

Problem 5.1:

Basically,

Basically, properties stay constant during a simulation; whereas, states can change.

	Properties	States
Basic data types	17	8
Runtime change	No	Yes
Where stored	object instance	object runspace
Server example	processing time	number processed
Entity example	initial speed current speed	
Cost example	cost per hour	accrued cost
Failure example	failure rate	last failure time
Batching example	desired batch size	current batch size

Problem 5.2:

The *parent object* is an instance of the object in which the process is defined. The *associated object* is the related object that triggered this process to execute.

Problem 5.3:

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average	Minimum	Maximum	Half Width
ModelEntity	PCB	[Population]	Content	NumberInSystem	Average	2.9923	2.7890	3.2786	0.0596
			FlowTime	TimeInSystem	Average (Ho...	0.2997	0.2802	0.3248	0.0052
Server	Inspection	[Resource]	Capacity	UnitsUtilized	Average	0.4979	0.4845	0.5132	0.0031
		InputBuffer	Throughput	NumberExited	Total	9,981.7200	759.0000	162.0000	44.5533
		OutputBuffer	Throughput	NumberExited	Total	9,981.7600	758.0000	161.0000	44.5525
		Processing	Throughput	NumberExited	Total	9,981.7600	758.0000	161.0000	44.5525
	Placement	[Resource]	Capacity	UnitsUtilized	Average	0.6662	0.6494	0.6841	0.0041
		InputBuffer	Throughput	NumberExited	Total	9,982.1600	766.0000	163.0000	44.2107
		OutputBuffer	Throughput	NumberExited	Total	9,981.9200	765.0000	163.0000	44.2555
		Processing	Throughput	NumberExited	Total	9,981.9200	765.0000	163.0000	44.2555
Sink	BadParts	[DestroyedEntities]	FlowTime	TimeInSystem	Average (Ho...	0.2988	0.2699	0.3259	0.0062
		InputBuffer	Throughput	NumberExited	Total	803.2800	745.0000	859.0000	11.4010
	GoodParts	[DestroyedEntities]	FlowTime	TimeInSystem	Average (Ho...	0.2998	0.2791	0.3248	0.0052
		InputBuffer	Throughput	NumberExited	Total	9,178.4800	365.0000	374.0000	44.5022

Problem 5.4:

I had to revisit the lecture video to figure out how to get the timing to work. I ended up using a rate table to schedule arrivals between 9am and 5pm (thank you for mentioning that the integers inputted there are actually interpreted as random exponential means). In addition, I set up two work schedules, but I had schedule these from 9am to 6pm in order to ensure that all customers already in the system were served and exited before the servers stopped processing. I supposed I could have used the fixed capacity setting, basically saying that servers need to stay at work until all customers are served.

Apologies for the eye chart (like the pun?), but the experiment results for 30 replications of 24 hours each are shown below:

Scenario			Replications		Responses										
✓	Name	Status	Required	Completed	AvgTIS	MaxTIS	RecUtil	RecQueAvgUnits	RecQueMaxUnits	VisionUtil	VisionQueAvgUnits	VisionQueMaxUnits	WrittenUtil	WrittenQueAvgUnits	WrittenQueMaxUnits
▶	Scenario 1	Completed	30	30 of 30	0.723128	1.21839	0.277...	0.596897	6	0.139821	0.0246351	1	0.576478	0.0165687	1

Problem 5.5:

