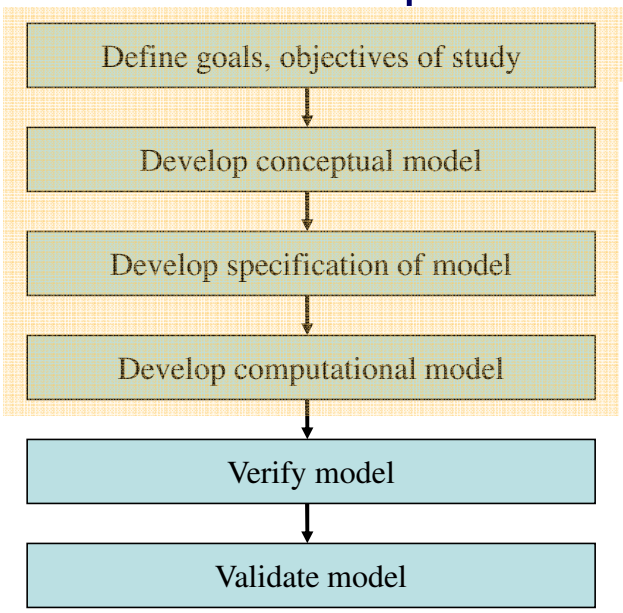
Model Development Life Cycle



Focus on these steps

Fundamentally an iterative process

Determine Goals and Objectives

- What do you hope to accomplish with the model
  - May be an end in itself
    - Predict the weather
    - Train persons to develop certain skills (e.g., driving)
  - More often a means to an end
    - Optimize a manufacturing process or develop the most cost effective means to reduce traffic congestion in some part of a city
- Goals may not be known when you start the project!
  - One often learns things along the way

Develop Conceptual Model

- An abstract (i.e., not directly executable) representation of the system
- What should be included in model? What can be left out?
- What abstractions should be used
  - Level of detail
  - Often a variation on standard abstractions
- What metrics will be produced by the model
- Appropriate choice depends on the purpose of the model

Develop Specification Model

- A more detailed specification of the model including more specifics
- Collect data to populate model
  - Traffic example: Road geometry, signal timing, expected traffic demand, driver behavior
- Development of algorithms necessary to include in the model

Develop Computational Model

- Executable simulation model
- Software approach
  - General purpose programming language
  - Special purpose simulation language
  - Simulation package
  - Approach often depends on need for customization and economics

Queueing Network Applications

Queueing networks useful for many applications

- Customers utilizing business services (e.g., bank, hospital, restaurant …)
- Manufacturing: assembly lines
- Supply chains
- Transportation (aircraft, vehicles)
- Computer communication networks
- Computer systems (jobs being processes by a set of compute servers; I/O systems)

An Example: Single Server Queue

Outline

- Problem description
- Conceptual model: queuing networks
- Specification model
- Time stepped(Fixed time) implementation

Problem Description

A certain airport contains a single runway on which arriving aircraft must land. Once an aircraft is cleared to land, it will use the runway, during which time no other aircraft can be cleared to land. Once the aircraft has landed, the runway is available for use by other aircraft. The landed aircraft remains on the ground for a certain period of time before departing.

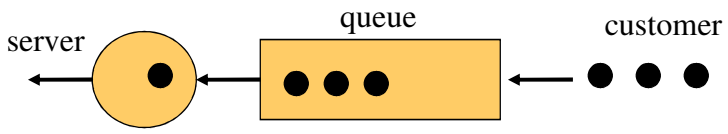
The objective is to determine

- The average time an aircraft must wait when arriving at an airport before they are cleared to land
- The maximum number of aircrafts that will be on the ground at one time

Problem Description

- The output metrics suggest focusing on
  - Waiting process
  - Number of aircraft on the ground
- We could develop a detailed model keeping track of the position of each aircraft every second, but this may be too much.
- Queuing models are a natural abstraction for modeling systems like these that include
  - Customers competing to use limited resources
  - Waiting (queuing) to use the resource
  - Primary metrics of interest have to do with resource utilization, time customer is being served or waiting
  - Details of what customer is doing while waiting are not important

Conceptual Model



- Customer (aircraft)
  - Entities utilizing the system/resources
- Server (runway)
  - Resource that is serially reused; serves one customer at a time
- Queue
  - Buffer holding aircraft waiting to land

Specification Model

◆ **Customers**

Schedule of aircraft arrivals:

Often, probability distribution defines time between successive customer arrivals (interarrival time)

Assumes interarrival times *independent, and identically distributed*(iid)

Customer attributes? e.g., priorities

◆ **Servers**

How much *service time* is needed for each customer?

May use probability distribution to specify customer service time (iid)

How many servers?

◆ **Queue**

Service discipline - who gets service next?

- First-in-first-out (FIFO), Last-in-first-out (LIFO), random …
- May depend on a property of the customer (e.g., priority, "smallest" first)

Specification Model

Assume

- Customers
  - Assume arrivals are iid, following an exponential distribution for interarrival times with mean μ
  - Assume all customers are identical (no specific attributes)
- Servers
  - Assume customer service time is iid, exponentially distributed, mean λ
  - Assume one server (one runway)
- Queue
  - Assume first-in-first-out queue (first-come-first-serve) discipline
  - Assume queue has unlimited capacity

Computational Model

A computer simulation is a computer program that models the behavior of a physical system over time. To do this, we must

- Define a computer representation of the state of the system, i.e., define state variables that encode the current state of the physical system
- Determine the state of the system over all points in time in which we are interested (compute a *sample path*)
  - Define a simulation program that modifies state variables to model the evolution of the physical system over time.

Key questions

- What are the state variables?
- How does the state change (what rules are used)?

Discrete Event Simulation

Discrete event simulation: computer model for a system where changes in the state of the system occur at *discrete* points in simulation time.

Fundamental concepts:

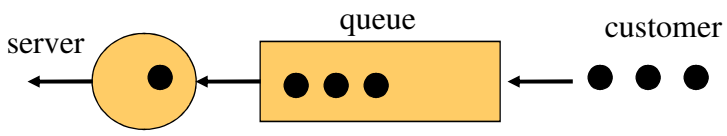
- system state (state variables)
- state transitions (events)
  - each event has a timestamp indicating when it occurs

An DES computation can be viewed as a sequence of event computations, with each event computation is assigned a (simulation time) time stamp

Each event computation can

- modify state variables
- schedule new events

State Variables (back to the ex.)



State:

- InTheAir: number of aircraft either landing or waiting to land
- OnTheGround: number of landed aircraft
- RunwayFree: Boolean, true if runway available

Events

- An event must be associated with any change in the state of the system
- Airport example:
  - Event for an aircraft arrival (`InTheAir, RunwayFree`)
  - Event for aircraft landing (`InTheAir, OnTheGround, RunwayFree`)
  - Event for aircraft departure (`OnTheGround`)

Evolving System State

- Given the current state of the system, how do we determine the new system state?
- At which points in time (in the simulated system) we need to compute the system state: Two Approaches
  - Fixed increment time advance (time stepped simulation)
  - Next event time advance (Irregular) (typically, when the state changes)

Fixed Increment Time

```
/* ignore aircraft departures */
Float InTheAir: # aircraft landing or waiting to land
Float OnTheGround: # landed aircraft
Boolean RunwayFree: True if runway available
Float NextArrivalTime: Time the next aircraft arrives
Float NextLanding: Time next aircraft lands (if one is landing)

For (Now = 1 to EndTime) { /* time step size is 1.0 */
   if (Now >= NextArrivalTime) /* if aircraft just arrived */
   {  InTheAir := InTheAir + 1;
      NextArrivalTime := NextArrivalTime + RandExp( μ);
      if (RunwayFree)
      { RunwayFree := False;
        NextLanding := Now + RandExp( λ);
      }
   }
   if (Now >= NextLanding) /* if aircraft just landed */
    { InTheAir := InTheAir - 1;
      OnTheGround := OnTheGround + 1;
      if (InTheAir > 0) NextLanding := Now + RandExp(λ)
      else {RunWayFree := True; NextLanding := EndTime+1;}
   }
```

Problems With Fixed increment time advance

- State changes may occur between time steps
  - Use small time steps to minimize error
- Multiple state changes within the same time step may be processed in the wrong order
  - Solvable by ordering state changes within time step (more work)
- Inefficient
  - Many time steps no state changes occur, especially if small time steps

The Example: Air traffic…

Single runway for incoming aircraft, ignore departure queueing

- `L` = mean time runway used for each landing aircraft (exponential distrib.)
- `G` = mean time on the ground before departing (exponential distribution)
- `A` = mean interarrival time of incoming aircraft (exponential distribution)

State:

- `Now:` current simulation time
- `InTheAir`: number of aircraft landing or waiting to land
- `OnTheGround`: number of landed aircraft
- `RunwayFree`: Boolean, true if runway available

Events:

- `Arrival`: denotes aircraft arriving in air space of airport
- `Landed`: denotes aircraft landing
- `Departure`: denotes aircraft leaving

Landing Event

An aircraft has completed its landing.

- `L` = mean time runway is used for each landing aircraft
- `G` = mean time required on the ground before departing
- `Now:` current simulation time
- `InTheAir`: number of aircraft landing or waiting to land
- `OnTheGround`: number of landed aircraft
- `RunwayFree`: Boolean, true if runway available

```
Landed Event:
InTheAir:=InTheAir-1;
OnTheGround:=OnTheGround+1;
Schedule Departure event @ Now + RandExp(G);
If (InTheAir>0)
   Schedule Landed event @ Now + RandExp(L);
Else
   RunwayFree := TRUE;
```

Arrival Event

New aircraft arrives at airport. If the runway is free, it will begin to land. Otherwise, the aircraft must circle, and wait to land.

- `A:` mean interarrival time of incoming aircraft
- `Now:` current simulation time
- `InTheAir`: number of aircraft landing or waiting to land
- `OnTheGround`: number of landed aircraft
- `RunwayFree`: Boolean, true if runway available

```
Arrival Event:
InTheAir := InTheAir+1;
Schedule Arrival event @ Now + RandExp(A);
If (RunwayFree) {
  RunwayFree:=FALSE;
  Schedule Landed event @ Now + RandExp(L);
}
```

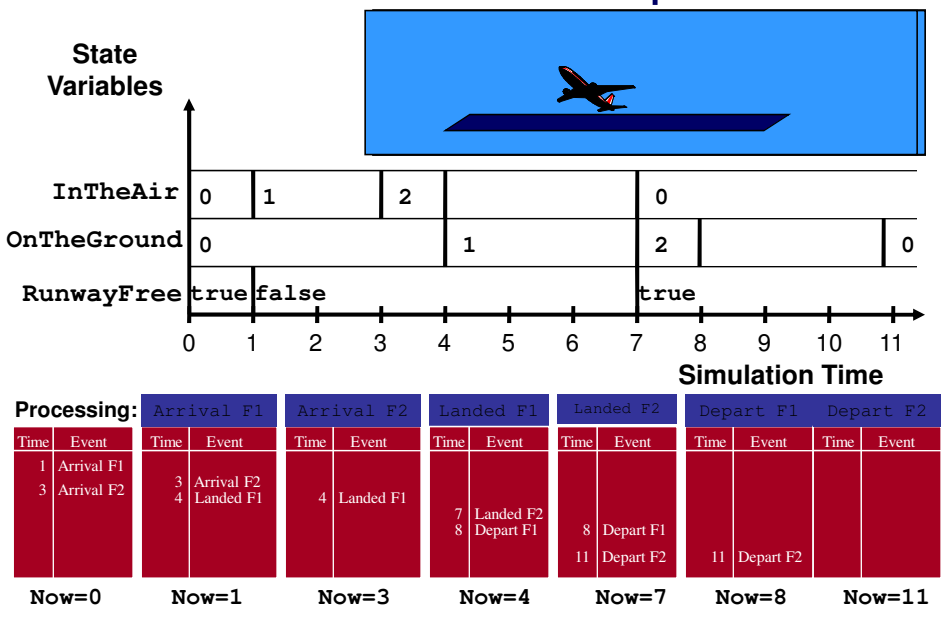
Departure Event

An aircraft now on the ground departs for a new destination.

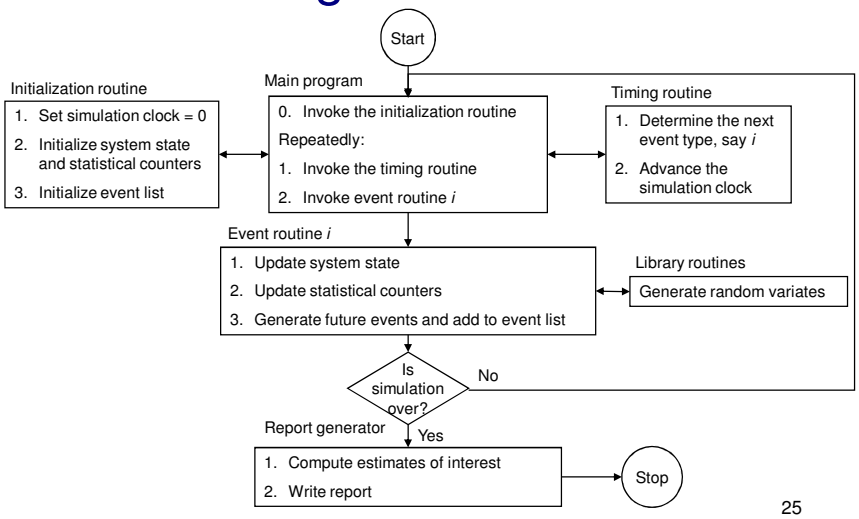
- `Now:` current simulation time
- `InTheAir`: number of aircraft landing or waiting to land
- `OnTheGround`: number of landed aircraft
- `RunwayFree`: Boolean, true if runway available

```
Departure Event:
OnTheGround := OnTheGround - 1;
```

Execution Example



Discrete-Event Simulation Logical Flow



Discrete-Event Simulation Components

Initialization routine: A subprogram to initialize the simulation model at time 0

Timing routine: A subprogram that determines the next event from the event list and then advances the simulation clock to the time when the next event is to occur

Event routine: A subprogram that updates the system state when a particular type of event occurs (there is one event routing for each type of event)

Library routines: A set of subprograms used to generate **random observations from probability distributions** that were determined as part of the simulation model

Discrete-Event Simulation Components

Report generator: A subprogram that computes estimates (from the statistical counters) of the desired measures of performance and produces a report when the simulation ends

Main program: A subprogram that invokes the timing routing to determine the next event and then transfers control to the corresponding event routine to update the system state appropriately. The main program may also check for termination and invoke the report generator when the simulation is over.

Discrete-Event Simulation Stopping Rules

Number of events of a certain type reached a pre-defined value

- Example: stop M/M/1 simulation after the 1000th departure

Simulation time reaches a certain value; this is usually implemented by scheduling an "end-simulation" event at the desired simulation stop time.

Output Statistics

Compute

- The maximum number of aircraft that will be on the ground at one time
- Average time an aircraft must wait before they are cleared to land

- Maximum on ground
  - Variable for airport indicating number currently on ground
  - Maximum “on the ground” so far
- Wait time
  - Variables for airport indicating total wait time, number of aircraft arrivals
  - State variable for each aircraft indicating arrival time

Probability

- Probability is a measure of how likely it is for an event to happen.
- We name a probability with a number from 0 to 1.
- If an event is certain to happen, then the probability of the event is 1.
- If an event is certain not to happen, then the probability of the event is 0.

Probability Vs Statistics

- In probability theory: R.V. is specified and their parameters are known.
- Goal: Compute probabilities of random values that these variables can take.
- In statistics: The values of random variables are known " from experiment" but theoretical characteristics are unknown.
- Goal: To determine the unknown theoretical characteristics of R.V.
- Probability and Statistics are complementary subjects

What is an Event?

- In probability theory, **an event** is a set of outcomes (a subset of the sample space) to which a probability is assigned.
- Typically, when the sample space is finite, any subset of the sample space is an event (i.e. all elements of the power set of the sample space are defined as events).

Fundamentals

- We measure the probability for Random Events
  - How likely an event would occur
- The set of all possible events is called Sample Space
- In each experiment, an event may occur with a certain probability (Probability Measure)
- Example:
  - Tossing a dice with 6 faces
  - The sample space is {1, 2, 3, 4, 5, 6}
  - Getting the Event « 2 » in on experiment has a probability 1/6

Examples

- A single card is pulled (out of 52 cards).
  - **Possible Events**
    - having a red card (P=1/2);
    - Having a Jack (P= 1/13);
- Two true 6-sided dice are used to consider the event where the sum of the up faces is 10.
  - P = 3 / 36 = 1/12

Probability

- The **probability** of every set of possible events is between 0 and 1, inclusive.
- The probability of the whole set of outcomes is 1.
  - Sum of all probability is equal to one
  - Example for a dice: P(1)+P(2)+P(3)+ P(4)+P(5)+P(6)=1
- If A and B are two events with no common outcomes, then the probability of their union is the sum of their probabilities.
  - Event E1={1},
  - Event E2 ={6}
  - P(E1 v E2)=P(E1)+P(E2)

Random Variables

An Experiment: is a process whose outcome is not known with certainty

Sample Space: set of outcomes S

Ex: S= {H,T} , S= {1,2,3,4,5,6}

Random Variable: also known as **stochastic variable**. is a function that assigns a real number to each point in the space

Random Variable is either discrete or continuous

A random variable: Examples.

- The waiting time of a customer in a queue
- The number of cars that enters the parking each hour
- The number of students that succeed in the exam

Probability Distribution

- The **probability distribution** of a discrete random variable is a **list of probabilities** associated with each of its possible values.
- It is also sometimes called the **probability function** or the **probability mass function** (**PMF**) for discrete random variable.

Probability Mass Function (PMF)

- the **probability distribution** or **probability mass function (PMF)** of a **discrete random variable** X is a function that gives the probability p(xi) that the random variable equals some value xi, for each value xi:
- It satisfies the following conditions:

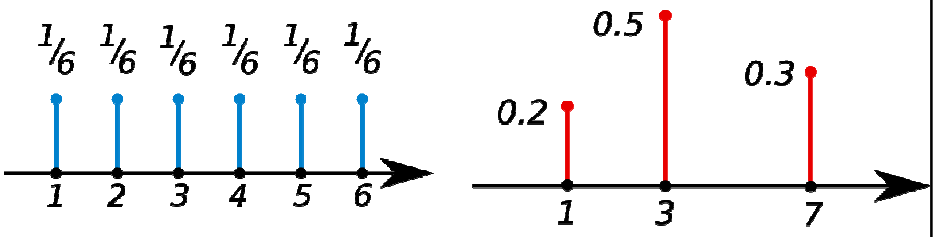
$$p(x_i) = P(X = x_i)$$

$$0 \le p(x_i) \le 1$$

$$\sum_i p(x_i) = 1$$

Probability Mass Function

PMF of a fair Dice



Continuous Random Variable

- A **continuous random variable** is one which takes an infinite number of possible values.
- Continuous random variables are usually measurements.
- Examples include height, weight, the amount of sugar in an orange, the time required to run a mile.

Distribution function *aggregates*

- For the **case of continuous variables**, we do not want to ask what the probability of "1/6" is, because the answer is always 0...
- Rather, we ask **what is the probability that the value is in the interval (a,b)**.
- So for continuous variables, we care about the derivative of the distribution function at a point (that's the derivative of an integral). This is called a **probability density function (PDF)**.
- The probability that a random variable has a value in a set A is the integral of the p.d.f. over that set A.

Probability Density Function (PDF)

- The **Probability Density Function (PDF)** of a **continuous random variable** is a function that can be integrated to obtain the probability that the random variable takes a value in a given interval.
- More formally, the probability density function, $f(x)$, of a **continuous random variable** X is the derivative of the cumulative distribution function $F(x)$:

$$f(x) = \frac{d}{dx} F(x)$$

- Since $F(x)=P(X \leq x)$, it follows that:

$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f(x) \cdot dx$$

Cumulative Distribution Function (CDF)

- The **Cumulative Distribution Function (CDF)** is a function giving the probability that the random variable X is less than or equal to x, for every value x.
- Formally
  - the cumulative distribution function F(x) is defined to be: $\forall -\infty < x < +\infty,$

$$F(x) = P(X \leq x)$$

Cumulative Distribution Function (CDF)

- For a **discrete random variable**, the cumulative distribution function is found by summing up the probabilities as in the example below.

$$\forall -\infty < x < +\infty,$$

$$F(x) = P(X \le x) = \sum_{x_i \le x} P(X = x_i) = \sum_{x_i \le x} p(x_i)$$

- For a **continuous random variable**, the cumulative distribution function is the integral of its **probability density function f(x)**.

$$F(a) - F(b) = P(a \le X \le b) = \int_a^b f(x) \cdot dx$$

Cumulative Distribution Function (CDF)

- **EX- Discrete case**: Suppose a random variable X has the following probability mass function p(xi):

xi	0	1	2	3	4	5
$p(xi)$	1/32	5/32	10/32	10/32	5/32	1/32

- The cumulative distribution function F(x) is then:

xi	0	1	2	3	4	5
$F(xi)$	1/32	6/32	16/32	26/32	31/32	32/32

Discrete Distribution Function

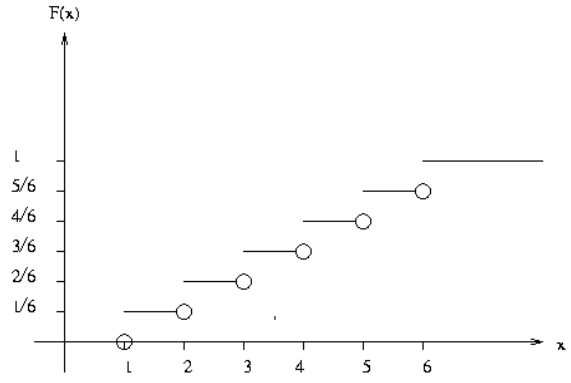


Figure 1.4: Fair die: Graph of the distribution function.

Mean or Expected Value

Expectation of discrete random variable X

$$\mu_X = E(X) = \sum_{i=1}^{n} x_i \cdot p(x_i)$$

Expectation of continuous random variable X

$$\mu_X = E(X) = \int_{-\infty}^{+\infty} x \cdot f(x) dx$$

Example: Mean and variance

- When a die is thrown, each of the possible faces 1, 2, 3, 4, 5, 6 (the xi's) has a probability of 1/6 (the p(xi)'s) of showing. The expected value of the face showing is therefore:

$$\mu = E(X) = (1 \times 1/6) + (2 \times 1/6) + (3 \times 1/6) + (4 \times 1/6) + (5 \times 1/6) + (6 \times 1/6) = 3.5$$

- Notice that, in this case, E(X) is 3.5, which is not a possible value of X.

Variance

- The **variance** is a measure of the 'spread' of a distribution about its average value.
- Variance is symbolized by $V(X)$ or $Var(X)$ or σ^2.
  - The **mean** is a way to describe the location of a distribution,
  - the **variance** is a way to capture its scale or degree of being spread out. The unit of variance is the square of the unit of the original variable.

Variance

- The Variance of the random variable X is defined as:

$$V(X) = \sigma_X^2 = E\left(X - E(X)\right)^2 = E\left(X^2\right) - E(X)^2$$

  - where E(X) is the expected value of the random variable X.
- The **standard deviation** is defined as the square root of the variance, i.e.:

$$\sigma_X = \sqrt{\sigma_X^2} = \sqrt{V(X)} = s$$

Coefficient of Variation

- The Coefficient of Variance of the random variable X is defined as:

$$CV(X) = \frac{V(X)}{E(X)} = \frac{\sigma_X}{\mu_X}$$

Bernoulli Trials

Any simple trial with two possible outcomes.

EX: Tossing a coin, repeat, with counting # of success *p*
" the number of heads"

Then # of failure *q* (1-*p)* , *" the number of tails"*

P(HHT) =p.p.q

P(TTT)=q.q.q