

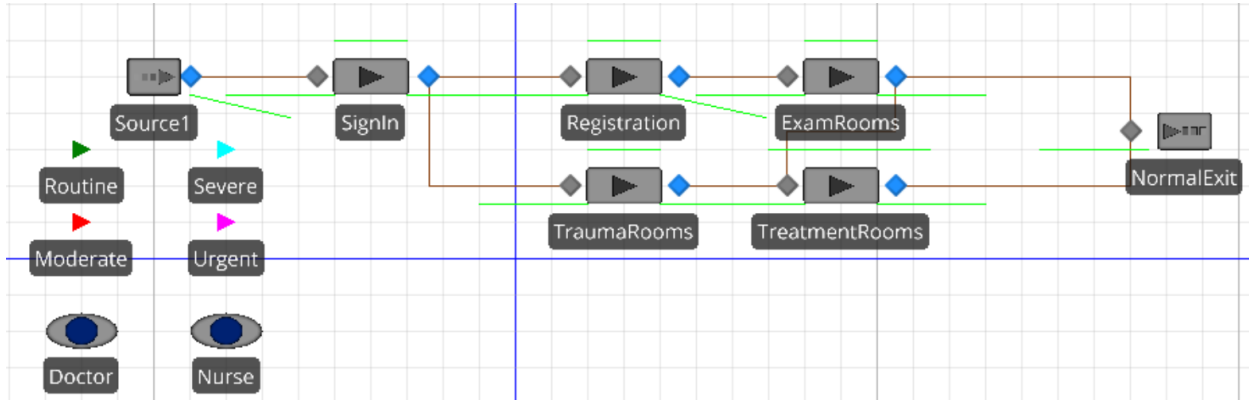
DATA 604 Assignment 11: Advanced Modeling

Dan Smilowitz

April 23, 2017

Optimization of Current Model

The facility for Model 9-1 as illustrated in the text is shown below:



Running this model for 10 days with the default values, the average wait time for urgent patients is just short of 2 hours:

UrgentWaiting	Average (Hours)	1.9930
	Maximum (Hours)	3.7247
	Minimum (Hours)	0.2259
	Observations	28.0000

Using the *OptQuest* and *Select Best Scenario Using KN* plugins, along with the *Subset Selection* tool, the optimal Total Cost (TC) with a minimum **Satisfaction** of patients of 0.8 occurs with the following setup:

- Staffing:
 - 1 Doctor
 - 1 Nurse
- Capacity:
 - 1 Exam Room
 - 3 Treatment Rooms
 - 1 Trauma Room

Scenario		Replications		Controls							Responses	
<input type="checkbox"/> Name	Status	Required	Completed	DoctorCapacity	ExamCapacity	TreatmentCapacity	TraumaCapacity	NurseCapacity	SatisfactionThreshold	TC	▲	Satisfaction
<input type="checkbox"/> 062	Idle	7	7 of 7	1	1	2	1	1	1	2.89523		0.714286
<input checked="" type="checkbox"/> 048	Idle	6	6 of 6	1	1	3	1	1	1	3.16409		0.833333
<input checked="" type="checkbox"/> 061	Idle	6	6 of 6	1	1	3	1	1	2	3.16409		0.833333
<input checked="" type="checkbox"/> 084	Idle	6	6 of 6	1	1	3	1	1	3	3.16409		0.833333
<input checked="" type="checkbox"/> 088	Idle	6	6 of 6	1	1	3	1	1	5	3.16409		0.833333
<input checked="" type="checkbox"/> 098	Idle	6	6 of 6	1	1	3	1	1	4	3.16409		0.833333
<input checked="" type="checkbox"/> 059	Idle	6	6 of 6	1	1	2	2	1	1	3.30559		0.833333
<input checked="" type="checkbox"/> 065	Idle	6	6 of 6	1	1	4	1	1	2	3.41409		0.833333
<input checked="" type="checkbox"/> 052	Idle	5	5 of 5	1	2	2	2	1	2	3.46668		0.8
<input checked="" type="checkbox"/> 054	Idle	6	6 of 6	1	1	3	2	1	5	3.56409		0.833333
<input checked="" type="checkbox"/> 063	Idle	6	6 of 6	1	1	3	2	1	3	3.56409		0.833333
<input checked="" type="checkbox"/> 083	Idle	6	6 of 6	1	1	3	2	1	2	3.56409		0.833333

Preemption for Urgent Patients

In order to adapt the system to allow for urgent patients to preempt a doctor assigned to another non-urgent patient, the following high-level logic must be implemented via processes:

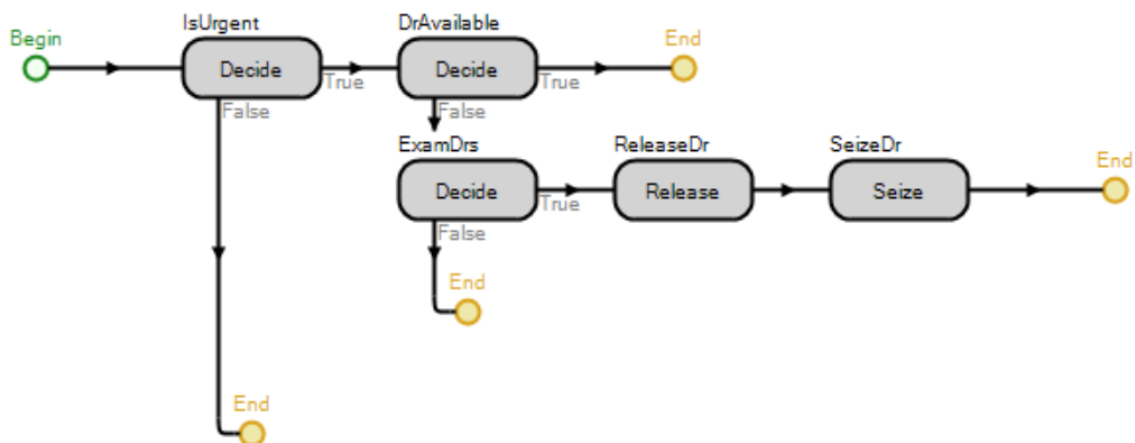
- When urgent patient arrives, check if doctor available
 - If so, proceed normally
- If doctor is unavailable, check if currently occupied with urgent patient
 - If so, proceed normally
- If not, interrupt processing and release doctor
- Seize doctor and process patient
- After urgent patient is processed, release doctor and return to previous patient

Implementation of this logic was attempted through a number of add-on processes.

Processes at Servers

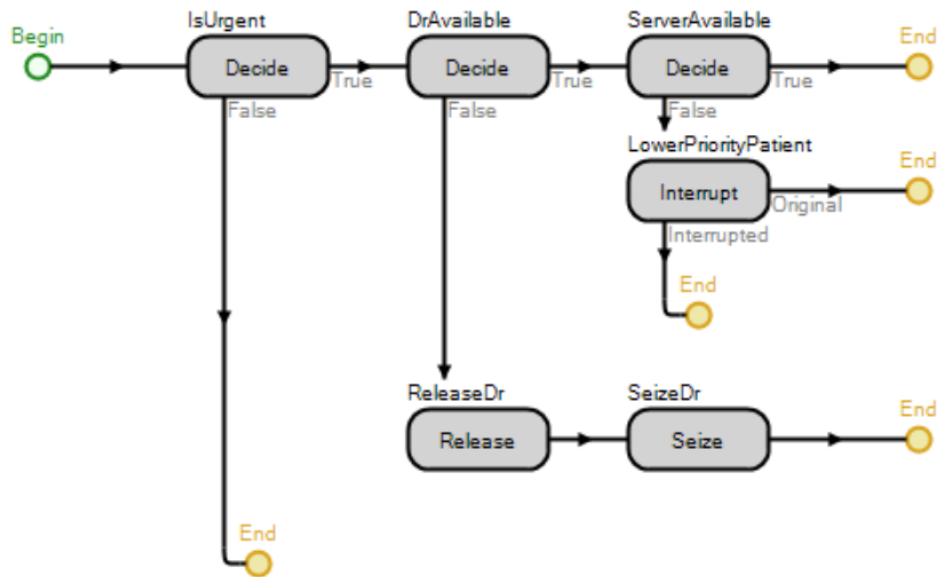
First, an add-on process was added at the Trauma and Treatment servers, both set to be triggered when a resource enters the respective servers:

Trauma_Preempt_Doctor



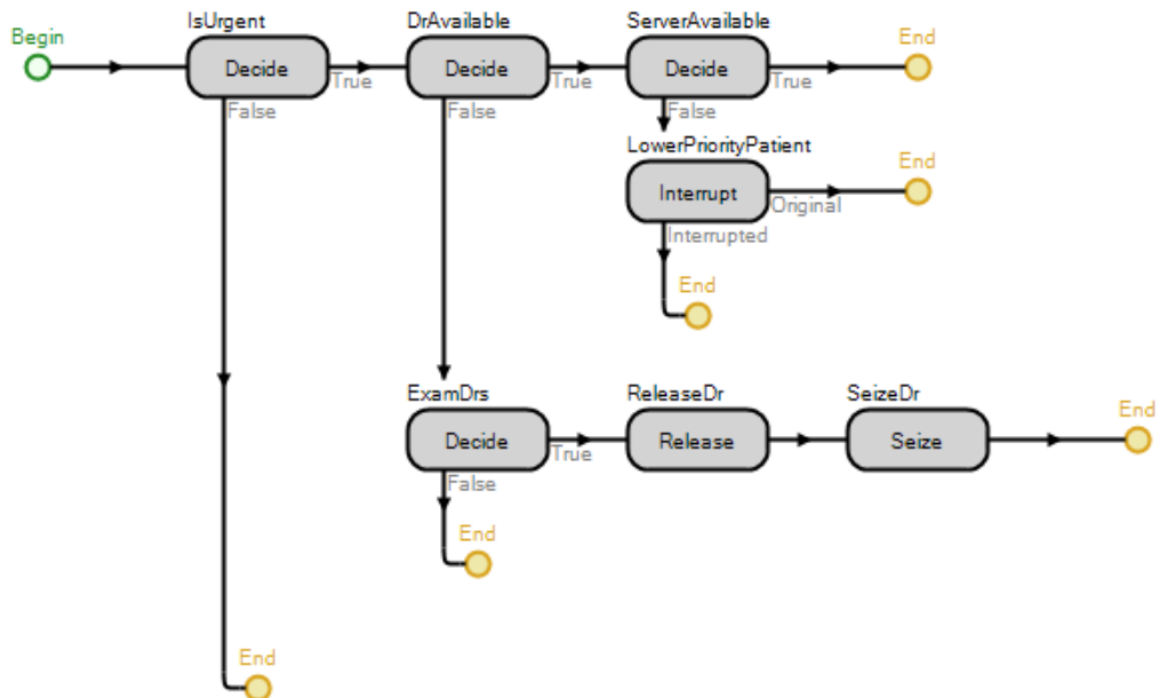
This process caused an error in the release step, so was modified:

Treatment_Preempt_Doctor



This process did not create an error, but yielded the same wait times as above for urgent patients; thus it was easily concluded that it did not have the desired effect. As such, the process was further tweaked to see if the doctor resource was owned at the Exam Room server prior to releasing & seizing:

Treatment_Preempt_Doctor_1

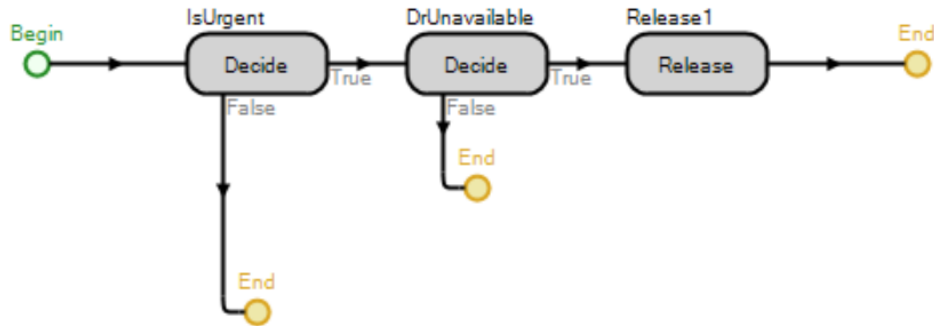


Unfortunately, this process yielded the same lack of impact. As such, processes related to the doctor resource itself were considered.

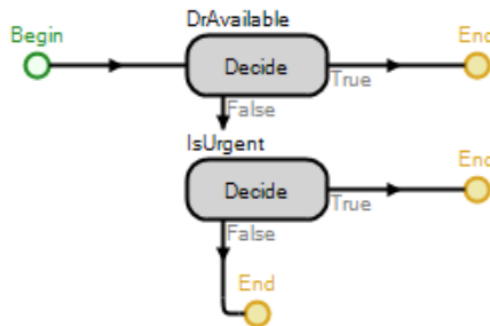
Processes at Resource

The Doctor resource has different add-on process triggers than the servers; these include *Allocated*, *Released*, and *Evaluating Seize Request*. Given the shortcomings of the server-based add-on processes, the preemption was attempted via use of these triggers:

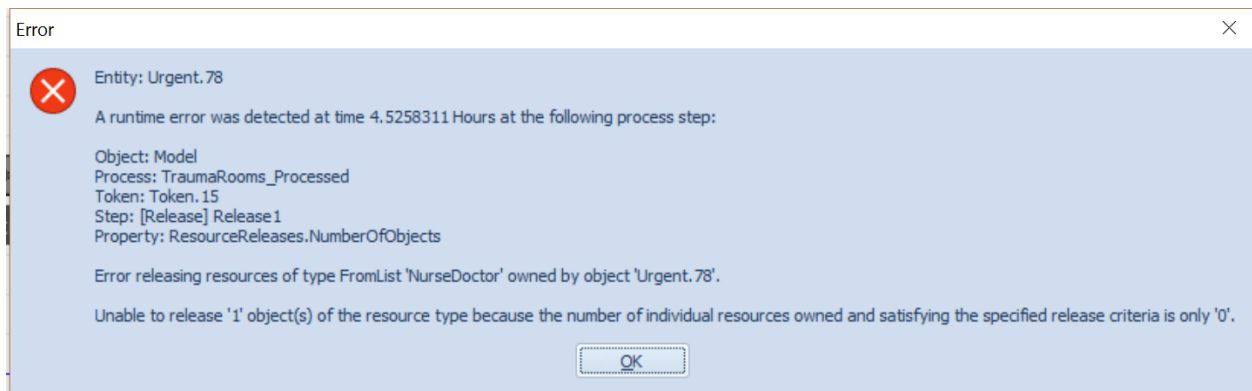
Doctor_Allocated



Doctor_EvaluatingSeizeRequest



These processes seek to carry out preemption in the following way: - If the doctor is available, proceed as normal - If the doctor is unavailable and the patient is urgent, accept the request - When allocating, release the doctor from the current patient, unless the current patient is also urgent. This also yielded an error:



Curiously, the error message indicates that the error in the release step was triggered by the *TraumaRooms_Processed* process, which was in place and working successfully in the as-is model.

Unfortunately, trial and error with a combination of process steps, logic, and routing were unsuccessful in creating a system where urgent patients are able to preempt doctors from non-urgent patients.