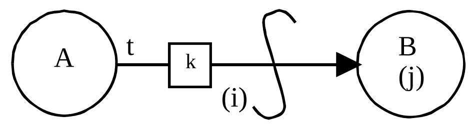
C. More Resident Entity Queuing Models

An efficient means of tracking transient entities is to attach their relevant attributes to events on the future events list. For example, the time a part first entered a manufacturing activity can be recorded on the future events list along with the other information for a machine start event that begins work on that part.

In this section you will learn:

1. To pass on additional information about a future event at the time it is scheduled on the future events list.
2. How to model the impact of limited resources.

An enhanced form of an event graph is shown to the right. This graph is read as "if (i) is true when event A occurs it will schedule B to occur t time units later with parameter j equal to the current value of k. (Note that values are being passed, not variable names or formulas.)



C1. Exercise -- Event Graphs with Parameter Passing

Reproduce the event graph in Figure 5.14 on a sheet of paper.

1. Use a sequence of future events lists based on this event graph to understand the logic of the model. Use initial values of S = 2 and Q=0, t_a = 7 and t_s = 13. Check the future event list calculations frequently by simultaneously doing Exercise C2. Generate the future events lists through the third customer to leave.

2. This exercise illustrates the passing of part arrival times to let you determine wait times and the use of a single data storage array for storing two different quantities. What are those two quantities?

3. Augment the event graph computations on the event graph so that running totals of the total time and average time customers spend in the system are maintained. Compare your changes to those in C01106A.MOD.

4. The size of the array limits the total number of customers that can be run in the simulation. Can you think of a way to get around this limitation without unduly increasing the size of the array?

C2. Computer Exercise -- Part Wait Times

Run BANK2.MOD in single step mode using the values from exercise C1. Compare the future events lists with those that you generated in exercise C1. (Do not overwrite the original BANK2.MOD.) Note that BANK2.MOD allows you to begin with parts in the queue and to start processing them first.

C3. Computer Exercise -- Part Wait Times

Run BANK2.MOD and refer to figure 5.14. Set the mean interarrival time to 7 and mean service time to 13. (IAT and MST in the Run pull down menu.) Set S to 2, Q to 0 and the random number seed to 12345. Single-step through the fourth customer to leave and observe all the future events lists, including the traced variables. Note that the third customer (identified by index IN) leaves before the second customer. What is the time in process for each of the customers?

C4. Computer Exercise -- Part Wait Times

1. Refer to the event graph of figure 5.1. Arriving parts form a single queue. The parts are to be processed on a single machine. Let T_S be the average processing time on a machine and $Q(t_i)$ be the number of parts in the queue the instant before the i + 1st change in queue length. Let N be the number of parts that have completed service. Run BANK1.MOD and look at the step plot of QUEUE versus time. Identify $Q(t_i)$, $Q(t_{i+1})$, $Q(t_{i+2})$, etc.

2. Suppose you run the simulation for a large number of parts, say N=10,000. Give a formula in terms of $Q(t_i)$, T_S, and t_i for estimating the wait time for a part and **clearly** define each of the terms. You **do not** have to alter the event graph to answer the question. Use it to remind you of what events happen and when state variables change.

3. How would you incorporate the calculations in part 2 into BANK1.MOD? Do this on a piece of paper and then compare your changes to C01106A.MOD.

C5. Computer Exercise -- Part Wait Times

There are two measures that people typically look at that depict waiting: 1) the average time spent waiting in line and 2) the average time spent in the system.

1. How are the two wait times related? Give a formula. Refer to the discussion in Section 5.7.1 of the text. If you were running a business where arriving customers had to wait, why might the average time spent waiting in line be a more important measure to focus on than the average time spent in the system?

2. I have modified BANK1 and BANK2 simulations as C01106A.MOD and BK219116. Classify the type model in each case. Look at these models and tell me how they calculate average wait time. Give a formula for each case.

3. Using three servers, an arrival delay of 4*ERL{1} minutes and a service time of 10*ERL{1}, run both models for 200 customers (make sure the initial queue is zero). For C01106A.MOD and BK219116.MOD, make step plots of the average time spent waiting in line, average time spent in the system and average service time. Compare the two results.

4. Based on your runs for 200 customers, would you be willing to give an estimate for the true average service time (the average time over a large number of customers, say 100,000)? How about the average time spent waiting in line or average time spent in the system? Why or why not? Use your plots from part 3 to form your arguments. If you do not have a good estimate of

any of the quantities, make a run that will give you a good estimate. Which type of model, C01106A.MOD or BK219116, did you use? Why?

C6. Computer Exercise -- Part Wait Times and Replicating Simulation Runs

Refer to exercise C5. Set up BK219116.MOD as a single server model with customer arrival delays 5*ERL{1} and customer service delays of 4.5*ERL{1}. Change the dimensioning of W[ ] as required.

1. Try 10 different starting random number seeds and report the results.

2. Rerun the BK219116.MOD with two servers each having customer service delays of 7*ERL{1}. Try 10 different starting random number seeds and report the results. Compare with the results of part 1 and comment on the differences.

3. Further alter BK219116.MOD so that it repeats an eight hour day 200 times. At the end of each daily replication it calculates the average daily wait times. Compare your program with BANK2REP.MOD. Compare the results for the one and two server case and comment on the difference. Plot histograms of the average daily wait times.

C7. Computer Exercise – Tracking Simulation Performance

Access STAT2.MOD.

1. This program tracks various simulation performance measures. The variables, P, MAXTS, MAXW, AVGTS, AVGW, MAXQ, TAVGQ, and UTIL are traced. These variables will be updated as the SIGMA program runs. Run the program and observe each of these variables. Use the "write in English" translation to get the better idea of what measure of performance each variable tracks.

2. Explain how the calculation for TAVGQ and UTIL work.

3. Change the interarrival distribution to a exponential distribution with mean = 5 minutes, and change the services time distribution to a triangular distribution with minimum 1 minute, mode 3 minutes, and maximum 6 minutes. Run the simulation for 20 minutes and 1000 minutes. What are the values of the performance statistics?

4. Compare the values to the values on the table 2-3 in the Arena book. Explain why the performance values from the program are different from the hand simulation.

C8. Computer Exercise -- Resources

Access TWOQUEUE.MOD.

1. What does this program do? The language in the descriptions seems to not make a particularly easily read "write in English" translation. There are also some inconsistencies in the descriptions versus what is actually implemented in the vertices and edges. Rework the descriptions to produce a good understandable "write in English" translation.

2. What is going on with a vertex without edges going into or out of it? How does it get executed? What is its purpose?

3. Add a vertex that switches workers in the opposite direction from CHANGE. Print out the resulting event graph and show the state changes for the new node only.

4. You are going to experiment with the apportionment of servers between the two operations, starting with the initial apportionment of 2 and 6. (Assume that the skill level required is equal for both operations.) Make sure that the initial queues are zero, that $t_a(1)$=5*RND, $t_s(1)$=5 and that $t_s(2)$=4*ERL{1}. What additional variables do you need to graphically evaluate the different apportionment as the simulation is running?

5. Can you, if you are quick, grab all of the servers away from operation 2? Try to do it. You have to make sure that there is, in fact, a free server before making the switch. Most were able to get within one. A few got them all. You can slow the simulation down to give you a better chance.

6. Experiment with the simulation to find an optimal apportionment of servers between the two operations. Justify your result.

7. Give an example of a real situation where this model would apply.

C9. Exercise -- Blocking

Consider the event graph in Figure 5.18. Let S[1] = S[2] = 1, B[1] = B[2] = 2, t_a = 5, $t_{S[1]}$ = 4 and $t_{S[2]}$ = 12.5.

1. Provide future events lists through the first instance of a job being turned away from the queue in front of the first machine.

2. Access TWOQUES.MOD and check your results in 1. by running in single step mode with the given settings. Correcting your future events lists as needed.

3. Run TWOQUES.MOD with the settings (S[1]=3, S[2]=2, B[1]= 5 and B[2]=2, t_a=5*ERL{1}, t_{S1}=4+16*RND and t_{S2}=10+30*RND) and comment on the differences.

4. Using the settings in part 3 and a random number seed of 12345, run TWOQUES.MOD for 10,000 minutes. Keep track of number of customers turned away and the number of times blocking occurs.

C10. Computer Exercise -- Blocking and Parameter Passing

Consider the SIGMA program TWOQUES1.MOD.

1. Draw a properly and fully annotated event graph of TWOQUES1.MOD. Describe what the model does. What are the differences from the model of exercise C9?

2. Make any necessary modifications to TWOQUES1.MOD and run it with the same settings as in exercise C9 part 3. Do you get the same results? Why or why not?

C11. Exercise -- Project Management

The Schruben SIGMA text develops PERT (Program Evaluation and Review Technique) activity precedence networks using an activity-on-node structure. This approach is rarely seen these days. Instead, the activity-on-arc structure has been the approach taught for more than 30 years. The PERT chart from either approach results in an event graph, but the activity-on-arc PERT chart is easier to construct and the accompanying event graph logic is somewhat easier to follow. This exercise and exercise C12 leads you through the activity-on-arc approach for a simple problem.

From the table below, construct PERT network event graph. Label the edges with the appropriate activity and number the vertices. Using a triangular distribution TRI{M}, where M is the mode normalized to the [0, 1] interval, provide the probability distribution for the delays for each of the activities. Set up a variable in each of the vertices to count the completion of activities leading to that vertex. Annotate each of the edges leading from a vertex with a logic statement testing for the required count.

Activity	**Must Follow**	**Optimistic**	**Most Likely**	**Pessimistic**	**Distribution**
A	None	20	40	100	20+(100-20)*TRI{0.25}
B	None	20	30	40	
C	B	10	20	60	
D	A and C	15	25	35	
E	B	25	35	60	

C12. Computer Exercise -- Project Management

Use PERTCHRT.MOD as a model to program your event graph from exercise C11 in SIGMA. Replicate the project 10,000 times. Use a random number seed of 12345. Plot a histogram of the project completion times. What does this histogram represent? Estimate the probability that the project will take more than 80 days?