SYSC 4005 A
Discrete Simulation/Modeling

Deliverable 3

April 1st, 2021

Authors (Group 1)
Shoana Sharma (101036353)
Sarah Lamonica (101039213)
Mounica Pillarisetty (101041499)

Validation and Verification

Validation is the overall process of comparing the model and its behavior to the real system and its behavior. Verification means to check whether the software confirms a specification. The source code for input validation can be found in *validationAndVerification.py*. Since there is a lack of real system output data, we were not able to do the error evaluation and tests part of the model validation.

The file *validationAndVerification.py* validates the input data provided in the six .dat files. These files describe the inspectors inspection times and the workstations processing times. A different method was created for each element (inspectors and workstations). The function first calculates the average of the data provided in the .dat files (actual average) and then calls its respective function in *inputmodel.py* (developed in deliverable 2). Then the *validationAndVerification.py* calculates the average of the random values returned (random average). All of this data and the difference between the actual average and the random average is shown in the tables below. The difference is shown in % form.

There are two tables below, the first is for sample size of 1000 and the second is for sample size 50000. The sample sizes are arbitrary values to help with verification of the model.

The process in which we verified our model is by comparing two drastically different sample sizes with respect to the actual and random averages and their difference. It was clear that the actual average and the random average correlated well with a sample size of 1000 as the difference reached approximately a maximum of 3%. In order to verify our mode, the sample size of 50000 was used, with all the differences being below 0.3%. This was an indication that the input data that was being generated is accurate and was properly verified as well as validated.

Element	Actual Average	Random Average	Difference between Actual and Random Averages (%)
Inspector 1 for component 1	10.357909999999993	10.4359034653453	0.752984582269073
Inspector 2 for component 2	15.53690333333332	15.9937482374924	2.9403858308042006
Inspector 2 for component 3	20.632756666666666	20.3435083874909	1.401888675608054
Workstation 1	4.604416666666665	4.49423423423413	2.3929726696982527
Workstation 2	11.092606666666665	11.2349988835354	1.2836677721263188
Workstation 3	8.795580000000005	8.99543435305333	2.272213464641619

Table 1: Averages of all the elements for sample size of 1000.

Element	Actual Average	Random Average	Difference between Actual and Random Averages (%)
Inspector 1 for component 1	10.357909999999993	10.34894293482940	0.08657214795834925
Inspector 2 for component 2	15.53690333333332	15.50492356897435	0.20583100552836828
Inspector 2 for component 3	20.632756666666666	20.63542435809090	0.01292939895203456
Workstation 1	4.604416666666665	4.593848724820490	0.229517496161462
Workstation 2	11.092606666666665	11.10039838904908	0.0702424832733734
Workstation 3	8.795580000000005	8.779232342369778	0.18586219021630485

Table 2: Averages of all the elements for sample size of 50000.

There are other ways to validate our system and the alternatives include tracing the intermediate results and comparing them with observed outcomes. We can also check the simulation model output using various input combinations and compare final simulation results with the analytic results. Lastly, we can create programs or scripts to debug the program in sub-programs to validate the system. Programs can also be created to construct a structured walk-through policy in which more than one person is to read the program and validate the system.

Production Runs

The *Replications.py* calculates the service times and returns the results for all the elements in the Manufacturing Facility. The *Replications.py p*rogram also calculates the idle times, block times and the number of production for Workstation, Inspectors and Products respectively.

$$Average\ service\ time\ =\ total\ service\ time\ /\ 300\ (total\ number\ of\ data\ points\ from\ the\ .dat\ files)$$

Elements	Average Service Time
Inspector 1 (I1)	618.494052157
Inspector 2 for component 2(I2)	945.22222816
Inspector 2 for component 3(I3)	1274.54098613
Workstation 1 (W1)	277.230386365
Workstation 2 (W2)	677.852053529
Workstation 3 (W3)	522.901354347

Table 3: Service Times (Sample Data)

From Table 3, one can see that the time service time for I1 and W1 are comparatively much faster.

Elements	**Confidence Interval**
Blocked from Inspection Time:	
Inspector 1 (I1)	0.0990748910612 ± 0.0802883813785
Inspector 2* for component 2(I22)	846.192376962 ± 84.3618416394
Inspector 2* for component 3 (I23)	807.79593036 ± 111.578605526
Idle Time:	
Workstation 1 (W1)	407.049048374 ± 5.58369821716
Workstation 2 (W2)	4640.64190391 ± 182.616949565
Workstation 3 (W3)	9520.43044741 ± 825.541041057
Production:	
Product 1 (P1)	40.113 ± 0.352350808173
Product 2 (P2)	3.177 ± 0.122824213798
Product 3 (P3)	0.486 ± 0.0610149606695

*note: Will use I2 for Inspector 2 in general

Table 4: Confidence Interval for Elements (Sample Data)

Table 4, represents the *Confidence Interval* for elements in the Manufacturing Facility with 1000 replications with 95% confidence. The confidence interval was calculated using the python library libraries.

Analysis

Base Case:

In Table 3, the rate of production is higher when I1 and W1 are in use. There are more products assembled when compared to I2 at W2 or W3.

In Table 4, W1 has the lowest idle time when compared to W2 and W3. The production of P1 is also significantly greater than P2 and P3. The I1 is usually never blocked when compared to I2.

This also shows that the rate of production is very much dependent on the inspecting time for I1 which can be verified from Table 3 and Table 4. The production is much higher at W1 when compared to W2 and W3.

Check 1: Doubling the Production Rate for W1

Naturally W1 will continue to produce more products than W2 but this was possible because of the priority buffer that gave in components to W1. I1 is able to inspect more products because of this, leading to a higher production rate. There is no requirement for wait times as well as the buffer is never full due to the fast inspection times by I1.

Check 2: Same Production Rate for W1 and W2

The rate of production is the same for the workstations for this check. When this is true, then I1, I22 and I23's service time are at par such that the enough components are available for product assembly.

Check 3: I2's rate of inspection is faster than I1

As a result the production rate of I2 is faster that I1. The buffer tends to be full in this case and creates a slower production rate for W2 and W3. I2 now has to wait to continue the production. The buffer requires to send a component from I1 to W2 or W3 contributing to a much better production rate at W1 due to the priority buffer.

Conclusion:

This also shows that the rate of production is very much dependent on the inspecting time for I1 which can be verified from Table 3 and Table 4. The production is much higher at W1 when compared to W2 and W3. This is proved through the base case and the above checks using the simulation model.