King Khalid University

College of Computer Science

Department of Computer Science

Subject Code: 233CSM

Subject Name: Computer Modeling and Simulation

Mini-Project-1

Instructions:

1. Store all the Programs/Results/Answers/Outputs and upload on the Blackboard System.

Student ID: 439805648 **Student Name:** Hatem Amer Essa Asiri **Section. No:** 1306

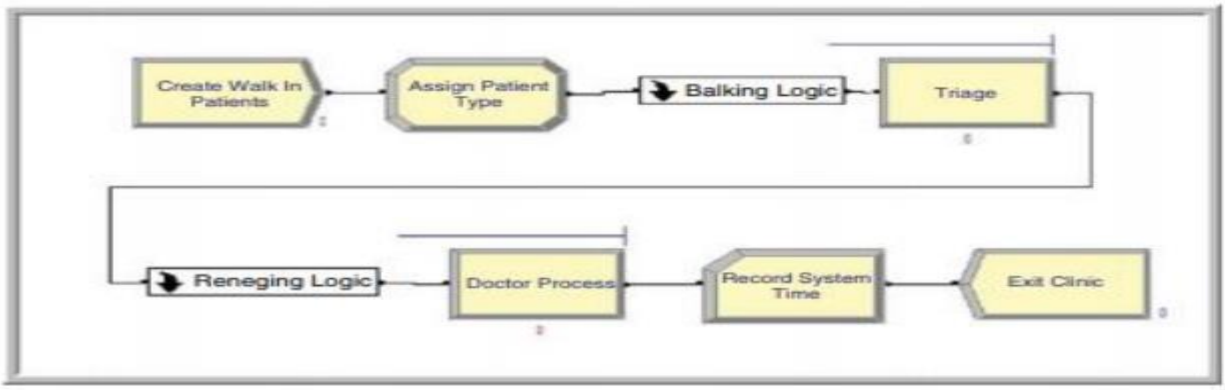
1. Develop an Arena model for a Health-Care Clinic as per the description given below:

A walk-in health care clinic has analysed their operations and they have found that they can classify their walk-in patients into three categories: high priority (urgent need of medical attention), medium priority (need standard medical attention), and low priority (non-urgent need of medical attention). On a typical day during the period of interest, there are about 15 arrivals per hour with 25% being high priority, 60% being medium priority, and the remaining being low priority. The clinic is interested in understanding the waiting time for patients at the clinic. Upon arrival to the clinic, the patients are triaged by a nurse into one of the three types of patients. This takes only 2–3 minutes uniformly distributed. Then, the patients wait in the waiting room based on their priority. Patients with higher priority are placed at the front of the line. Patients with the same priority are ordered based on a FCFS basis. The service time distributions of the customers are given as follows.

The clinic has four doctors on staff to attend to the patients during the period of interest. They have found through a survey that if there are more than 10 people waiting for service, an arriving low priority patient will exit before being triaged. Finally, they have found that the nonurgent (low priority) patients may depart if they have to wait longer than 15 ± 5 minutes after triage. That is, a non-urgent patient may enter the clinic and begin waiting for a doctor, but if they have to wait more than 15 ± 5 minutes (uniformly distributed), they will decide to renege and leave the clinic without getting service.

Priority	Service Time Distribution (in Minutes)
High	Lognormal(38, 8)
Medium	Triangular(16, 22, 28)
Low	Lognormal(12, 2)

I. With the help of the picture, develop an Arena model for the above system.



II. Estimate the following statistics:

- The average system time of each type of patient.
- The probability that low priority patient's balk.
- The distribution of the number of customers waiting in the doctor queue.

Figure 6.38: Overview of walk-in clinic model

Assignments

	Type	Attribute Name	Other	New Value
1	Attribute	myArrivalTime		TNOW
2	Attribute	myType		PatientTypeDist
3	Other	Attribute 3	Entity.Type	MEMBER(PatientTypeSet,myType)
4	Other	Attribute 4	Entity.Picture	MEMBER(EntityPictureSet,myType)

Double-click here to add a new row.

Figure 6.39: Assigning the type of patient

The Triage and Doctor PROCESS modules are done similarly to how you have implemented many of the prior models. In order to implement the priority for the patients by type, the QUEUE module can be used. In this case the `myType` attribute can be used with the lowest attribute value as shown in Figure 6.40. You should attempt to build this model or examine the supplied file for all the details.

Queue - Basic Process

	Name	Type	Attribute Name	Shared	Report Statistics
1	Triage.Queue	First In First Out	Attribute 1	☐	☑
2	DoctorProcess.Queue	Lowest Attribute Value	myType	☐	☑

Double-click here to add a new row.

The balking and reneging logic have been placed inside two different sub-models. Balking only occurs for non-urgent patients, so the type of patient is first checked. If the patient is of type non-urgent, whether a balk will occur can be recorded with the expression, `NQ(DoctorProcess.Queue) >= vBalkCriteria`. This expression evaluates to 1 for true or 0 for false. Thus, the probability of balking can be estimated. The patients who actually balk are then disposed. This is illustrated in Figure 6.41.

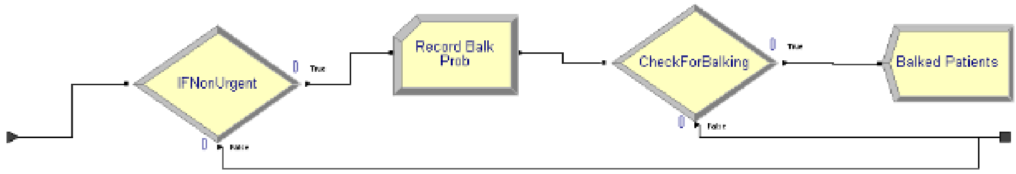


Figure 6.41: Balking logic sub-model

The reneging logic is more complicated. Figure 6.42 presents an overview of the sub-model logic for reneging.

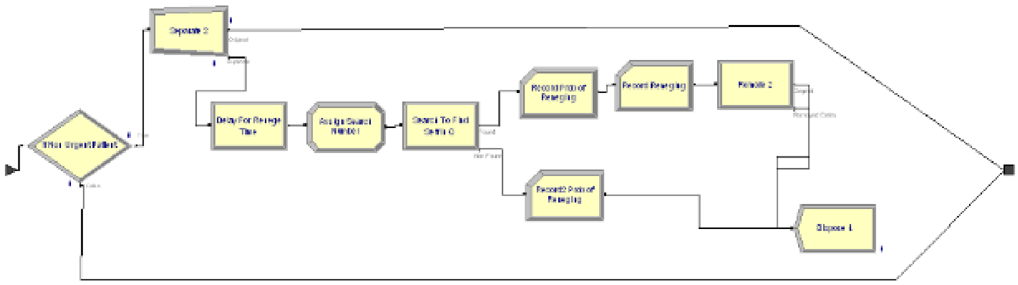


Figure 6.42: Reneging logic sub-model

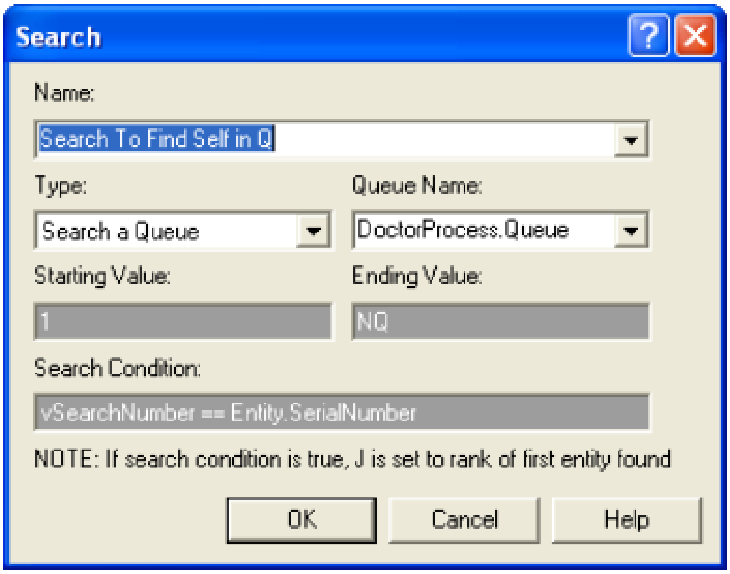


Figure 6.43: SEARCH module for reneging logic

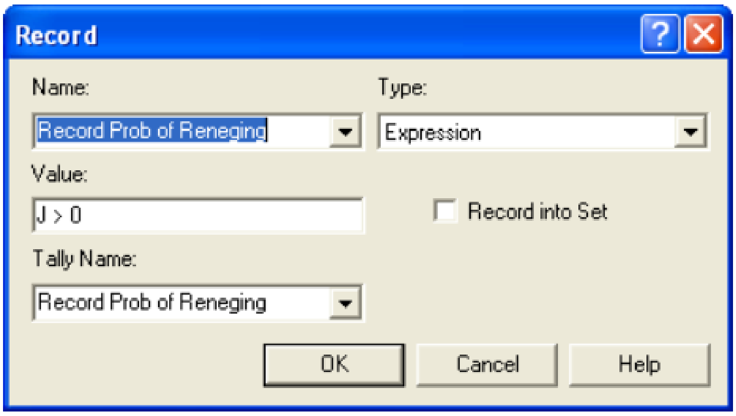


Figure 6.44: Recording whether reneging occurred

If $J > 0$, then the entity at rank J can be removed from the `DoctorProcess.Queue` as illustrated in Figure 6.45. The rest of the model is relatively straightforward and you are encouraged to explore the final dialog boxes.

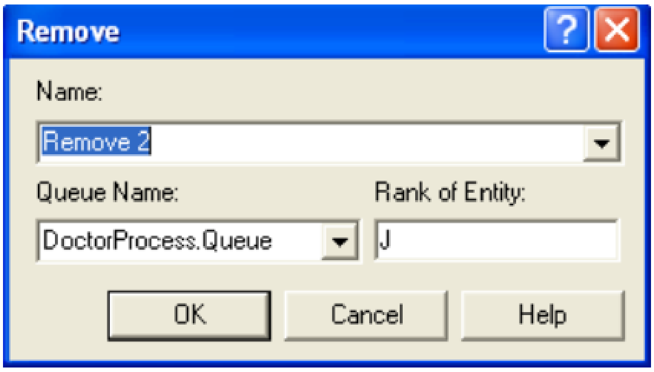
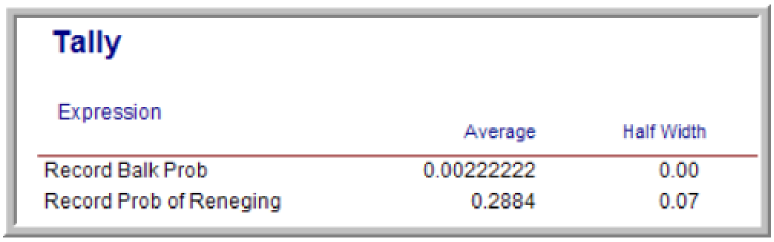


Figure 6.45: REMOVE module for removing reneging patient

Assuming that the clinic opens at 8 am and closes at 6 pm, the simulation was set up for 30 replications to yield the results shown in Figure 6.46. It appears that there is a relatively high chance (about 29%) that a non-urgent patient will renege. This may or may not be acceptable in light of the other performance measures for the system. The reader is asked to further explore this model in the exercises, including the implementation to collect statistics on the number of customers waiting in the doctor queue.



Tally

Expression	Average	Half Width
Record Balk Prob	0.00222222	0.00
Record Prob of Reneging	0.2884	0.07

Figure 6.46: Example output for walk-in clinic model

While some analytical work has been done for queuing systems involving balking and reneging, simulation allows for the modeling of more realistic types of queueing situations as well as even more complicated systems. In the next section, we introduce a very useful construct that enables condition based signaling and control of entity movement. When the entity waits for the condition, it waits in a queue. This permits a wider variety of modeling involving queues.