6			FIFO
q8	Q	machine7			FIFO
q9	Q	machine7			FIFO
q10	Q	machine8			FIFO
q11	Q	machine5			FIFO

Figure 5:

- The "Immediate Follower" column is used to define how customers (X, Y, Z and J in this case) flow thorough the system.
  - X goes to Q1, so that is its immediate follower above.
  - Y goes to Q2, so that is its immediate follower above.
  - Z goes to Q6, so that is its immediate follower above.
  - J goes to Q7, so that is its immediate follower above.
  - Q1 is followed by Machine1
  - Q2 is followed by Machine2
  - Q3 is followed by Machine3
  - Q4 is followed by Machine3
  - Q5 is followed by Machine5
  - Q6 is followed by Machine4
  - Q7 is followed by Machine6
  - Q8 is followed by Machine7
  - Q9 is followed by Machine7
  - Q10 is followed by Machine8
  - Q11 is followed by Machine5
  - Machine1 is followed by Q3
  - Machine2 is followed by Q4
  - Machine3 is followed by Q5
  - Machine4 is followed by Q11
  - Machine5 is followed by Q8
  - Machine6 is followed by Q9
  - Machine7 is followed by Q 10
- The "Input Rule" column is used to tell each server how to select a customer from the preceding queues. This column is only of significance when a server has more than one queue immediately preceding it. In my example this occurs at Machine3, where I have Q3 contain X parts and Q2 containing Y parts, Machine5, where I have Q11 contain Z parts and Q5 containing X+Y parts and at Machine7, where I have Q9contain J parts and Q8 containing X + Y + Z parts. The use of the word "Assembly" as the input rule for Machine3, Machine5 and Machine7 therefore says that one of each part must be available in Q3 and Q4 before the Machine3 activity can start, Machine5 therefore says that one of each part must be available in Q5 and Q11 before the Machine5 activity can start and Machine7 therefore says that one of each part must be available in Q8 and Q9 before the Machine7 activity can start.

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob	Input Rule	Queue Discipline	Queue Capacity	Interarrival Time Distribution	Service Time Distribution
x	C	q1				UNIFORM/0.2/1.0	
y	C	q2				UNIFORM/0.6/0.6	
z	C	q6				NORMAL/0.6/0.2	
j	C	q7				NORMAL/0.6/0.01	
machine1	S	q3					x/NORMAL/0.1/0.01
machine2	S	q4					y/NORMAL/0.6/0.02
machine3	S	q5	Assembly				x/CONSTANT/0.2,y/CONSTANT/0.2
machine4	S	q11					z/NORMAL/0.45/0.02
machine5	S	q8	Assembly				STANT/0.2,y/CONSTANT/0.2,z/CONSTANT/0.2
machine6	S	q9					j/NORMAL/0.2/0.02
machine7	S	q10	Assembly				constant/0.2,z/CONSTANT/0.2,j/CONSTANT/02
machine8	S						}/0.33,z/NORMAL/0.8/0.33,j/NORMAL/0.8/0.33
q1	Q	machine1		FIFO	30		
q2	Q	machine2		FIFO	30		
q3	Q	machine3		FIFO	50		
q4	Q	machine3		FIFO	50		
q5	Q	machine5		FIFO	30		
q6	Q	machine4		FIFO	10		
q7	Q	machine6		FIFO	10		
q8	Q	machine7		FIFO	100		
q9	Q	machine7		FIFO	100		
q10	Q	machine8		FIFO	150		
q11	Q	machine5		FIFO	200		

Figure 6:

A graphical representation of the system,produced by selecting from menu FORMAT and the SWITCH TO GRAPHIC MODEL, can be seen in Figure 7.

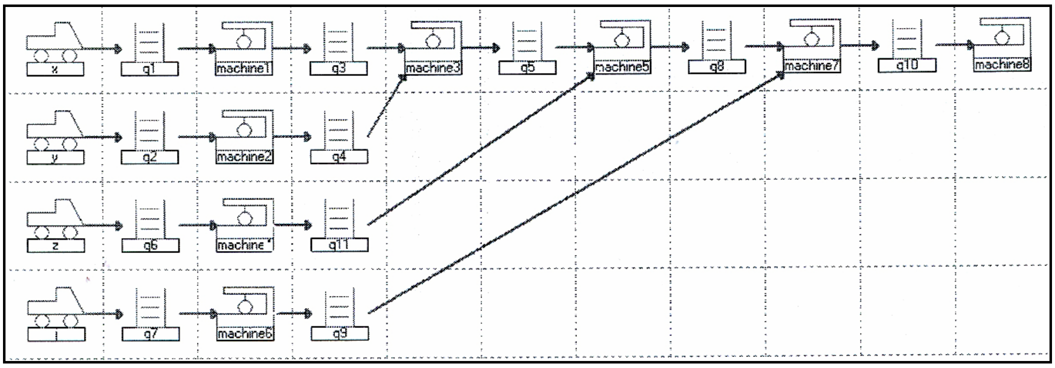


Figure 7:

2.1.2. WINQSB solution

The initial screen provided by WINQSB by selecting SOLVES and ANALYZE, and then PERFORM SIMULATION, is shown in Figure 8. I have chosen to simulate for 100 hours . This is to allow the system to "fill up".

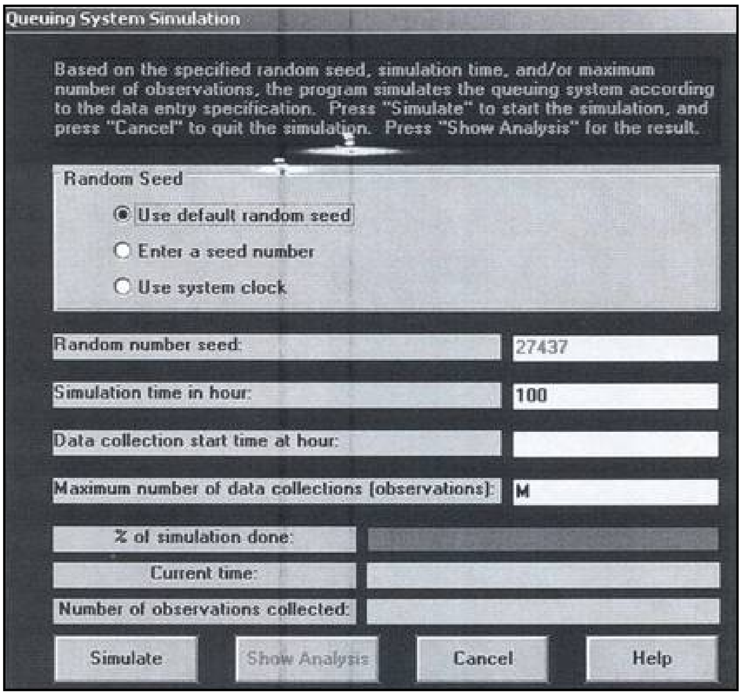


Figure 8:

2.1.3. Output Analysis

• The initial WINQSB output after simulation is shown in Figure 9. It can be seen that I have 127 observations to analyse. The simulation ended at T=100.1001, where as I actually specified I wished to simulate for 100 hours. The reason for this is that the WINQSB program runs until:

- it encounters an event (such as a new part appearing, an end of service) that necessitates a change in the system;
- the simulation time T is >= 100 (the simulation time we specified).

It then stops - here the first event necessitating a change in the system occurred at T=100.1001.

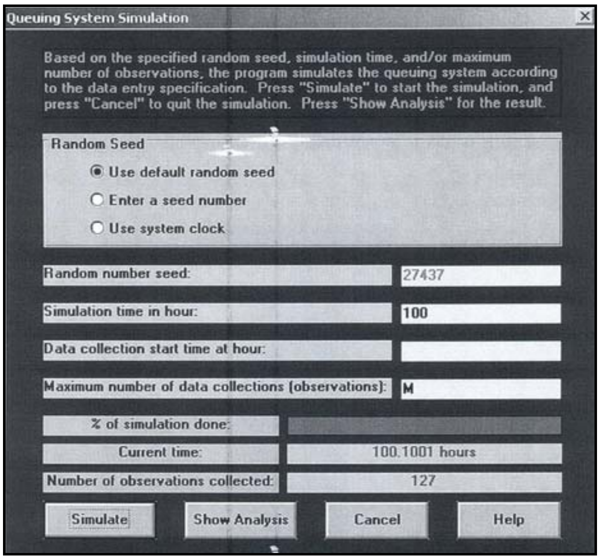


Figure 9:

The analysis produced by WINQSB as a result of this simulation is shown in Figure 10:

- It can be seen that 164 X parts, 166 Y parts, 165 Z parts and 166 J parts arrived in the system.
- The average (time-weighted) number in the system for X was 17.9724, for Y was 3.0310, for Z was 2.3696 and for J was 3.7424.
- 127 items have been finished
- It can be seen above that the package has designated these as X items. Hence we need to concentrate on X - the statistics above for X will refer to the flow through the system of X parts as they turn into assembled items and are eventually passed out of the system. Indeed this is why many of the statistics above for Y are given as zero.
- The average process time, time spent processing a X part as it flows through the system is 2.7153 hours with a standard deviation of 0.3153 hours.
- The average waiting time t experienced by a X part is 13.3000 hours with a standard deviation of 6.7898 hours. Theoretically the average processing time and the average waiting time should add to the average flow time (given above as 12.6030). Here there is a slight discrepancy due to the way the package internally keeps track of parts as they move through the system.

01-30-2003	Result	x	y	z	i	Overall
1	Total Number of Arrival	164	166	165	166	661
2	Total Number of Balking	0	0	0	0	0
3	Average Number in the System (L)	17.9724	3.0310	2.3696	3.7424	27.1154
4	Maximum Number in the System	37	7	7	8	59
5	Current Number in the System	37	3	3	4	47
6	Number Finished	127	0	0	0	127
7	Average Process Time	2.7153	0	0	0	2.7153
8	Std. Dev. of Process Time	0.3153	0	0	0	0.3153
9	Average Waiting Time (Wq)	13.3000	0	0	0	13.3000
10	Std. Dev. of Waiting Time	6.7898	0	0	0	6.7898
11	Average Transfer Time	0	0	0	0	0
12	Std. Dev. of Transfer Time	0	0	0	0	0
13	Average Flow Time (W)	12.6030	0	0	0	12.6030
14	Std. Dev. of Flow Time	6.2467	0	0	0	6.2467
15	Maximum Flow Time	23.4013	0	0	0	23.4013
	Data Collection: 0 to	100 hours				
	CPU Seconds =	2.6100				

Figure 10:

Figure 11 contains information about the utilization of the servers (machines) obtained from RESULTS and SHOW SERVER ANALYSIS.

01-30-2003	Server Name	Server Utilization	Average Process Time	Std. Dev. Process Time	Maximum Process Time	Blocked Percentage	# Customers Processed
1	machine1	16.24%	0.0996	0.0097	0.1240	0.00%	163
2	machine2	99.28%	0.6017	0.0199	0.6575	0.00%	165
3	machine3	32.40%	0.2000	0.0002	0.2000	0.00%	162
4	machine4	73.64%	0.4463	0.0213	0.5000	0.00%	165
5	machine5	32.40%	0.2000	0.0002	0.2000	0.00%	162
6	machine6	32.89%	0.1981	0.0207	0.2650	0.00%	166
7	machine7	32.40%	0.2000	0.0002	0.2000	0.00%	162
8	machine8	97.77%	0.7699	0.3133	1.5243	0.00%	127
	Overall	52.13%	0.3278	0.2369	1.5243	0.00%	1272
Data	Collection:	0 to	100	hours	CPU	Seconds =	2.6100

Figure 11:

- The program shows that most utilization machines are Machine2 and Machine8.

WINQSB can also produce an analysis of the queues, as in Figure 12.

01-30-2003	Queue Name	Average Q. Length (Lq)	Current Q. Length	Maximum Q. Length	Average Waiting (Wq)	Std. Dev. of Wq	Maximum of Wq
1	q1	0	0	1	0	0	0
2	q2	0.4654	0	1	0.2804	0.1263	0.4405
3	q3	0.2565	0	3	0.1574	0.3212	1.3943
4	q4	1.5720	2	6	0.9598	0.9001	3.2005
5	q5	0.0931	0	2	0.0574	0.1601	0.8613
6	q6	0.0753	0	2	0.0457	0.1127	0.8796
7	q7	0	0	1	0	0	0
8	q8	0	0	1	0	0	0
9	q9	3.4215	4	8	2.0900	0.9024	4.3640
10	q10	15.3615	34	34	9.6516	6.0629	18.9468
11	q11	1.5648	3	6	0.9540	0.8539	2.8989
	Overall	22.8102	43	34	1.1197	2.9983	18.9468
Data	Collection:	0 to	100	hours	CPU	Seconds =	2.6100

Figure 12:

- Here Q10 is the largest queue, confirming our observation above that Machine8 is the bottleneck in the production process.

- As well as the above tables of numbers can be produced graphical analysis, as in Figure 14 for example.
- Producing graphical analyses is common in discrete-event simulation. It often enables us to get a clearer picture of the system and how it is performing than we would get by just staring at tables of numbers.

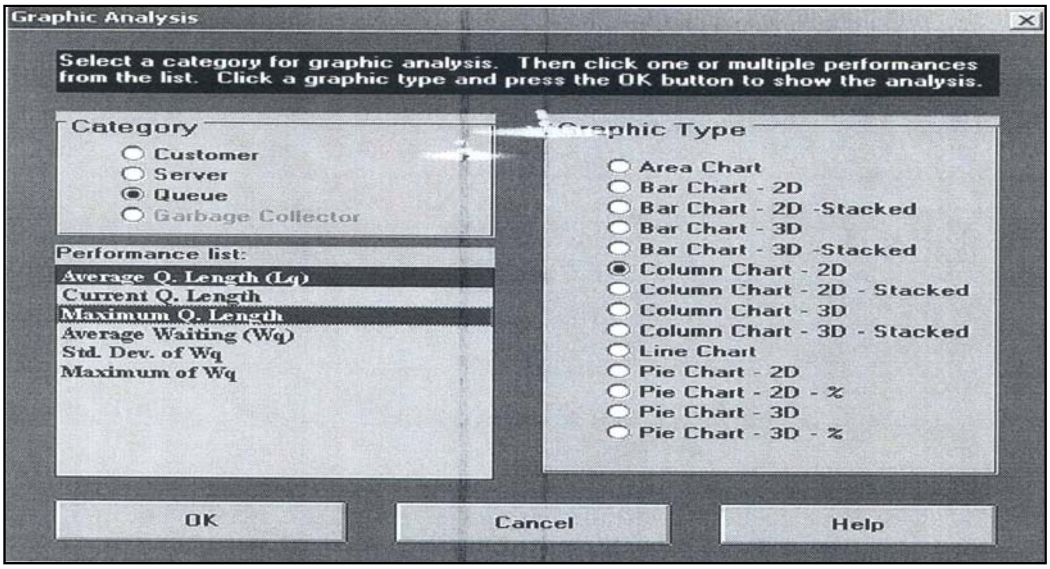


Figure 13:

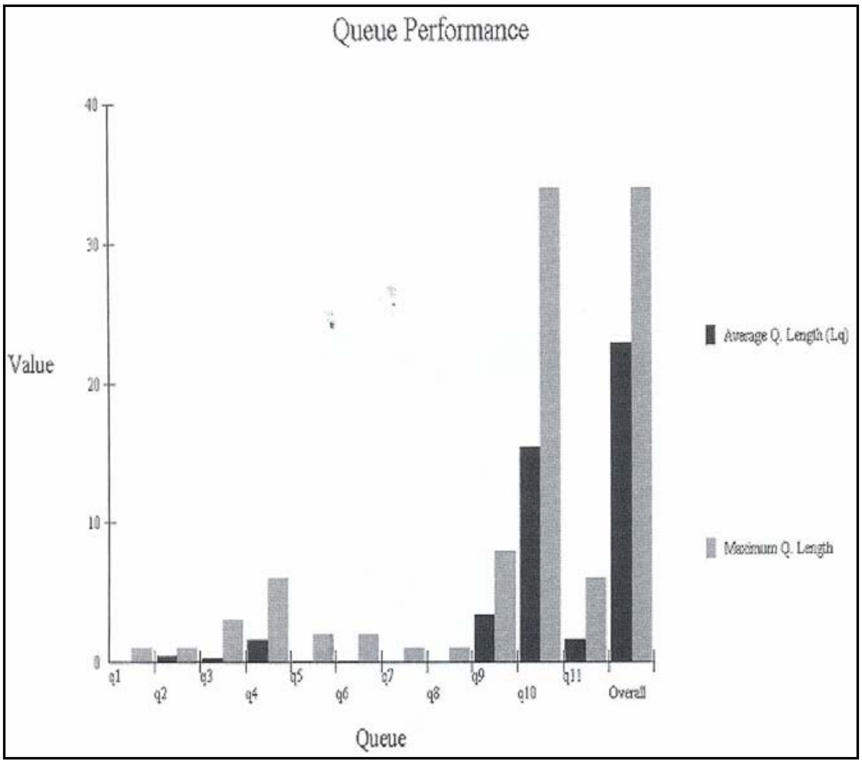


Figure 14:

2.2. Changing the system

- Once I have a simulation model for a system, such as above, then using it can give us useful insight into the behavior of the system. Real-life systems can be complex with many interactions and it can be difficult, given statistical variations in processing and arrival times, queue capacities, etc to understand the entire system without the numerical insight provided by simulation.

- In my system above, for example, it has become clear that Machine 8 is a bottleneck in terms of the overall production output. I might therefore be interested in seeing how I can improve the system
- Examining my simulation situation, as below, one option here is to change the queue discipline for Q 10, which is the queue just before Machine 8, the bottleneck.

2.2.1. Changing the queue discipline for Q10

- Suppose I change that queue discipline from FIFO, the current discipline, to (in package terminology) MaxWorkDone - choose the next item for processing to be the one that has had the maximum work (processing) carried out on it already.

Component Name	Type (C/S/Q/G)	Immediate Follower (Name / Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity
x	C	q1				
y	C	q2				
z	C	q6				
i	C	q7				
machine1	S	q3				
machine2	S	q4				
machine3	S	q5	Assembly			
machine4	S	q11				
machine5	S	q8	Assembly			
machine6	S	q9				
machine7	S	q10	Assembly			
machine8	S					
q1	Q	machine1			FIFO	30
q2	Q	machine2			FIFO	30
q3	Q	machine3			FIFO	50
q4	Q	machine3			FIFO	50
q5	Q	machine5			FIFO	30
q6	Q	machine4			FIFO	10
q7	Q	machine6			FIFO	10
q8	Q	machine7			FIFO	100
q9	Q	machine7			FIFO	100
q10	Q	machine8			MaxWorkDone	150
q11	Q	machine5			FIFO	200

Figure 15:

In Figure 16 are shown the WINQSB results for the new inputs.

01-30-2003	Result	x	y	z	i	Overall
1	Total Number of Arrival	164	166	165	166	661
2	Total Number of Balking	0	0	0	0	0
3	Average Number in the System (L)	17.9724	3.0310	2.3696	3.7424	27.1154
4	Maximum Number in the System	37	7	7	8	59
5	Current Number in the System	37	3	3	4	47
6	Number Finished	127	0	0	0	127
7	Average Process Time	2.7276	0	0	0	2.7276
8	Std. Dev. of Process Time	0.3140	0	0	0	0.3140
9	Average Waiting Time (Wq)	6.7113	0	0	0	6.7113
10	Std. Dev. of Waiting Time	5.4345	0	0	0	5.4345
11	Average Transfer Time	0	0	0	0	0
12	Std. Dev. of Transfer Time	0	0	0	0	0
13	Average Flow Time (W)	5.7025	0	0	0	5.7025
14	Std. Dev. of Flow Time	5.3024	0	0	0	5.3024
15	Maximum Flow Time	30.0325	0	0	0	30.0325
	Data Collection: 0 to	100 hours				
	CPU Seconds =	2.8590				

Figure 16:

- Comparing Figure 10 with Figure 16 it can be seen that I finish exactly the same number of items. This is as we might expect as the processing times on Machine 8, for which I have changed the queue discipline for item choice, are NOT associated with the items chosen but (conceptually) are taken from a predefined list of processing times - as in our simple example above.

- Hence changing the order in which we choose items for processing on Machine 8 does not make any difference to the total number completed. What it does affect however is the flow time. Now the average flow time is given as 12.8057 hours, before it was 5.7025 hours.
- Conceptually individual items are spending less time in the system on average, although the total system output is unchanged.

2.2.2. Get another Machine 8

- One option I might explore to try and achieve more output is to get another Machine 8. Two machines at that point in the production system should enable us to substantially increase output. Indeed, since Machine 8 is the bottleneck common sense might well say that two machines would enable us to DOUBLE production.
- I can investigate this situation numerically using my simulation model. Focusing on just the final portion of our production system the situation now is presented in Figure 17, where Q10 feeds both machines (designated as 8a and 8b).

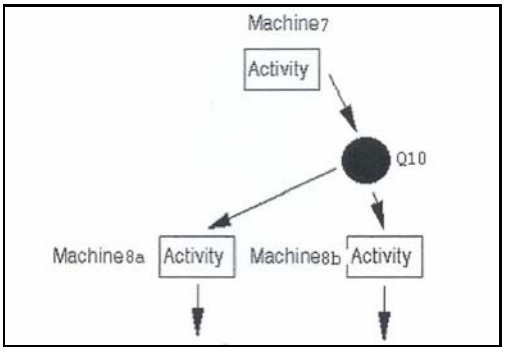


Figure 17:

- The WINQSB input is as in Figurel8.

Component Name	Type (C/S/Q/G)	Immediate Follower (Name : Prob / TransferTime, separated by ',')	Input Rule	Output Rule	Queue Discipline	Queue Capacity	Attribute Value	Interarrival Time Distribution
x	C	q1						UNIFORM/0.2/1.0
y	C	q2						UNIFORM/0.6/0.6
z	C	q6						NORMAL/0.6/0.2
i	C	q7						NORMAL/0.6/0.01
machine1	S	q3						
machine2	S	q4						
machine3	S	q5	Assembly					
machine4	S	q11						
machine5	S	q8	Assembly					
machine6	S	q9						
machine7	S	q10	Assembly					
machine8a	S							
machine8b	S							
q1	Q	machine1			FIFO	30		
q2	Q	machine2			FIFO	30		
q3	Q	machine3			FIFO	50		
q4	Q	machine3			FIFO	50		
q5	Q	machine5			FIFO	30		
q6	Q	machine4			FIFO	10		
q7	Q	machine6			FIFO	10		
q8	Q	machine7			FIFO	100		
q9	Q	machine7			FIFO	100		
q10	Q	machine8a,machine8b			FIFO	150		
q11	Q	machine5			FIFO	200		

Figure 18:

The graphical model:

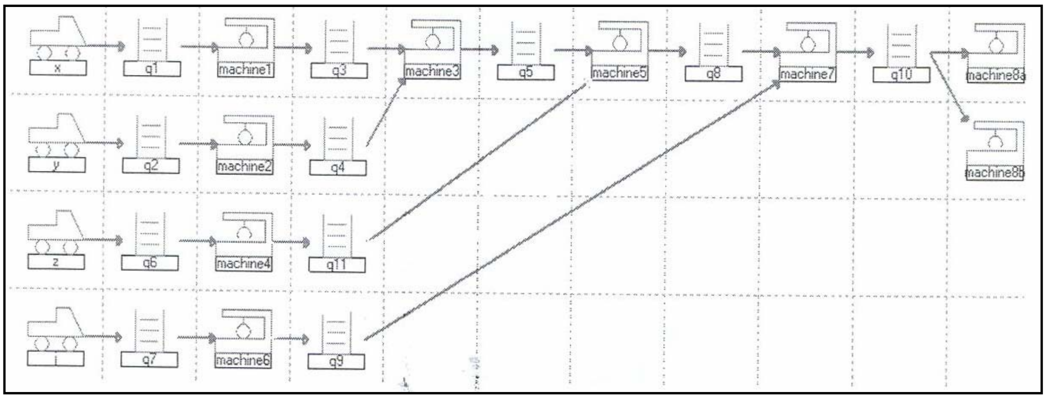


Figure 19:

- The results, after simulating for the same time period as before, are presented in: Figure20, Figure21 (for utilisation of the servers) and in Figure22 (for analysis of the queues).

01-30-2003	Result	x	y	z	i	Overall
1	Total Number of Arrival	170	166	170	167	673
2	Total Number of Balking	0	0	0	0	0
3	Average Number in the System (L)	7.0251	1.1451	4.3844	1.8369	14.3915
4	Maximum Number in the System	11	2	9	3	25
5	Current Number in the System	7	1	5	3	16
6	Number Finished	163	0	0	0	163
7	Average Process Time	2.9457	0	0	0	2.9457
8	Std. Dev. of Process Time	0.3519	0	0	0	0.3519
9	Average Waiting Time (Wq)	5.6668	0	0	0	5.6668
10	Std. Dev. of Waiting Time	1.5539	0	0	0	1.5539
11	Average Transfer Time	0	0	0	0	0
12	Std. Dev. of Transfer Time	0	0	0	0	0
13	Average Flow Time (W)	4.6643	0	0	0	4.6643
14	Std. Dev. of Flow Time	0.9438	0	0	0	0.9438
15	Maximum Flow Time	6.2829	0	0	0	6.2829
	Data Collection: 0 to	100 hours				
	CPU Seconds =	2.9370				

Figure 20:

01-30-2003	Server Name	Server Utilization	Average Process Time	Std. Dev. Process Time	Maximum Process Time	Blocked Percentage	# Customers Processed
1	machine1	17.26%	0.1015	0.0101	0.1320	0.00%	170
2	machine2	99.10%	0.6006	0.0220	0.6638	0.00%	165
3	machine3	33.00%	0.2000	0.0002	0.2000	0.00%	165
4	machine4	76.37%	0.4519	0.0199	0.5136	0.00%	169
5	machine5	32.80%	0.2000	0.0002	0.2000	0.00%	164
6	machine6	66.37%	0.3998	0.0204	0.4534	0.00%	166
7	machine7	32.80%	0.2000	0.0002	0.2000	0.00%	164
8	machine8a	65.94%	0.7758	0.3308	1.5449	0.00%	85
9	machine8b	63.18%	0.8100	0.3693	1.8034	0.00%	78
	Overall	54.09%	0.3671	0.2544	1.8034	0.00%	1326
Data	Collection:	0 to	100	hours	CPU	Seconds =	2.9370

Figure 21:

01-30-2003	Queue Name	Average Q. Length (Lq)	Current Q. Length	Maximum Q. Length	Average Waiting (Wq)	Std. Dev. of Wq	Maximum of Wq
1	q1	0	0	1	0	0	0
2	q2	0.1539	0	1	0.0927	0.0694	0.2337
3	q3	4.5477	5	8	2.7174	1.0409	4.7122
4	q4	0.0002	0	1	0.0001	0.0016	0.0200
5	q5	0	0	1	0	0	0
6	q6	0.1194	0	2	0.0702	0.1305	0.7671
7	q7	0	0	1	0	0	0
8	q8	0	0	1	0	0	0
9	q9	1.1688	2	2	0.7092	0.0688	0.8693
10	q10	0.0155	0	1	0.0095	0.0387	0.2980
11	q11	3.4995	4	7	2.0814	1.0481	4.1451
	Overall	9.5050	11	8	0.5134	1.0208	4.7122
Data	Collection:	0 to	100	hours	CPU	Seconds =	2.9370

Figure 22:

- It can be seen that my expectation as to doubled production is way off the mark ¬according to my results, I have increased production from the 127 items (approximately 127/100 = 1.27 items per hour) we had before to 163 items now (approximately 163/100 = 1.63 items per hour). This is far from a doubling of production.
- Well perhaps the problem is that I have not simulated for long enough. Let see what happens if I simulate for ten times as long (say for 1000 hours, where data collection starts after 200 hours).

2.2.3. Simulating for 1000 hours

- The output is shown in Figure 23- production in (approximately) 800 hours is 1335 items, i.e. 1.7 items per hour.
- It is clear from simulating for longer that the effect we are seeing, having two Machine 8's does not double production is real. But why?

The simple answer is that this situation is common in simulation - linked systems of queues and activities, such as we have here, are notoriously difficult to analyses in terms of common sense. Instead a detailed examination of simulation output is necessary to see what is occurring.

01-30-2003	Result	x	y	z	i	Overall
1	Total Number of Arrival	1338	1333	1337	1333	5341
2	Total Number of Balking	0	0	0	0	0
3	Average Number in the System (L)	4.1263	3.0583	8.7190	3.4961	19.3997
4	Maximum Number in the System	11	12	17	12	52
5	Current Number in the System	6	2	6	3	17
6	Number Finished	1335	0	0	0	1335
7	Average Process Time	2.9552	0	0	0	2.9552
8	Std. Dev. of Process Time	0.3386	0	0	0	0.3386
9	Average Waiting Time (Wq)	8.6592	0	0	0	8.6592
10	Std. Dev. of Waiting Time	3.1590	0	0	0	3.1590
11	Average Transfer Time	0	0	0	0	0
12	Std. Dev. of Transfer Time	0	0	0	0	0
13	Average Flow Time (W)	6.8801	0	0	0	6.8801
14	Std. Dev. of Flow Time	2.5400	0	0	0	2.5400
15	Maximum Flow Time	11.7315	0	0	0	11.7315
	Data Collection: 200 to	1000 hours				
	CPU Seconds =	27.5160				

Figure 23:

- For our particular example the answer as to why we are not producing double the amount lies in a consideration of the queue statistics.
- Summarise all the records obtained:

Table 1: Final results

	Initial system	Changing the queue discipline for Q10	Getting two Machines8	Simulating for 1000 hours	Conclusions
Total number of final items	127	127	163	1335	
Average number of final items per hour	127/100=1.27 items per hour	127/100=1.27 items per hour	163/100=1.63 items per hour	1335/800=1.668 items per hour	No doubled production
Average process time	2.7153 hours	2.7163 hours	2.9457 hours	2.9552 hours	
Average flow time	12.6030 hours	12.8057 hours	4.6643 hours	6.8801 hours	
Machine1 utilisation	16.24 %	16.24 %	17.26 %	16.70 %	Machine1 is not fully utilised
Machine2 utilisation	99.28 %	99.28 %	99.10 %	100.01 %	Machine2 is fully utilised
Machine3 utilisation	32.40 %	32.40 %	33 %	33.38 %	Machine3 is not fully utilised
Machine4 utilisation	73.64 %	73.64 %	76.37%	75.25 %	Machine4 is not fully utilized
Machine5 utilisation	32.40 %	32.40 %	32.80%	33.35 %	Machine5 is not fully utilised
Machine6 utilisation	32.89 %	32.89 %	66.37 %	66.68 %	Machine6 is not fully utilised
Machine7 utilisation	32.40 %	32.40 %	32.80 %	33.38 %	Machine7 is not fully utilised
Machine8 utilisation	97.77 %	97.77 %			Machine8 was a bottleneck
Machine8a utilisation			65.94 %	67.15 %	Machine8a is not fully utilised
Machine8b utilisation			63.18 %	67.19 %	Machine8b is not fully utilised
Average Q1 length	0	0	0	0	There exist no X parts in system
Average Q2 length	0.4654	0.4654	0.1539	0.9362	There exist Y parts in system
Average Q3 length	0.2665	0.2665	4.5477	1.5491	There exist X parts in system
Average Q4 length	1.5720	1.5720	0.0002	1.1223	There exist Y parts in system
Average Q5 length	0.0931	0.0931	0	0.0127	There exist X + Y parts in system
Average Q6 length	0.0753	0.0753	0.1194	0.0858	There exist Z parts in system
Average Q7 length	0	0	0	0	There exist no J parts in system
Average Q8 length	0	0	0	0	There exist no X + Y + Z parts in system
Average Q9 length	3.4215	3.4215	1.1688	2.8294	There exist J parts in system
Average Q10 length	15.3615	15.3615	0.0155	0.0532	There exist X+Y+Z+J parts in system
Average Q11 Length	1.5648	1.5648	3.4995	7.8800	There exist J parts in system

3.CONCLUSION

After deep analyses of problem in my producing system I realized that by adding new machine I don't get a doubled production.
At first analyses I noticed that machine2 is full utilized like machine8, but after graphic analyses I noticed that average Q length has the biggest value for Q 10 and because that I suspected that machine8 is bottleneck.
After I added new machine (machine8b) in to the my system I noticed that that average Q length on the Q 10 has smaller value, but in the same moment average Q length on the Q11 and Q3 was increased as and Machine 6 utilization.
Interesting moment in my simulation is that Machine 2 utilization after I added new machine in the system was little decreased.
After I made decision to increase time, some of the variables showed to me possible reason because my production wasn't doubled.
At first I had to notice that Machine 2 utilization was increased on 100.01% which means that adding new machine didn't resolve my problem because I got new bottleneck.
And average Q length on the q 11 was increased.
My analyses shows me full complexity of my system and in order to improve performance off that system I have. to do much more analysis.

REFERENCES

[1] Y.L.Chang, *WinQSB, Decision Suport Software for MS/OM*, John Wiley&Sons, Inc. New York, 1998
[2] Winston, L. W. *Operations Research Applications and Algorithms*.,Wadsworth Publishing Company, 1994.
[3] www.microsoft.com/msproject
[4] http://home.ubalt.edu/ntsbarsh/Business-stat/opre/partX.htm
[5] Thomas G. G. Robertazzi, Thomas G. Robertazzi, *Computer Networks and System: Queueing Theory and Performance Evaluation,* Springer-Verlag New York, LLC,2004
[6] George Liberopoulos, *Stochastic Modeling of Manufacturing Systems: Advances in Design,Performance Evaluation and Control Issues,* Springer-Verlag New York,2006