TA: Xifan Zheng
Email: zhengxifan0403@gmail.com

Welcome to CPSC 441!

Today's Tutorial

Discrete-event simulation(DES) "models the operation of a system as a discrete sequence in time. Each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation can directly jump in time from one event to the next."

Components of DES

- Simulations contain

– **activities** where things happen to entities during some time (which may be governed by **a probability distribution)**

– **queues** where entities wait an undetermined time

– **entities** that wait in queues or get acted on in activities. entities can have attributes like kind, weight, due date, priority

Overview

- System is modeled as a set of entities that affect each other via events (msgs)
- Each entity can have a set of states
- Events happen at specific points in time and trigger state changes in the system
- Discrete event means that time advances until the next event can occur

  - time steps during which nothing happens are skipped

  -duration of activities determines how much the clock advances

Implementation

- Typical implementation involves an event list, ordered by time
- Process events in (non-decreasing) timestamp order, with seed event at t=0
- Each event can trigger 0 or more events
  - Zero: "dead end" event
  - One: "sustaining" event
  - More than one: "triggering" event
- Simulation ends when event list is null, or desired time duration has elapsed

Simple Bank example

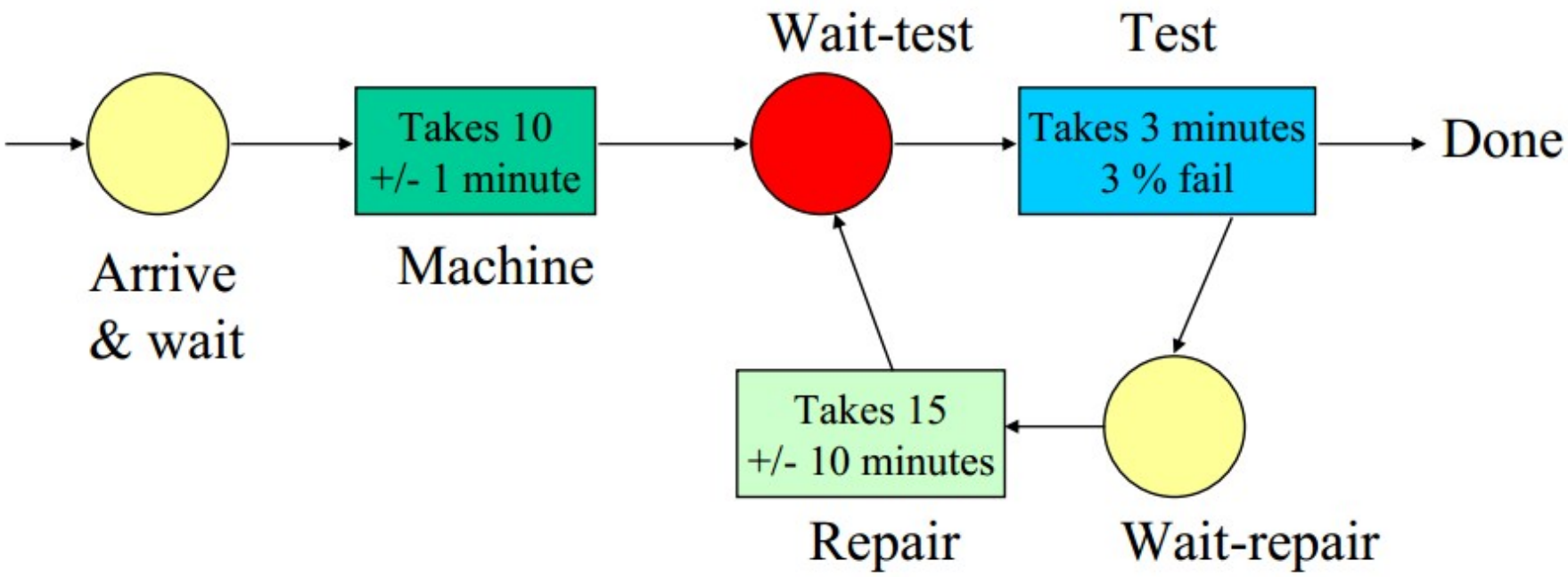
- **Entities**: Customer-queue and tellers
- **Events**: Customer-arrival and Customer-departure
- **States**: Number-of-customer-in-queue (from 0 to n) and teller-status(busy or idle)
- **Random variables**: Customer-interarrival-time and teller-service-time

Simulation of bank example

the event CUSTOMER-ARRIVAL at time t would:

- if the CUSTOMER_QUEUE was empty and TELLER was idle, the subsequent event CUSTOMER-DEPARTURE occur at time t+s, where s is a number generated from the SERVICE-TIME distribution.
- Otherwise, the customer will be put in the queue and wait all customers before him to be served, and then get served and leave at time $t+s_1+s_2+\ldots+s_n+s$, where s_n in the random value of service time of nth customer in queue.

More complicated simulation example



Note: If Wait-repair and Wait-test both are full, the system will become deadlocked as soon as the next unit needs repair because the test station will be unable to unload and take the next unit.

Poker game example

Setting: Four players get 1 card each round and the one with biggest hand value win the wages on table.

- ✓8 rounds maximum
- ✓each player has original 10 bank roll
- ✓wages from 1-5
- ✓card values from 1-10

Simulation of poker game

- For each round

1. Report bank balance for each player
2. Give out card with different value (no duplicate value allowed; bankrupt players will not be allowed to play any more)
3. Collect wages from each player and calculate total wages
4. Determine the winner and give him all wages from this round

More resources

- This simple simulation source code can be found on course webpage (porker.c)
- You can also find a continuous event simulation on course webpage (barber.c)

Thanks for attending!