Date: Feb 10th, 2023

SYSC 4005 – Project

Deliverable 1

Group 17

Adi El-Sammak – 101104212

Kareem El-Hajjar – 101109259

Khalid Kana'an – 101114513

Problem Formulation

The problem formulation for the manufacturing facility simulation study is to evaluate the performance of the system by measuring key metrics such as inspector block time, throughput, and average buffer occupancy. The study aims to analyze the historical data of the inspectors' and workstations' service times & average buffer occupancy to calculate the probability of each workstation being busy and each inspector being blocked. The objective is to enhance Inspector 1's delivery strategy in order to increase throughput and, if possible, decrease inspector block time. In this project, we will utilize the input data from the files servinsp1.dat, servinsp22.dat, servinsp23.dat, ws1.dat, ws2.dat, and ws3.dat to conduct the simulation and analyze the results.

Objectives

- To evaluate the facility's performance by measuring the throughput, probability of busy workstations, average buffer occupancy, and probability of inspector block/idle time.
- To identify areas of improvement in the facility's operation to optimize its performance and increase efficiency.
- To propose and recommend an alternative operating policy for Inspector 1, to improve the facility's performance.

Overall Project Plan

Method for Evaluating the Alternatives:

- Evaluation criteria: Throughput, Probability of workstation idle time, Average buffer occupancy, Probability of inspector being blocked.
- Simulation method using a Python module.

Number of People Involved:

- 3 students in a group.

Cost of the Study:

- Free since we are using Python and a free simulation module (SimPy).

Schedule:

- Deliverables: 4 bi-weekly deliverables.
- February 10th: Problem Formulation, Objectives & Overall Project Plan, Model Conceptualization, Source Code, and Model Translation.
- February 24th: Data Collection, Input Modeling, and Input Generation.
- March 10th: Model Verification & Validation, and Production Runs & Analysis
- March 24th: An Alternative Operative Policy, Conclusion, and Final Report.

Potential Alternative Design:

The objective is to introduce an alternate design by simulating the presence of a new inspector who will solely handle the inspection of component 3, thus relieving inspector 2 of that responsibility. This results in one inspector for each component, potentially speeding up workstation processing times, but also potentially causing inspector idle time to rise. Currently, inspectors are implemented as Python classes, each handling a specific component. To add a new inspector, a new class would be created to represent them, with some modifications to its behavior. Inspector 2's behavior would also be altered so it no longer inspects component 3. This would be done by modifying the method that generates values for inspector 2's service time on component 3, so it can instead be used to pass values to the new inspector, representing the time needed to service component 3.

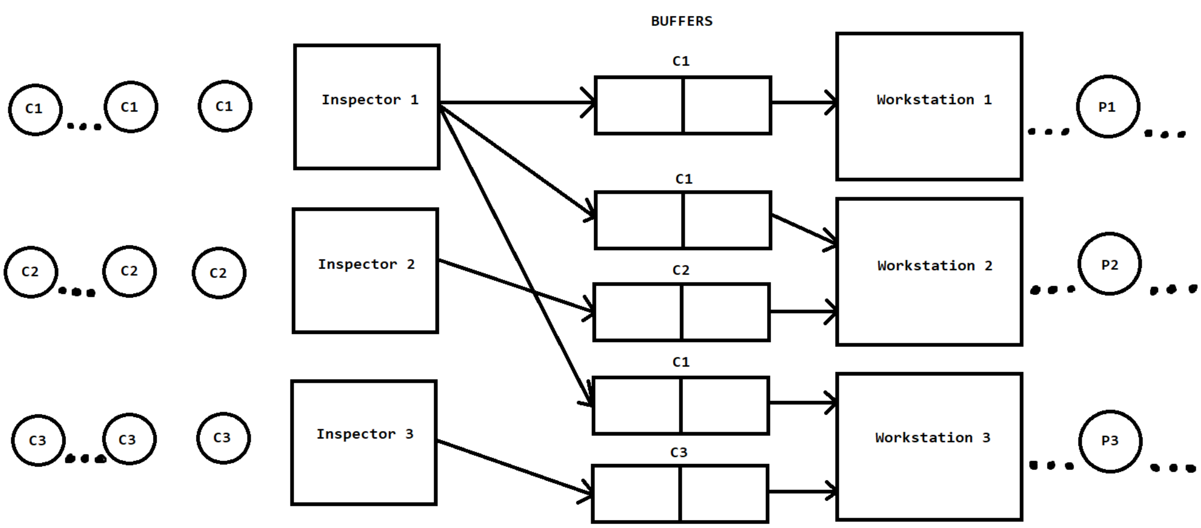

Figure 1 - Potential Alternative Design

Model Conceptualization

The simulation model will represent the manufacturing facility, including the components, inspectors, workstations, and buffers. The model will consider the routing policy of Inspector 1 and the random order of Inspector 2's inspection of components C2 and C3. The model will simulate the processing time at each workstation and the inspection time at each inspector. The model will track the buffer occupancy and the inspector idle time.

Model Translation

As a group, we decided to choose Python as the simulation language due to its versatility, user-friendliness, and the group's practical experience with it. Although MATLAB is also a suitable simulation language, we did not choose it since it is an unconventional software that we are not comfortable with. To help us with the simulation, we will utilize the "SimPy" Python library which is used to perform simulations on systems that interact through discrete events. We determined "SimPy" to be the most feasible simulation module available for Python as it is easy to use, elastic and there is an abundance of online documentation/community support.

The simulation language chosen for this project is Python. The model implementation includes the following files:

- Inspectors package:
  - Inspector1.py
  - Inspector2.py
  - Purpose:
    - These classes were created to represent the Inspectors which are responsible for inspecting & designating the components to their respective buffers.
    - Each class parses its corresponding data file, generates service times and then simulates its inspection. Once the component has been fully inspected, it is passed to the buffer with the least number of components in it.
- Workstations package:
  - Workstation1.py
  - Workstation2.py
  - Workstation3.py
  - Purpose:
    - These classes were created to represent the Workstations which are responsible for retrieving components from their respective buffers and assembling the product.
    - Each class parses its corresponding data file, generates service times, and then simulates its assembly of the product. Once the product has been fully assembled, the workstation's products counter is incremented.

- helperfunctions.py
    - Purpose:
        - Contains two functions that take a list of input data and converts it from string values to float & returns the mean value (find_list_mean function). The second function is called exponential_dist which takes the list of input data and generates random values using the exponential distribution of the mean value of the list.
- main.py
    - Purpose:
        - This file serves as the simulation run file which instantiates and starts the run function of Inspector and Workstation classes as SimPy processes.
        - Additionally, it prints out the results of the simulation by showing the execution times and average service times for each inspector and workstation.

The simulation model will use object-oriented programming concepts to represent the components, inspectors, workstations, and buffers as objects. The simulation module will use functions to execute the simulation using a SimPy simulation module and determine the quantities of interest.

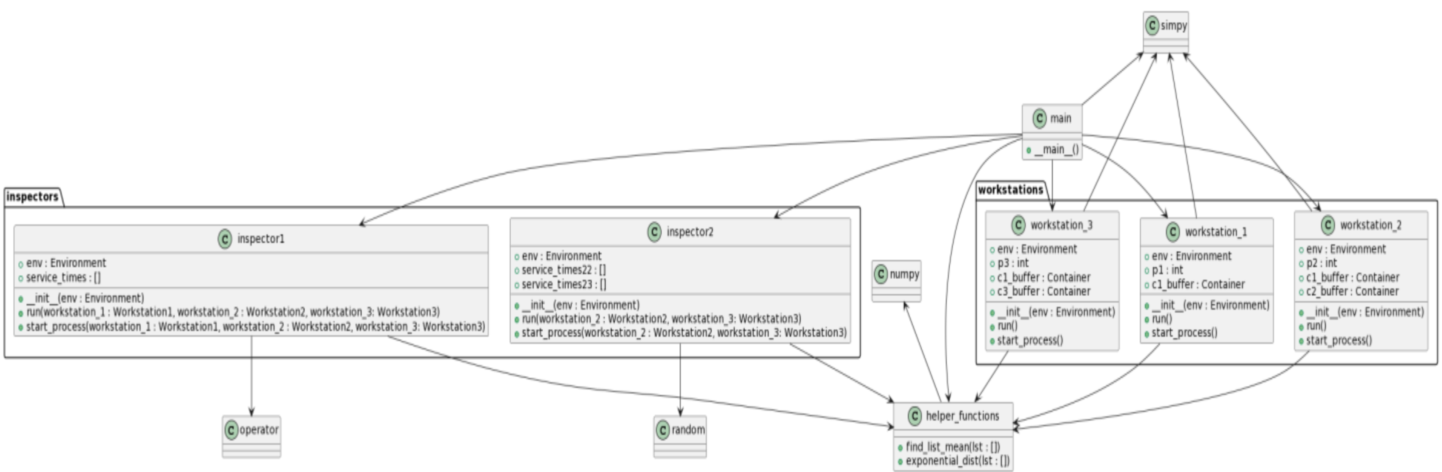

Figure 2 - UML Class Diagram

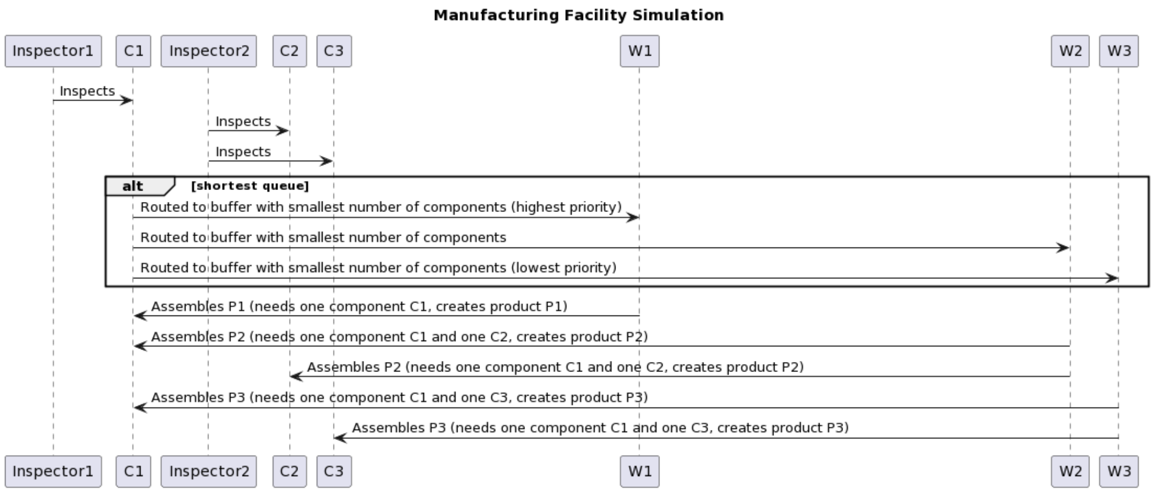

Figure 3 - Interaction Diagram

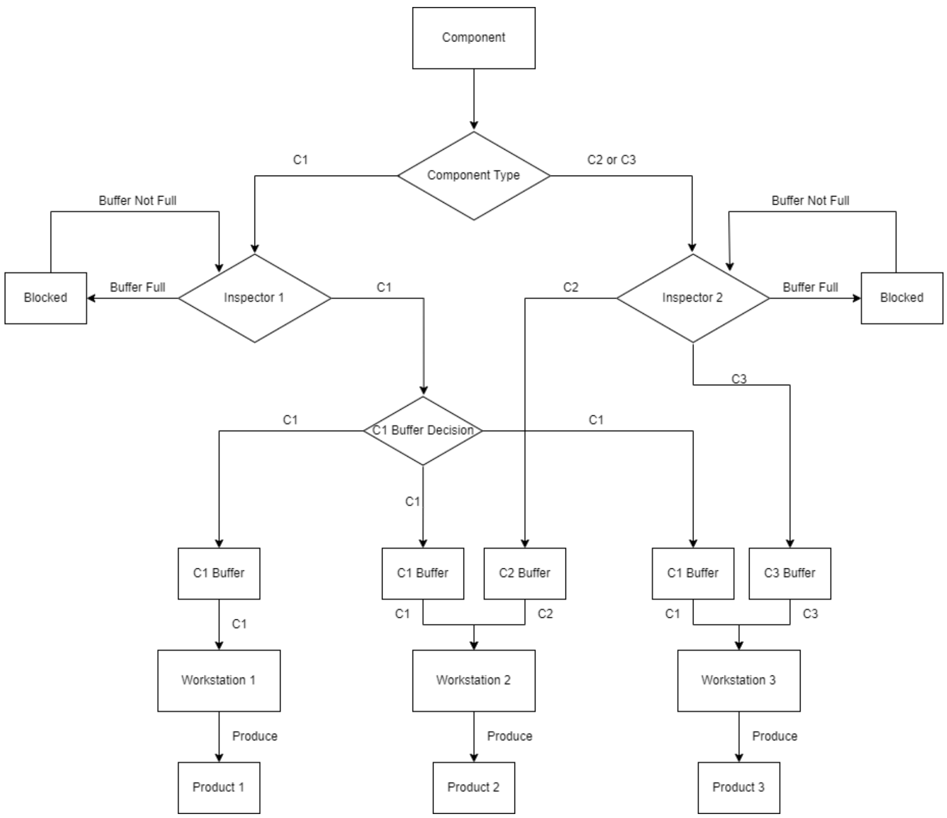

Figure 4 - Flow Chart

In conclusion, the simulation study of the manufacturing facility will provide valuable insights into its performance and assist us in finding the optimal alternative operating policy.