

# **Discrete Event Simulation Using Python's SimPy Library**

James C.K.K., Garo B., and Manny H.

MSDS 460: Decision Analytics

June 2nd, 2024

Table of Contents

Executive Summary-----2

Introduction----- 3

Literature Review----- 3

Method-----4

Results----- 4

Discussion----- 5

References-----7

## Executive Summary

This study conducted an analysis that employed discrete event simulation (DES; Allen, Spencer, and Gibson, 2015) in Python using the *SimPy* package ([simpy.readthedocs.io](https://simpy.readthedocs.io), 2020) to simulate processes and behaviors in a cafe and bistro business setting. The aim of this study sought to implement DES and explore ways to improve processes - namely, customers arriving and being served. This aim was explored by implementing DES to simulate a unique dining experience: one where the restaurant staff served the customer throughout their entire experience.

In this analysis, 1,000 customers were served and this scenario suggested that this would take nearly three full days. Following the implementation and interpretation of this DES exercise, there were four areas for the café and bistro to consider: 1) staff optimization, 2) process enhancements, 3) technology integration, and 4) customer experience. The unique dining experience should be revised to optimize the way in which servers are utilized. Unless the café and bistro expect to operate 24/7 then it may be prudent to assign servers to complete one or two specific activities rather than completing all activities per customer.

## **Introduction**

Discrete event simulation (DES) is a method of modeling a system as a sequence of distinct events where each event occurs at specific points in time. The value of DES lies in its utility to provide detailed insights into system dynamics, process inefficiencies, and about the impact of different operational strategies. Applied to a restaurant-type setting, DES can provide recommendations about how to optimize resource allocation, enhance customer satisfaction, and streamline operations by simulating varying service scenarios. This project aimed to implement DES to simulate service in a cafe and bistro setting. Simulating real-world behaviors in this manner is crucial for testing changes and making informed, data-driven decisions without disrupting actual operations.

## **Literature Review**

Zinoviev (2024) beautifully stated that simulation methods “[are] the priceless art of bringing to “life” systems and behaviors that would otherwise be prohibitively expensive...”. DES is one avenue by which this is accomplished. To execute this approach, Zinoviev outlines four main steps to implement DES: 1) building a model, 2) executing simulations, and 3) visualization, and 4) optimization.

Robinson (2014) explored using computer simulations to model service environments that involved serving customers such as in coffee shops. Robinson suggests that traditional simulation methods may not capture customer behavior well because it does not take into account the effects of a customer’s behavior on events. So, the article suggests that an adapted method where each customer is individually represented should be utilized, making it easier to plan things like shop layout. Unlike before, we don't need to explicitly simulate lines; they form

naturally as customers move around. The article shows an example of this method using Excel. It discusses the differences from older methods, the benefits, and ideas for further improvements.

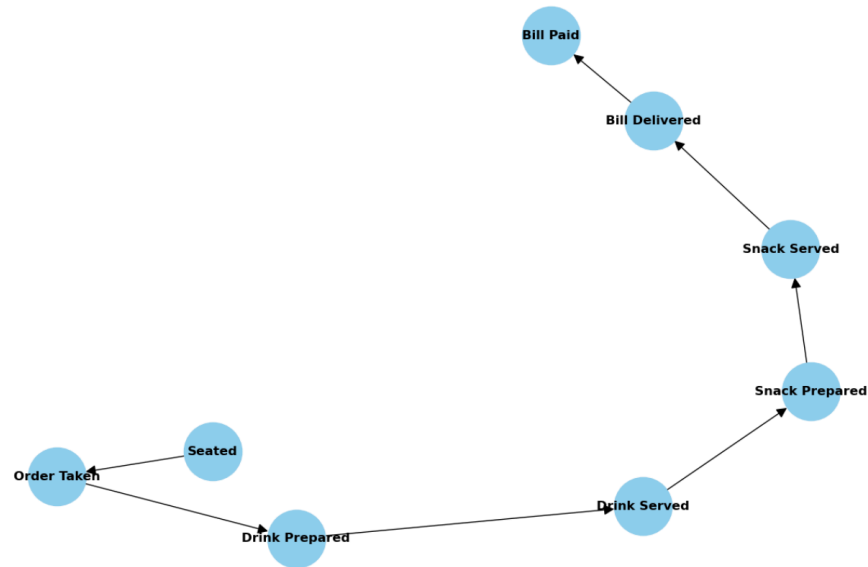
## Method

The goal of this analysis was to implement DES using Python and *SimPy* ([simpy.readthedocs.io](https://simpy.readthedocs.io)) in a real-world example (a café and bistro that serves customers). All analyses were conducted using Python programming within a Jupyter Notebook.

In this programming exercise, there were five servers that served one customer throughout the dining experience. In total there were eight activities that comprised this dining experience: 1) seating the customer, 2) taking the order, 3) preparing drinks, 4) serving the drinks, 5) preparing food, 6) serving the food, 7) bringing the bill, and 8) the bill being paid. All of these activities occurred in sequential order as listed. The DES was executed for 1,000 customers. Then, a graphical illustration was created using the *networkx* Python library. Finally, the output log of the DES was exported as a CSV file.

## Results

This DES analysis explored a unique dining experience where the staff handle all activities per customer. This DES example simulated serving 1,000 customers in this café and bistro. To serve 1,000 customers in this unique dining experience, the simulation log suggests that it would take nearly three full days with five servers. Below is a visual representation of the activities as produced by the *networkx*.



## Discussion

Following the implementation and interpretation of this DES exercise, there were four areas for the café and bistro to consider: 1) staff optimization, 2) process enhancements, 3) technology integration, and 4) customer experience.

First, the way in which staff are delegated tasks needs to be examined. In this scenario, staff handle all activities per customer. While this may be the case for some smaller or family-owned businesses, this may not be the case for larger businesses with more resources. This aspect of the simulation should be conducted in addition to this scenario such that the following simulation assigns one activity to one staff (e.g., Server A handles seating customers, Server B handles taking orders). An additional way to optimize staff is by increasing the number of staff during peak hours. Next, to enhance processes, there may be utility in streamlining the preparation of drinks and food orders.

This DES scenario may benefit by examining the effect of integrating technology into the business such as deploying a mobile application for ordering. This addition would add

complexity to the DES because it would add time to the activities involving preparing drinks and food, and also the activities involving serving those orders. This addition would increase the time for all customers with notable time increases for those customers who are physically present in the café and bistro.

Finally, this DES scenario presents challenges, limitations, and next steps. This analysis is limited by the fact that it is unlikely for a café and bistro to operate around the clock as this analysis suggested that serving 1,000 customers would take nearly three full days. This limitation should be addressed in a future revision unless the café and bistro anticipates operating 24/7. These changes and improvements all lead to increasing customer satisfaction and, in turn, revenue.

## References

- Allen M, Spencer A, Gibson A, et al. Right cot, right place, right time: improving the design and organisation of neonatal care networks – a computer simulation study. Southampton (UK): NIHR Journals Library; 2015 May. (Health Services and Delivery Research, No. 3.20.) Chapter 5, What is discrete event simulation, and why use it? Available from: <https://www.ncbi.nlm.nih.gov/books/NBK293948/>
- Robinson, S. Modelling without queues: adapting discrete-event simulation for service operations. *J Simulation* **9**, 195–205 (2015). <https://doi.org/10.1057/jos.2014.21>
- Team SimPy. (2023). *Documentation for SimPy*. [simpy.readthedocs.io](https://simpy.readthedocs.io).  
<https://simpy.readthedocs.io/en/latest/>
- Zinoviev, D. (2024, April 3). *Discrete Event Simulation: It's Easy with SimPy!* Arxiv.  
<https://arxiv.org/pdf/2405.01562>