Discrete Event Simulation

CS 681 and CS 462

Fall 2013

What/Why is a Queue?

- The systems whose performance we study are those that have some *contention* for *resources*
  - If there is no contention, performance is in most cases not an issue
  - When multiple "users/jobs/customers/ tasks" require the same resource, use of the resource has to be regulated by some discipline

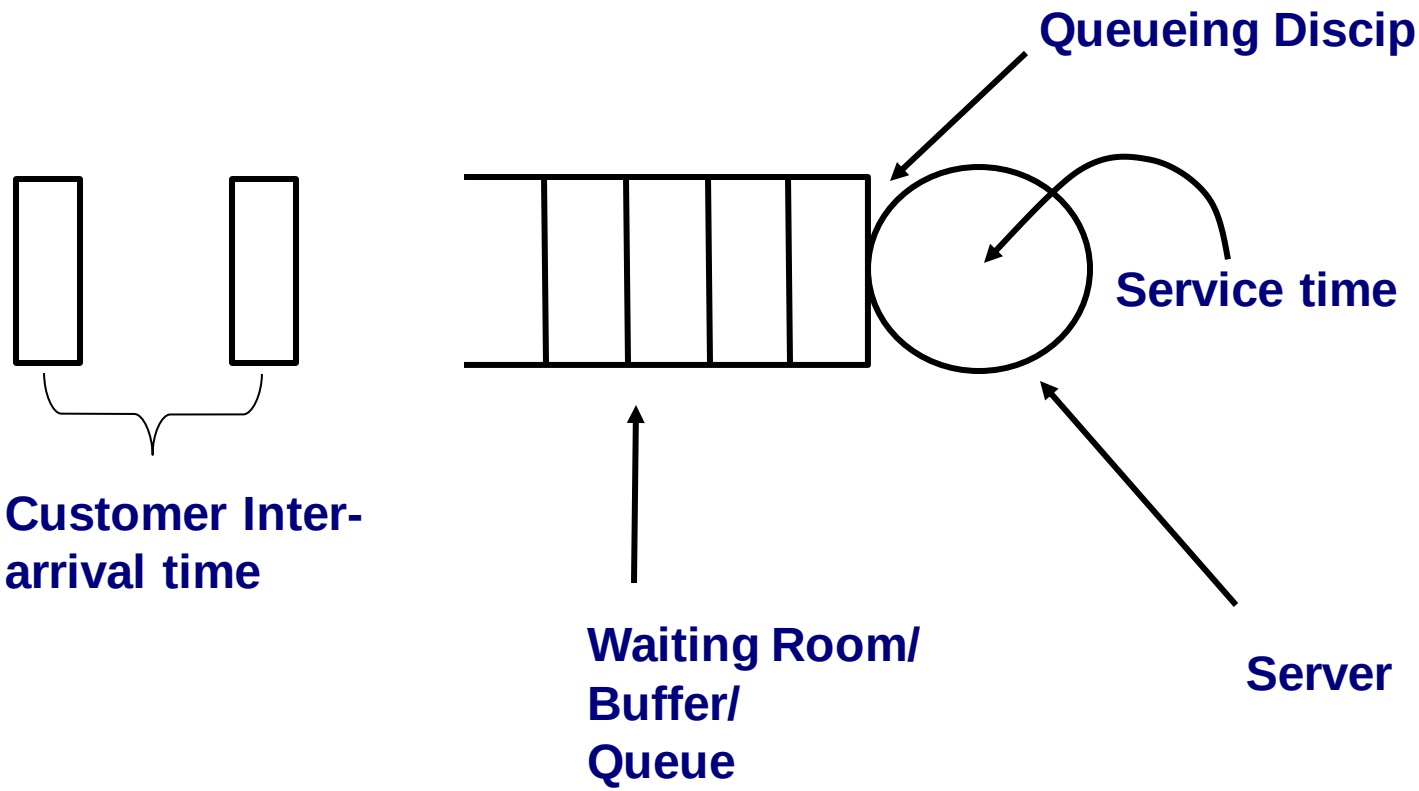
…What/Why is a Queue?

- When a customer finds a resource busy, the customer may
  - Wait in a “queue” (if there is a waiting room)
  - Or go away (if there is no waiting room, or if the waiting room is full)
- Hence the word “queue” or “queuing system”
  - Can represent any resource in front of which, a queue can form
    - In some cases an actual queue may not form, but it is called a “queue” anyway.

Examples of Queuing Systems

- CPU
  - Customers: processes/threads
- Disk
  - Customers: processes/threads
- Network Link
  - Customers: packets
- IP Router
  - Customers: packets
- ATM switch:
  - Customers: ATM cells
- Web server threads
  - Customers: HTTP requests
- Telephone lines:
  - Customers: Telephone Calls

Elements of a Queue



Elements of a Queue

- Number of Servers
- Size of waiting room/buffer
- Service time distribution
- Nature of arrival "process"
  - Inter-arrival time distribution
- Queuing discipline: FCFS, priority, LCFS, processor sharing (round-robin)

Parameters of a Queuing System

- Number of Servers: 1,2,3….
- Size of buffer: 0,1,2,3,…
- Service time & Inter-arrival time
  - Given as probability distribution
  - Or as “trace” or log

Queue Performance Measures

- Queue Length: Number of jobs in the system (or in the queue)
- Waiting time (average, distribution): Time spent in queue before service
- Response time: Waiting time+service time
- Utilization: Fraction of time server is busy or probability that server is busy
- Throughput: Job completion rate

How can we calculate queue performance measures?

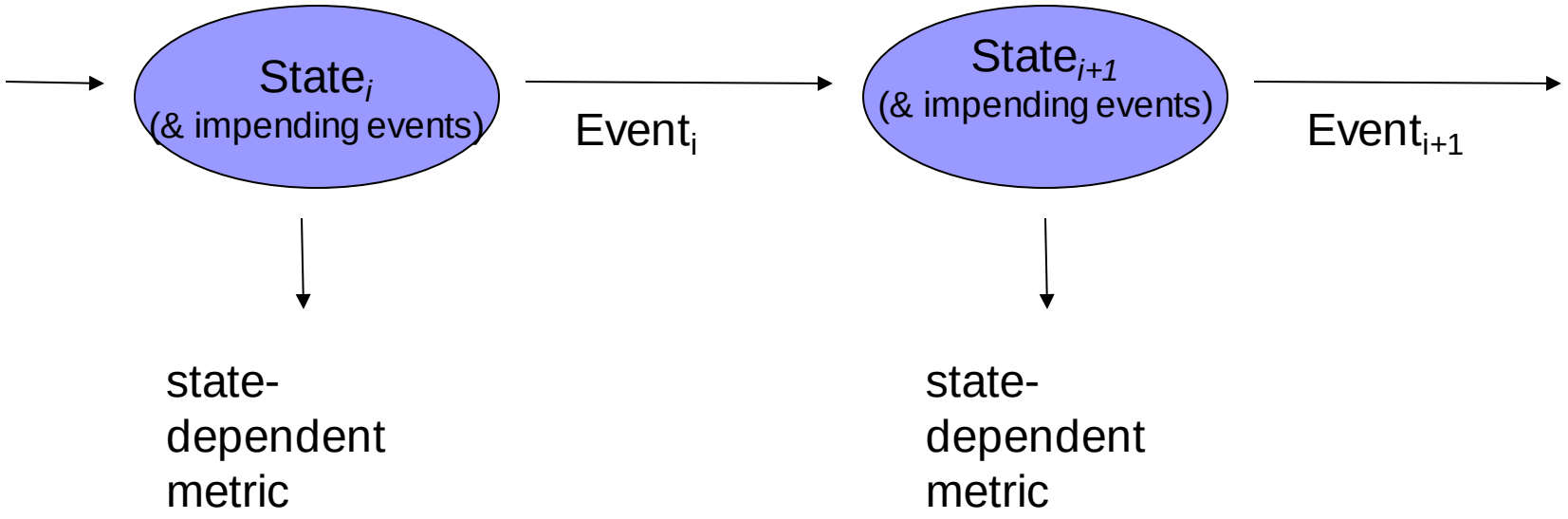
- Maths (probabilistic analysis)
- Write a program that “simulates” the entire system.

Discrete Event Simulation

- A general methodology for studying behaviour of dynamic systems & for calculating system performance metrics
- A dynamic system has
  - *a state* (performance metric is related to the state)
  - *A number of impending events*
    - When an event happens, the state of the system changes
    - When state of the system changes, new events become possible

Discrete Event Simulation

- System can be visualized as being in a perpetual "loop" of changing state and events
- The "discrete" means that system state (is assumed to) change at *discrete* points in time, not continuously
- These discrete points in time are when "events" happen
- Thus Discrete Event Simulation



Example: queuing system

- State?
  - Server busy or not
  - Customers waiting in the queue, their service times
- Impending events?
  - Arrival of a new customer
    - State change?
      - (buffer state, server state if it was idle)
    - New events?
      - (next arrival)
  - Departure of a customer who was being processed
    - State change?
      - (server may become idle)
    - New events?
      - (if next customer started then, its departure is now impending)

Sample queue simulation

- show spreadsheet

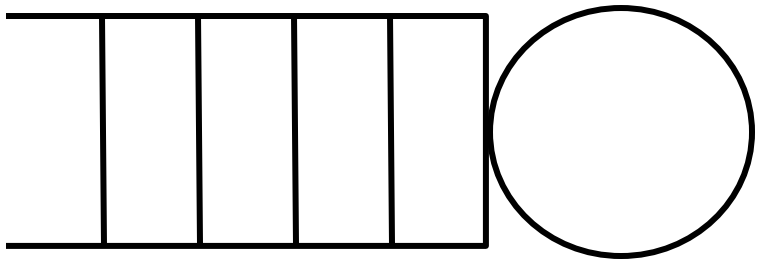
General Simulation Logic

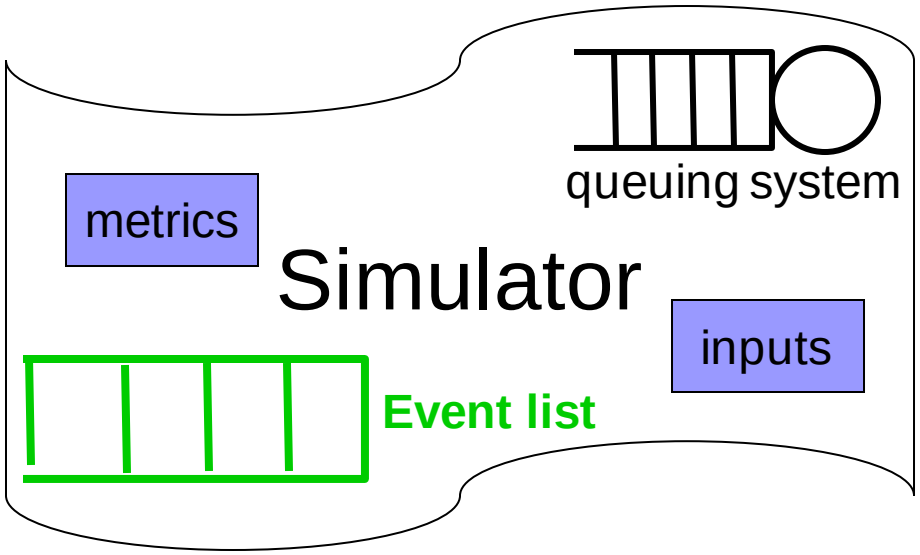
- Initialize System State
- Initialize Event List
- while (events to process) {
  - remove next event from event list
  - advance simulation clock
  - process event
    - change system state, schedule new events (add them to event list)
    - collect any metrics
- }

Turning this into a program

- Identify entities in the system
- Turn them into classes

Identify entities in the queue simulator

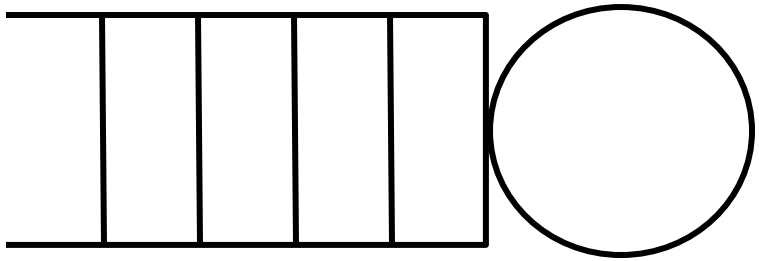




- Queuing system
  - Server
  - Buffer
    - Requests
- Simulator
  - Queueing System (State)
  - Eventlist
    - Events
  - Input parameters
- Metrics?
  - class by itself?
  - associated with entities?

Class design of entities

(my design, just an example, not perfect)

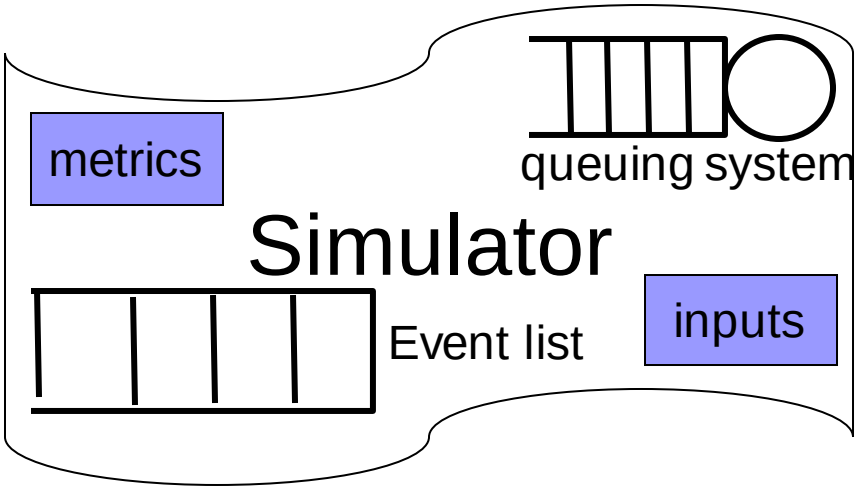


- Server
  - ID
  - busy/idle
  - some metrics
  - pointer to Request in service
- Operations:
  - get/set request in service
  - isBusy

- Request
  - ID
  - arrivalTime
  - serviceTime
  - serverAssigned (not used currently)

- Queuing system
  - Server
  - Buffer
    - queue of requests
- Operations
  - enqueue
  - dequeue
  - nextReq
  - isBusy
  - setAvg

Class design of entities



- Simulation
  - eventList (priority queue)
  - queueing system
  - some metrics
  - structures to hold trace data