Emergency Room Simulation: Patient Flow, Triage, and Resource Use

Aashna Suthar, Ash Arul
CS 4632 Modeling and Simulation
Section W01 Fall 2025

***Abstract*—Emergency rooms are one of the busiest areas in a hospital and are never easy to manage. Patients show up at random times and with different levels of seriousness. Some patients might have something small like a cut, while others come in with life-threatening problems. Doctors and nurses need to figure out who should be treated first and how to make the best use of their time and space. This project is about making a simulation of how patients move through an emergency room (ER) and how staff handle them. This project will use discrete-event simulation with random arrivals and severity-based triage to collect data on patient wait times, staff workload, and throughput. By testing out different triage rules and staffing levels, the project will show how the ER could perform under normal and busy conditions. By the end, it should show how simulation can help answer real-world questions in healthcare without needing to test things on actual patients.**

I. Project Overview

A. Project Title

Emergency Room Simulation: Patient Flow, Triage, and Resource Use

B. Domain

This project is focused on healthcare systems, more specifically how emergency rooms operate. ERs are a good system to study because they deal with a lot of uncertainty and fast decisions. Patients don't come in on a schedule, and every case is different. Some are minor and can wait, while others are critical and need help right away. Because of this, ERs are a strong example of a system where modeling and simulation can be useful to test ideas without risking patient safety.

C. Problem Statement

The problem this project looks at is how triage rules and staffing choices change how the ER runs. The ER always has a limited number of staff and rooms, but the flow of patients keeps coming no matter what. If the wrong decisions are made, wait times can get very high and critical patients may not get help in time. The main question is: how do different triage strategies and staff levels affect average wait times, the number of patients treated, and how busy the staff become? By simulating these factors, we can see which setups work better.

D. Scope

The project is only going to look at the main parts of an ER: patients arriving, getting triaged, waiting if needed, receiving treatment, and then leaving. To keep things focused, it will not include labs, surgeries, inpatient admissions, or ambulance routing. By setting these limits, the project stays manageable and still shows the main challenges of ER operations.

II. System Description

A. System Components

There are four main parts of the system:

- **Patients:** Every patient has an arrival time, a severity level, and a treatment time. Some are quick to treat while others take much longer. Critical patients should be seen before others.
- **Doctors and Nurses:** These are the staff who treat patients. They can only handle one patient at a time. Measuring how much work they do (utilization) is important to see if they are overworked or underused.
- **ER Rooms:** Patients need a room to be treated. Even if staff are free, they cannot work without space. This adds a realistic limit to the system.
- **Queue Manager:** This part decides who gets treated next. It follows the triage rules, meaning severe cases go first while less urgent cases wait longer.

B. System Dynamics

The system works in a cycle. Patients arrive randomly and are sorted by triage. The most serious cases move up in priority. Doctors and nurses treat one patient at a time and then move on to the next. Patients leave after treatment. Over time, the simulation shows how patients pile up, how long they wait, and how staff are used. Sometimes the system runs smoothly, but during busy times it shows stress and bottlenecks.

C. Core Models and Algorithms

The project will use simple but effective models:

- **Patient Arrivals:** Modeled with a Poisson distribution to show random but steady arrivals.
- **Triage Rules:** Patients are sorted with a priority queue so critical ones are seen first.
- **Service Times:** Treatment times are random, using exponential or lognormal distributions depending on severity.

D. Assumptions

The simulation will make a few simplifications:

- Triage is instant and always correct.
- Treatment time only depends on severity, not on the individual patient.

- The number of rooms equals the number of doctors, so space matches staff.
- No patients leave before being seen, even if wait times are long.

III. Implementation Approach

The simulation will be coded in the Programming Language: Python, because it is easy to use and has many libraries for randomness and graphs. I will use VS Code for development and GitHub for version control so that the work is backed up and changes are tracked. The simulation will follow the discrete-event approach, where events like arrivals, service starts, and completions are what move the system forward. I will collect data on things like average and max wait times, how many patients get treated, how often staff are busy, and whether critical patients are seen in less than 30 minutes. This will help me compare different setups and see what works best.

IV. Literature Review

This part will cover at least five sources that talk about hospital simulation, queuing, triage, or patient flow. For each source, I will explain the main idea, how it relates to my project, and what I will take from it.

- Jun, Jacobson, and Swisher [4] review the use of discrete-event simulation in health care. Their work shows that simulation is a common way to study patient flow and system performance. This supports our choice to use discrete-event simulation for modeling the emergency room.
- Green [2] focuses on patient flow and reducing delays in healthcare delivery. The main idea is that bottlenecks can be studied with queueing models. This connects to our project because we are analyzing waiting times and how resources are used. We will use this to compare different resource setups.
- The Emergency Severity Index guide [1] explains how triage is performed in real hospitals. It assigns patients to priority levels based on severity. This is directly related to our model because we use priority scheduling to represent triage. We will follow this idea to give patients different priorities in the queue.
- Wiler et al. [6] discuss emergency department crowding and the need for interventions. They highlight that long waits and lack of capacity reduce quality of care. This relates to our project because we can test how changes in staff or rooms affect crowding. From this, we can take ideas about what scenarios to simulate.
- Van den Bergh et al. [5] provide a review of nurse scheduling methods. The main idea is that efficient scheduling improves staff utilization and patient care. Our project connects because staff scheduling is one of the resources we want to optimize. This source gives us background for building scenarios with different schedules.
- Herring and Wilper [3] apply queueing theory to emergency departments. They explain how queues affect wait times and resource use. This supports our model since we use queueing for patient arrivals and treatment. It will help us measure the impact of different queue setups.

V. UML Diagrams

A. Class Diagram

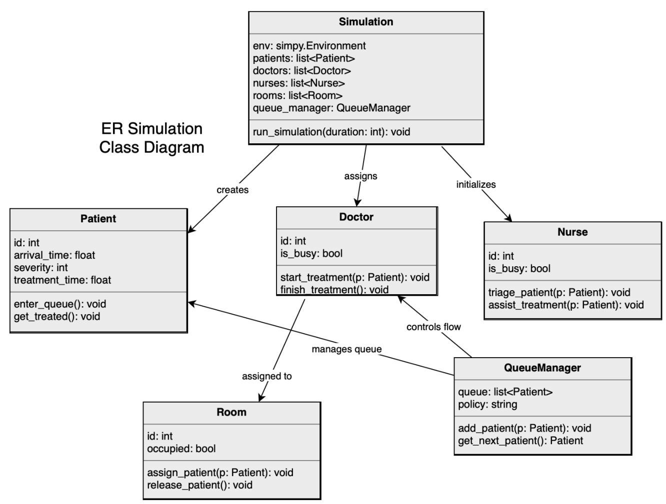

Fig. 1. Class Diagram for ER Simulation

This diagram shows the main objects in the system and how they interact. The Simulation class manages the overall environment and connects to other core components such as Patients, Doctors, Nurses, Rooms, and the QueueManager. Each class has its own attributes and behaviors: Patients have properties like arrival time and severity, Nurses handle triage, Doctors provide treatment, Rooms can be occupied or released, and the QueueManager organizes the waiting line based on priority. Together, these classes illustrate how the ER system is structured in code and how the pieces fit together to represent patient flow and resource use.

B. Activity Diagram

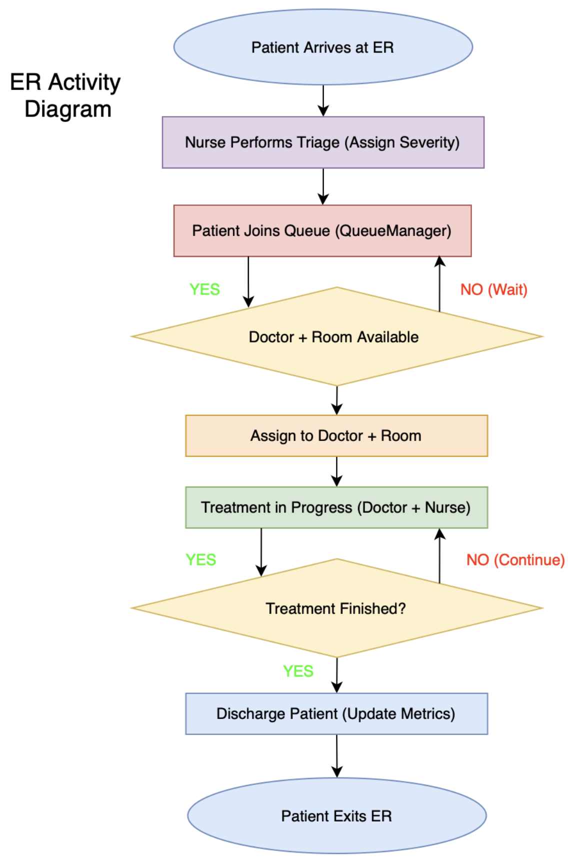

Fig. 2. Activity Diagram for ER Simulation

This diagram represents the flow of events from when a patient arrives at the ER to when they exit the system. Patients are first triaged by a Nurse, then they enter the Queue that is managed by the QueueManager. If a Doctor and Room are available, the patient is assigned to begin treatment; if not, they remain waiting in the queue. Treatment continues until finished, at which point the patient is discharged and leaves the system. The decision points (availability of staff/rooms and whether treatment is done) capture the uncertainty and variability that can occur in real emergency departments.

VI. Repository and Setup

I will create a GitHub repository called ”cs4632-er-sim”. It will have a ”README” that explains the project, a Python ”.gitignore”, and folders for source code and documents. The link will be added in the report and also in the D2L submission comments. This makes sure the work is organized and backed up throughout the semester.

GitHub Repository

The source code and project files are available on GitHub. The URL is https://github.com/asharulcoding/cs4632-er-sim.

VII. Conclusion

This project will build a simulation of an emergency room to see how patient flow and staffing choices affect performance. Using discrete-event simulation with random arrivals and severity-based triage, it will collect data on patient wait times, staff workload, and throughput. The idea is to keep the model simple but realistic enough to learn from. By the end, the project should show how simulation can help answer real-world questions in healthcare without needing to test things on actual patients.

References

[1] Agency for Healthcare Research and Quality. Emergency severity index (esi): A triage tool for emergency department care, version 4. Technical Report Publication No. 12-0014, AHRQ, 2012.

[2] Linda V. Green. *Patient Flow: Reducing Delay in Healthcare Delivery*. Springer, New York, 2006.

[3] Andrew Herring and Andrew P. Wilper. Using queueing theory to understand and optimize emergency department operations. *Western Journal of Emergency Medicine*, 20(4):1–7, 2019.

[4] Jin-Young Jun, Sheldon H. Jacobson, and James R. Swisher. Application of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society*, 50(2):109–123, 1999.

[5] Jeroen Van den Bergh, J. Beliën, P. De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review with a focus on nurse scheduling. *European Journal of Operational Research*, 226(3):367–385, 2013.

[6] Jennifer L. Wiler, Sherry Welch, Jesse M. Pines, Jeremiah D. Schuur, Nicholas Jouriles, and Mitchell A. Ross. Emergency department crowding: Time for interventions and policy evaluations. *Annals of Emergency Medicine*, 62(6):S69–S77, 2013.