A Report on

Hospital Simulation

Submitted for the Course Requirement of

Simulation And Modelling

Bachelor of Engineering in Software Engineering

under Pokhara University

Submitted by:

Bishnu Timilsena, 201746

Date :

28 January 2024

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Table of Contents

Introduction

Simulation

A simulation, in the context of this project, refers to the imitation of real-world processes or systems through the use of a computational model. In the "Hospital Simulation," the virtual representation enables the examination and experimentation of various hospital scenarios in a controlled and risk-free environment. Simulations are widely employed to gain insights into complex systems, test hypotheses, and optimize processes.

Hospital simulation

This project “Hospital SImulation”is a simulation of a patient flow in a healthcare setting using the SimPy library in Python. The simulation models the journey of a patient through different stages of care, including registration, triage, assessment by doctors in the Emergency Department (ED) or Acute Care Unit (ACU), and finally leaving the system.

Project Objectives

The primary objectives of this project are:

- To simulate and analyze the patient journey within a hospital setting.
- To assess the efficiency of resource utilization, including receptionists, nurses, and doctors, in handling a stream of patients.
- To understand the impact of various factors, such as inter-arrival times and assessment durations, on the overall system performance.
- To provide a platform for experimentation and improvement of hospital processes to enhance patient care.

Methodology

The simulation is based on a series of functions that generate patients and their activities. Patients arrive at the healthcare facility according to an exponential distribution, and their activities (registration, triage, and assessment) also follow exponential distributions. The healthcare professionals are modeled as resources in SimPy, and patients must wait for these resources to become available before they can proceed with their activities.

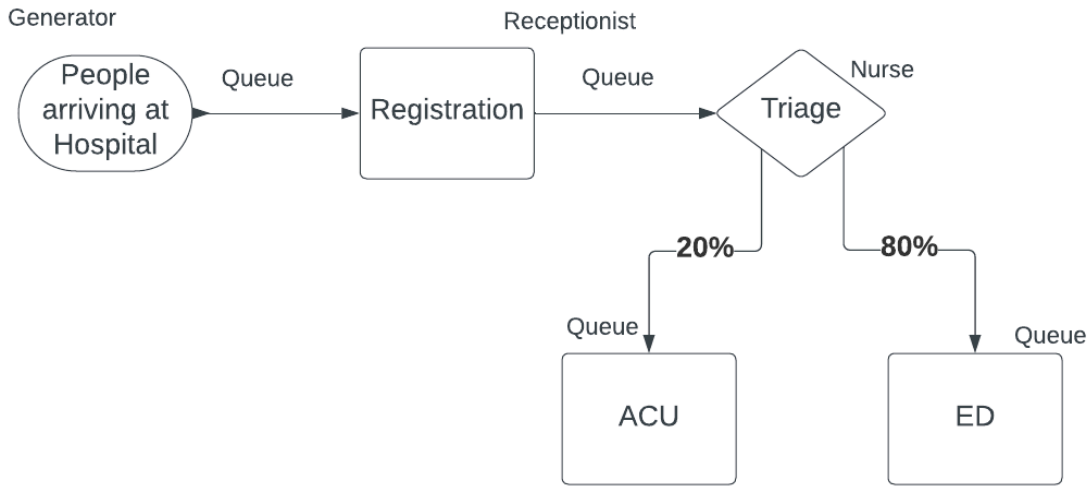

Fig:Flow of patients in hospital

Structure of the Code

simulation.py:

- `patient_generator` function:
    - Generates patients over time.
    - Uses the `activity_generator` function to create patient activities.
    - Patients arrive at random intervals.
    - Uses exponential distributions for delays between activities.
- `activity_generator` function:
    - Simulates the various activities a patient undergoes, such as registration, triage, assessments, and leaving the system.
    - Uses resource requests to model interactions with receptionist, nurse, ED doctor, and ACU doctor.
    - Decides whether the patient goes to ACU or ED based on a probability.
- `run_simulation` function:
    - Initiates the patient generation process and runs the simulation for a specified time (until=450).

main.py:

- Sets up the simulation environment, resources, and parameter values.
- Invokes the `run_simulation` function to initiate and run the simulation.

Simulation Parameters

- `ed_inter`: Mean inter-arrival time for patients.
- `mean_register`: Mean time for patient registration.
- `mean_triage`: Mean time for patient triage.
- `mean_acu_assess`: Mean time for ACU assessment.
- `mean_ed_assess`: Mean time for ED assessment.
- Resources (receptionist, nurse, ed_doctor, acu_doctor) are modeled with specific capacities.

Simulation Execution

The main script (`main.py`) sets up the simulation environment, defines resources, and executes the simulation for a specified number of patients (`num_patients`).

Output

The simulation outputs progress messages to the console, indicating the key events for each patient, such as queuing times and assessment durations.

Observations and Analysis

The simulation captures the patient flow through different stages of the medical system, allowing for analysis of resource utilization, waiting times, and patient pathways.

Future work

- Implement data collection during simulations.
- Integrate visualization tools for better analysis.
- Incorporate additional complexities, such as patient emergencies and resource constraints.
- Fine-tune parameters based on real-world data for more accurate simulations.
- The code currently focuses on simulating a fixed number of patients. To enhance realism, consider making the simulation continuous or dynamic.

Conclusion

The Hospital Simulation project provides a flexible framework for studying patient throughput and resource utilization in a hospital environment. By adjusting parameters such as arrival rates and service times, the simulation can be customized to represent different scenarios and aid in optimizing hospital processes.