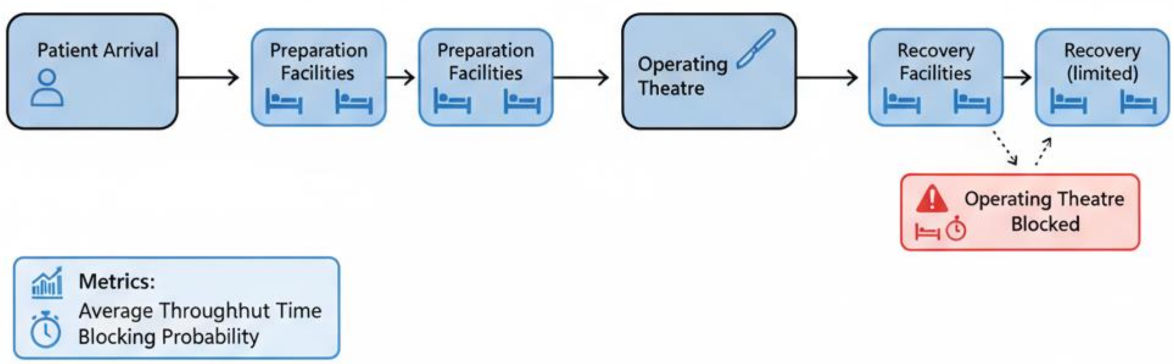
1. Process Modelling Goal

To model a continuous stream of patients that go through ***preparation → operation → recovery.*** Main performance measures:

- Average throughput time per patient (arrival → departure).
- Probability (and fraction of time) the operating theatre is blocked because no recovery bed is available.

Additional feature: Staff shifts / time-of-day effects that change effective capacities and service rates over time (e.g., fewer prep staff at night, slower service during shift handovers). This creates time-varying resource availability and arrival rates.

Modelling paradigm: Process-based — each patient is an active process following a lifecycle, requesting and releasing resources (pools) as needed. Resources have capacities that can change over time (shifts).



2. System components and variables

2.1 Resources (shared)

Name	Type	Description / role	Initial capacity (example)
PrepPool	Resource pool	Preparation stations; several identical units	3
Theatre	Resource	Operating theatre; single unit	1
RecoveryPool	Resource pool	Recovery beds/rooms; limited units	2

2.2 Entities

- **Patient** — active process. Attributes: id, arrival_time, type (optional), start_prep, end_prep, start_op, end_op, start_rec, end_rec, departure_time, priority_flag (if needed).

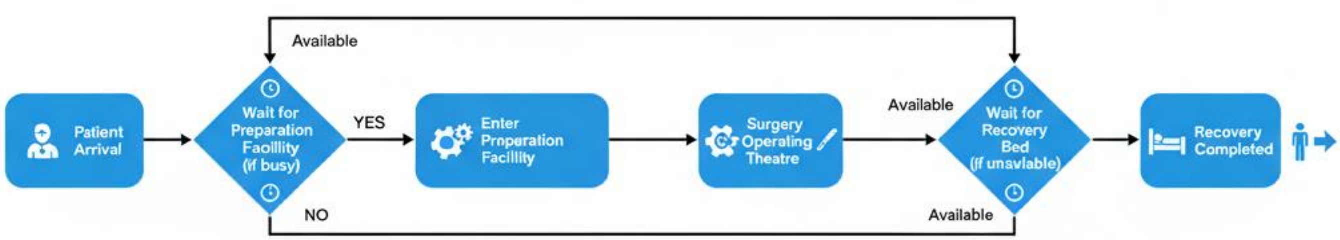
2.3 Time-varying elements (for shifts/time-of-day)

- capacity_prep(t): number of active prep units at time t (e.g., 3 daytime, 2 night).
- service_rate_prep(t): average preparation time distribution parameters depending on time of day.
- capacity_recovery(t): may be static or affected by staff availability (e.g., elective closures).
- arrival_rate(t): patient arrival intensity (e.g., higher daytime, lower night).

2.4 State variables & counters

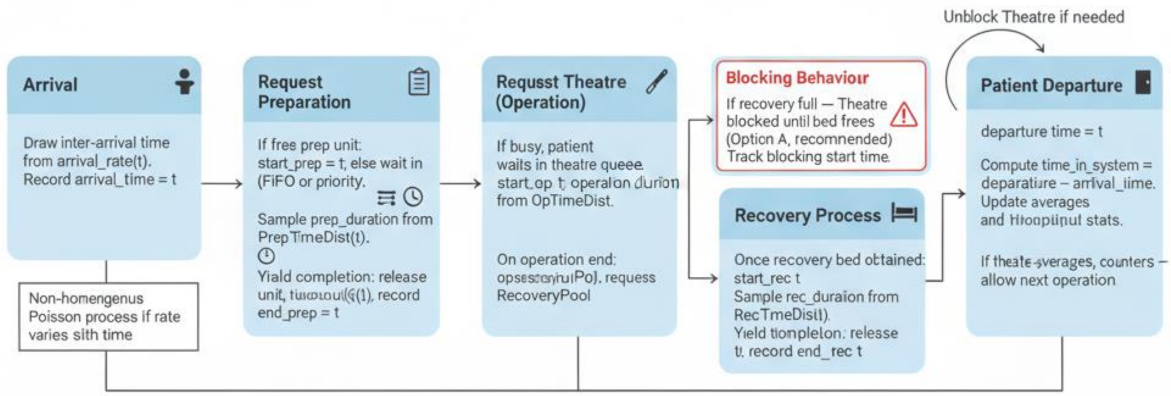
- num_in_prep_queue, num_in_theatre_queue, num_in_recovery_queue
- num_busy_prep, theatre_busy (0/1), num_busy_recovery
- total_patients_arrived, total_patients_departed
- sum_time_in_system (for mean throughput)
- theatre_blocked_time (cumulative time theatre is blocked)
- num_blocking_events (count of times theatre attempted to finish operation but no recovery bed)
- Time stamps for events, per-patient records (for distributions)

3. Patient process: lifecycle sketch



Each patient is a process that performs the following ordered sequence. For clarity, each action specifies what resource it requests, what it does while holding it, and what it does on release.

Lifecycle steps (detailed)



1. **Arrival**
   - Draw inter-arrival time from arrival_rate(t) (nonhomogeneous Poisson process if arrival_rate varies by time).
   - Record arrival_time = t.
2. **Request preparation**
   - yield PrepPool.request() — if a free prep unit exists, patient starts immediately; otherwise queues (FIFO or priority).
   - When granted: start_prep = t.
   - Preparation duration sampled from PrepTimeDist(t) (may depend on time of day or staff skill).
   - yield timeout(prep_duration).
   - On completion: release prep unit (PrepPool.release()), record end_prep = t.
3. **Request theatre (operation)**
   - yield Theatre.request(). If theatre busy, patient waits in theatre queue.
   - When granted: start_op = t.
   - Operation duration sampled from OpTimeDist (could be constant or stochastic).
   - yield timeout(op_duration).
   - On **Operation End**:
     - Attempt to request RecoveryPool for transfer.
     - If a recovery bed is available immediately:

            - start_rec = t; release Theatre (theatre becomes free); proceed to recovery stage.
        - **If all recovery beds are occupied**:
            - **Blocking behaviour:** Option A (recommended): patient *remains in Theatre* holding the theatre resource until a recovery bed is free. Mark theatre as **blocked**. Track blocking start time.
            - Option B (alternate): patient is moved to a theatre-adjacent queue / holding area (still occupying operating theatre or special holding resource). Implementation uses whichever matches system semantics — in this assignment we use Option A (theatre held until recovery bed free), because the task explicitly measures theatre blocked time.

4. **Recovery**
    - Once a recovery bed is obtained (either immediately after operation, or when one becomes free and theatre is unblocked): start_rec = t.
    - Recovery duration sampled from RecTimeDist (may vary by surgery or by time of day).
    - yield timeout(rec_duration).
    - On completion: release RecoveryPool, end_rec = t.
5. **Departure**
    - departure_time = t.
    - Update statistics: time_in_system = departure_time - arrival_time; increment counters, add to sums for averages. If this release unblocks theatre and theatre is waiting, allow theatre to start next patient.

Notes on blocking

- When theatre finishes operation but cannot release patient to recovery, theatre remains occupied by the patient and cannot start another operation. The model must:
    - Record blocked_start when theatre cannot hand off patient.
    - Record blocked_end when recovery bed becomes available and transfer occurs.
    - Update theatre_blocked_time += blocked_end - blocked_start.
    - Increase num_blocking_events by 1.

5. Essential data to collect & monitoring plan

Per-patient data (record for each patient)

- id, arrival_time, start_prep, end_prep, start_op, end_op, start_rec, end_rec, departure_time
- blocked_flag (true if operation end had to wait), blocked_duration (if any)
- patient_type (optional — e.g., elective vs emergency)

System-level time series (sampled at intervals or updated on events)

- num_waiting_prep(t), num_busy_prep(t)
- num_waiting_theatre(t), theatre_busy(t), theatre_blocked(t)
- num_busy_recovery(t), num_waiting_recovery(t)
- capacity_prep(t), capacity_recovery(t) (to verify shifts)

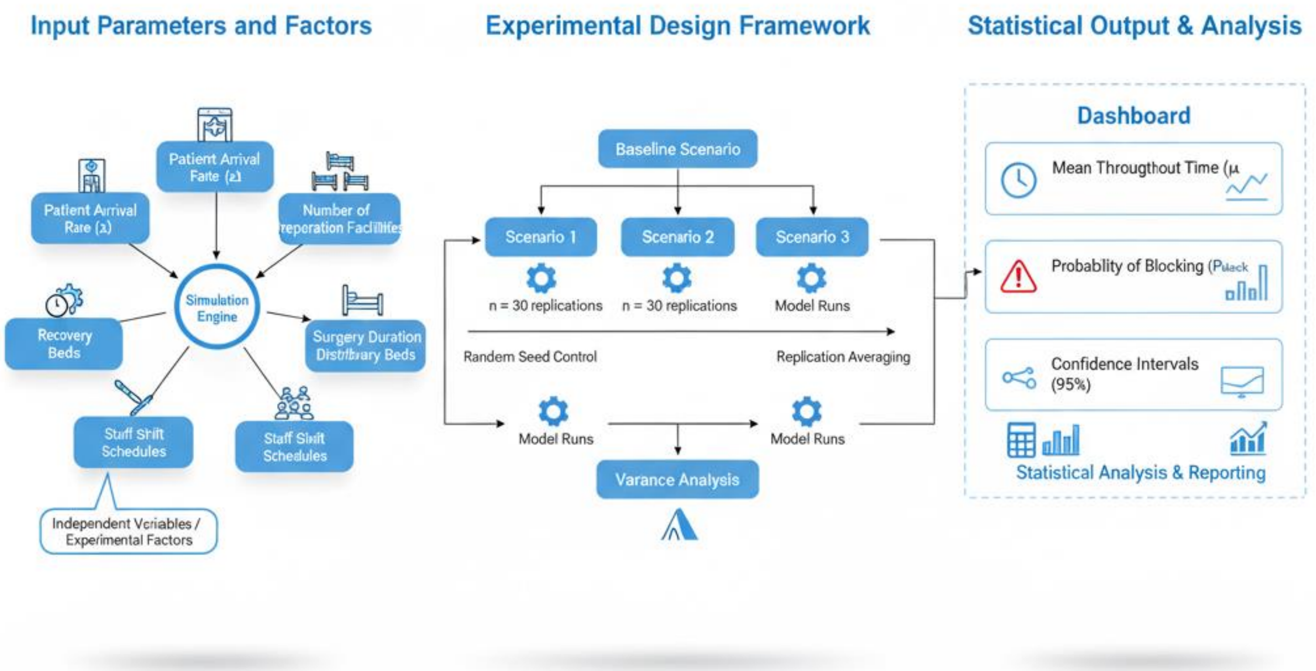
Aggregated metrics (computed after replication)

- **Average throughput time** = mean of departure_time - arrival_time.
- **Blocking probability**:
  - Option 1: fraction of completed operations that experienced blocking = num_blocking_events / total_operations.
  - Option 2: fraction of simulation time theatre is blocked = theatre_blocked_time / simulated_time.
  - Both are useful: Option 1 is event-based; Option 2 measures utilization impact.
- **Queue statistics**: average and max queue lengths per queue.
- **Resource utilization**: util_prep = total_busy_time_prep / (sim_time * nominal_prep_capacity) — care with time-varying capacity; compute utilization per shift or as integral over time: $\int$busy(t) dt / $\int$capacity(t) dt.
- **Delay distributions**: histograms of waiting times for prep, theatre, and recovery.
- **Time-of-day breakdowns**: compute metrics separately for day and night periods.

Logging for reproducibility

- Recording random seeds and replication IDs, shift schedule, distribution parameters, and run length.

6. Suggested statistical settings and experimental design



6.1 Warm-up and run length

- Use a **warm-up period** to avoid bias from initial empty system (common in open arrival processes). Example: discard first 8 hours (or until system reaches steady-state for stationary parts). For time-varying (day/night) experiments, prefer **multiple repeating cycles** (e.g., simulate several 24-hour days) and discard the first day.
- **Run length**: simulate N_days (e.g., 30 days) per replication to capture daily patterns, or simulate a continuous 720 hours if longer horizon desired.

6.2 Replications

- Use multiple independent replications (e.g., 30–50) with different RNG seeds to construct confidence intervals.
- For each metric compute sample mean and 95% CI.

6.3 Scenario experiments

Run these scenarios and compare outcomes:

1. **Baseline**: fixed capacities, no shifts, constant arrival rate (for sanity check).
2. **Time-of-day arrivals**: time-varying arrivals but fixed capacities.

3. **Shifts**: time-varying capacities + time-varying service performance (the proposed feature).
4. **Sensitivity**: vary recovery capacity (e.g., 1,2,3 beds) to see blocking sensitivity.
5. **Handover effect**: include short handover slowdowns and measure their effect.

6.4 Key comparisons

- Throughput time (mean and CI) across scenarios.
- Blocking probability by scenario.
- Peak queue length and utilization during day vs night.
- Impact of staff reductions on theatre blocked time.

6. Example parameter set (fillable — use for experiments)

Parameter Category and Variable Names	Values and Distributions	Notes / Observations
Arrival process		
λ_day = per hour (average interarrival min)		
λ_night = per hour (average interarrival min)		
Preparation time		
Day: or Gamma(α, β) as desired		
Operation time		
Normal(( )), tructed to > (min		
Recovery time		
Theatre =		
PrepPool, day = night = RecoveryPool) minutes)	Effective prep cap'uced by night = night =	
Shift handover		
(minutes prep capacity at shift change Effective prep capacity by service times increased by (22222?%)	888 =	

- Arrival process:
  - λ_day = 6 per hour (average interarrival 10 min)
  - λ_night = 2 per hour (avg interarrival 30 min)
- Preparation time:
  - Day: Exp(mean=20 minutes) or Gamma(α,β) as desired
  - Night: Exp(mean=25 minutes)
- Operation time:
  - Normal(mean=60 min, sd=10 min) truncated to >20 min
- Recovery time:
  - Exp(mean=120 minutes)
- Capacities:
  - PrepPool: day=3, night=2
  - Theatre: 1
  - RecoveryPool: day=2, night=1
- Shift handover:
  - 10 minutes at shift change where effective prep capacity reduced by 1 or service times increased by 20%.