Simulation & Modeling

Event-Oriented Simulations

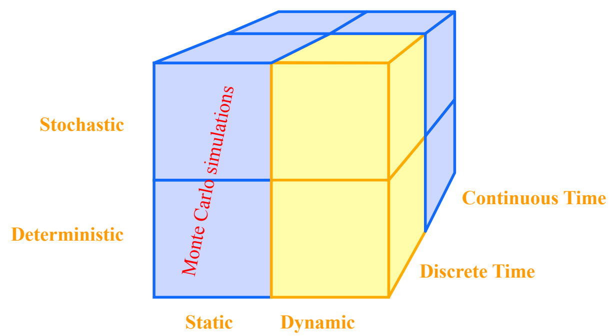
Outline

- Simulation modeling characteristics
- Concept of Time
- A DES Simulation (Computation)
- DES System = model + simulation execution
- Data Structures
- Program (Code)

Basic concepts

- Simulation modeling characteristics
- The Concept of Time
- Static or dynamic models
- Stochastic, deterministic or chaotic models
- Discrete or continuous change/models
- Aggregates or Individuals

Modeling Classification



Monte Carlo Simulation / Methods

Probability Simulations – simulates outcomes by generating 'random data' to understand the risks and uncertainties (the likelihood of possible outcomes).

- **Generate Inputs randomly from a probability distribution.**
- **Perform a deterministic computation on the input (repeat this step).**
- **Aggregate result (run multiple times with a different sample on the input) to 'approximate' the 'real' value and result.**

http://en.wikipedia.org/wiki/Monte_Carlo_method

Example Monte Carlo

- **History: Rolling a die, and determine the probability of outcomes.**
  - https://www.goldsim.com/web/introduction/probabilistic/montecarlo/
  - **Gaming/Casino: Roulette**
- **Related to what we done: Finance Models.**
  - **Models of stock prices, option pricing (Black Sholes Formula).**
  - **Risk Analysis**
  - **Estimate portfolio values**
- **In the Mainstream: Pokemon Go Battles**
  - https://www.pokebattler.com/

Static or dynamic models

- **Dynamic:**
  - **State variables change over time**
  - **System Dynamics, Discrete Event, Agent-Based**
- **Static:**
  - **Snapshot(s) at a single point in time**
  - **Monte Carlo simulation (large number of input samples, compute & aggregate results, time doesn't change), optimization models**

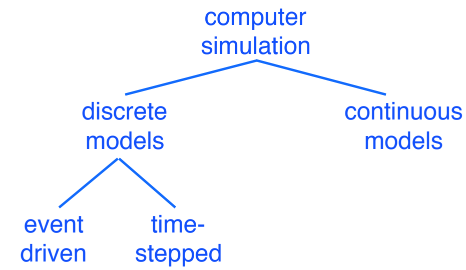
Deterministic, Stochastic or Chaotic

- **Deterministic:**
  - **Predictive behavior. The system is perfectly understood, then it is possible to predict precisely what will happen.**
  - **Repeatable**
- **Stochastic:**
  - **behavior cannot be entirely predicted.**
- **Chaotic:**
  - **deterministic model with a behavior that cannot be entirely predicted. Depends so sensitively on the system's initial conditions so that in effect it cannot be predicted.**

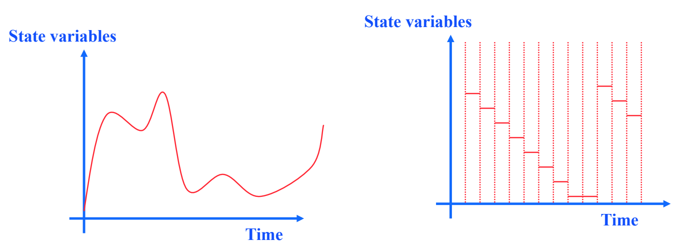
Discrete or Continuous models

- **Discrete model:**
  - » **state variables change only at a countable number of points in time.**
  - » **These points in time are the ones at which the event occurs/change in state.**
- **Continuous model:**
  - » **the state variables change in a continuous way, and not abruptly from one state to another.**
  - » **infinite number of states.**

Recap: Simulation *Taxonomy*

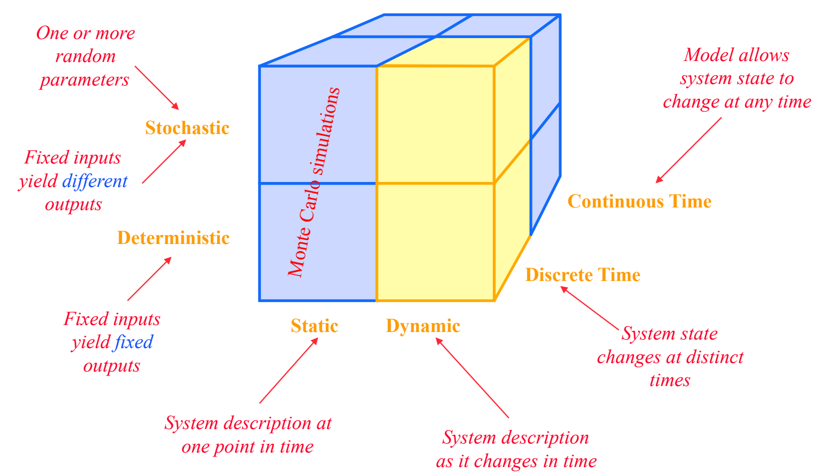


- **Continuous time simulation**
  - » **State changes occur continuously across time**
  - » **Typically, behavior described by differential equations**
- **Discrete time simulation**
  - » **State changes only occur at discrete time instants**
  - » **Time stepped: time advances by fixed time increments**
  - » **Event stepped: time advances occur with irregular increments**

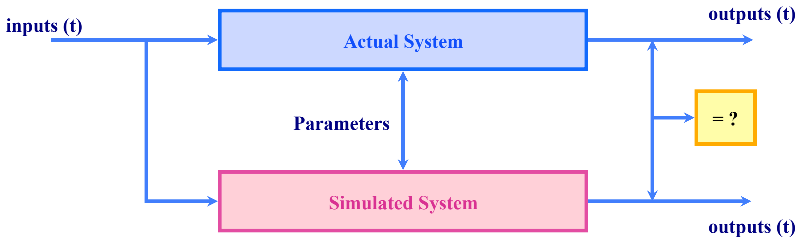


Continuous: State variables change continuously as a function of time
State variables = *f(t)*

Discrete: State variables change at discrete times

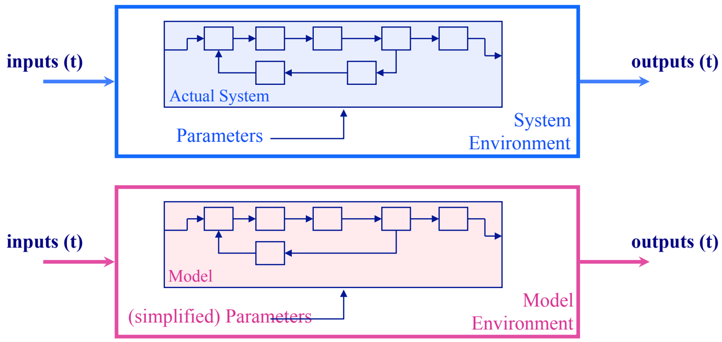


Simulation



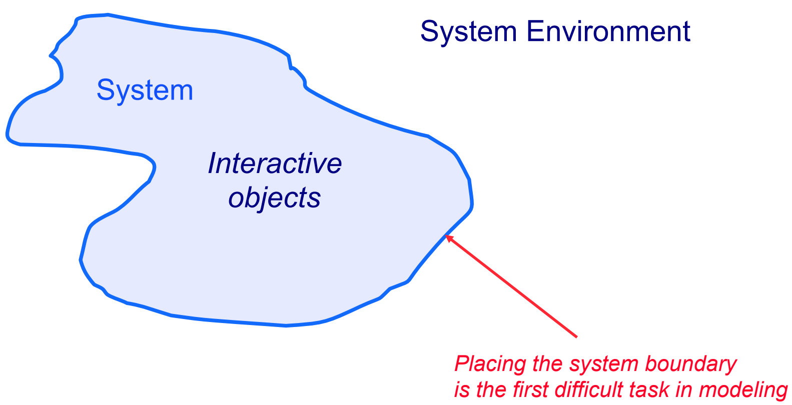
- Simulated system imitates operations of actual system over time
- Artificial history of system can be generated and observed
- Internal (perhaps unobservable) behavior of system can be studied
- Time scale can be altered as needed
- Conclusion about actual system characteristics can be inferred

What is a simulation model?

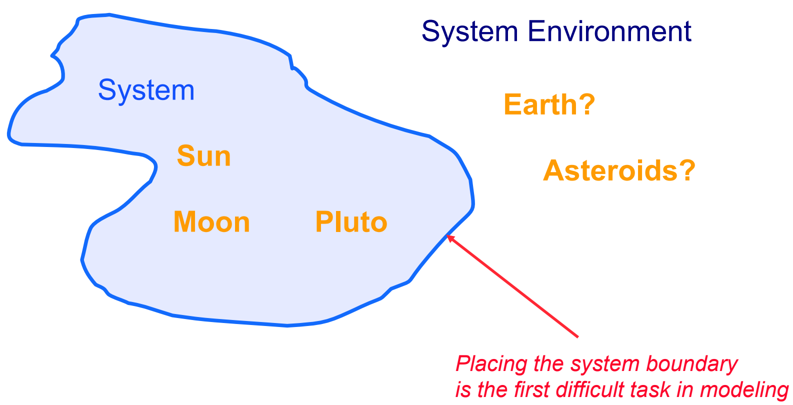


- An abstraction of a real system
- Simplified assumptions are used to capture (only) important behaviors

System's Modeling



System's Modeling

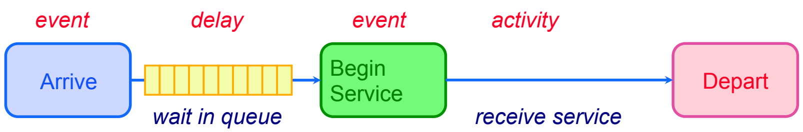


Entities, Attributes and Activities...

- An **entity** is an object of interest in the system
  - » Example: Customer Manager Cashier
- An attribute is a (relevant) property of an entity
  - » Example: Account balance Gender Skills
- Attributes are state variables

Activities & Delays

- An activity… … is a duration of a known (expected) length
  - » Example: drink coffee, serve customers
  - » Activities form part of the model specification
  - » Inter-arrival time, service time
  - » Deterministic or stochastic (probabilistic)
- A delay… … is a duration of unknown length
  - » waiting time in queue
- Delays form part of the simulation results
  - » Example: waiting time in queue
  - » Delays form part of the simulation results



State and State Variables

- The (system) state
  - » complete
  - » minimal
  - » contains sufficient information to describe the system at any point in time.
- A state variable
  - » Describes a portion of the state.
  - » Length of a queue, activity of a manager (sleeping, drinking coffee)

Events

- Event:
  - » Occurrence
  - » Instantaneous
  - » May change the state
- Example single server queue:
  - » Arrival -- while the server is busy, so queue length is incremented by 1;
  - » Departure -- the completion of service

Conditional and Primary Events

- Primary Events
  - » Scheduled at a certain time
  - » Arrival of customers
- Conditional Events
  - » triggered by a certain condition becoming TRUE -- a completion of a delay
  - » Customers moving from queue to service

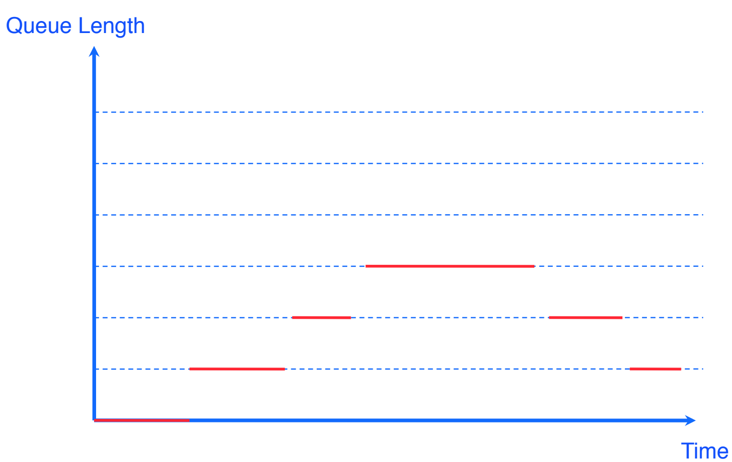
How to create a DES?

- DES Modeling raises the following questions?
  - » How does each event affect system state and attributes?
  - » How are activities defined?
    - – What events mark beginning and the end?
    - – What condition (if any) most hold?
  - » How are delays defined?
  - » How is the simulation initialized?

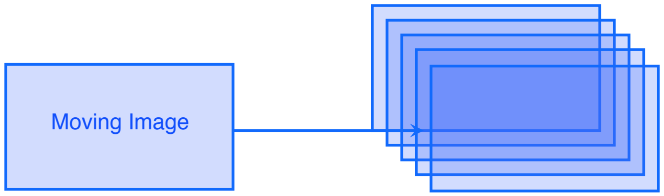
A Simulation classic

- Single-server Queue at a bank
- One possible problem formulation:
  - » "customer have to wait too long in my bank"
- Objective:
  - » Determine the effect of an additional cashier
- Data needed:
  - » inter-arrival time of customers
  - » Service times

Simulation Results

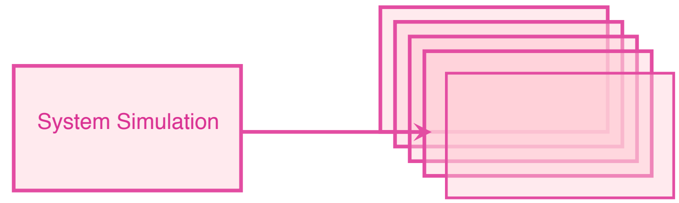


Movie



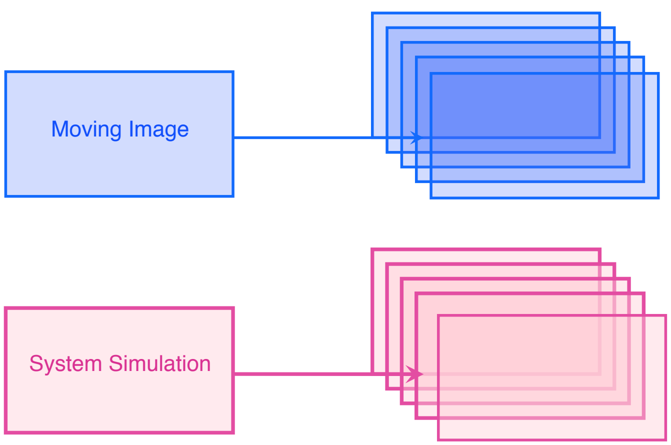
- Series of still images, sufficient to convey recognizable motion

System Snapshots



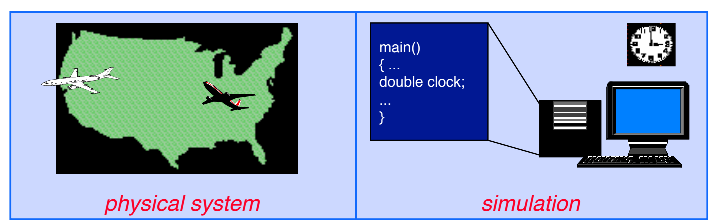
- Series of system snapshot
  - » system state
  - » activities in progress
  - » end time

System Snapshots



Time

- ***Physical system:*** actual or imagined system being modeled
- ***Simulation***: a system that emulates the behavior of a physical system



- **physical time**: time in the physical system
  - » Noon, December 31, 1999 to noon January 1, 2000
- **simulation time**: representation of physical time within the simulation
  - » floating point values in interval [0.0, 24.0]
- **wallclock time**: time during the execution of the simulation, usually output from a hardware clock
  - » 9:00 to 9:15 AM on September 10, 1999

Simulation Time

Simulation time *is defined as a totally ordered set of values where each value represents an instant of time in the physical system being modeled.*

- **For any two values of simulation time T_1 representing instant P_1, and T_2 representing P_2:**
- **Correct ordering of time instants**
  - » **If $T_1 < T_2$, then P_1 occurs before P_2**
  - » **9.0 represents 9 PM, 10.5 represents 10:30 PM**
- **Correct representation of time durations**
  - » **$T_2 - T_1 = k (P_2 - P_1)$ for some constant k**
  - » **1.0 in simulation time represents 1 hour of physical time**

Modes of Execution

- **As-fast-as-possible execution (unpaced):** no fixed relationship necessarily exists between advances in simulation time and advances in wallclock time
- **Real-time execution (paced):** each advance in simulation time is paced to occur in synchrony with an equivalent advance in wallclock time
- **Scaled real-time execution (paced):** each advance in simulation time is paced to occur in synchrony with S * an equivalent advance in wallclock time (e.g., 2 x wallclock time)

Converting from wallclock to Simulation Time:

Simulation Time = W2S(W) = $T_0 + S * (W - W_0)$

W = wallclock time; S = scale factor

W_0 (T_0) = wallclock (simulation) time at start of simulation

(assume simulation and wallclock time use same time units)

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)**

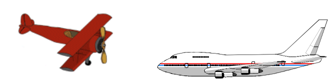
A DES computation: can be viewed as a *sequence of event computations*, with each event computation is assigned a (simulation time) time stamp. Each event computation can

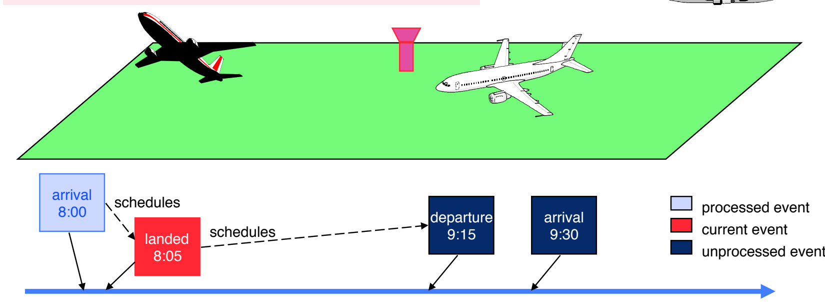
- **modify state variables**
- **schedule new events**

Discrete Event Simulation Computation

example: air traffic at an airport

events: aircraft arrival, landing, departure





- **Unprocessed events are stored in a pending event list**
- **Events are processed in time stamp order**

Discrete Event Simulation System

model of the physical system

Simulation Application
- state variables
- code modeling system behavior
- I/O and user interface software

calls to schedule events

calls to event handlers

Independent of the simulation application

Simulation Executive
- event list management
- managing advances in simulation time

Event-Oriented World View

Simulation application

state variables

```
Integer: InTheAir;
Integer: OnTheGround;
Boolean: RunwayFree;
```

Event handler procedures

```
Arrival Event    Landed Event    Departure Event
{                {               {
  …                …               …
}                }               }
```

Simulation executive

Now = 8:45

Pending Event List (PEL)

9:00 9:16 10:10

Event processing loop

```
while(simulation not finished)
  E = smallest time stamp event in PEL
  Remove E from PEL
  Now := time stamp of E
  call event handler procedure
```


Example: Air traffic at an Airport

Model aircraft arrivals and departures, arrival queuing

Single runway for incoming aircraft, ignore departure queuing

- R = time runway is used for each landing aircraft (constant)
- G = time required on the ground before departing (constant)

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

Arrival Events

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

- R = time runway is used for each landing aircraft
- G = 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

```
Arrival Event:
    InTheAir := InTheAir+1;
    if( RunwayFree )
      RunwayFree:=FALSE;
      Schedule Landed event @ Now + R;
```

Landed Event

An aircraft has completed its landing.

R = time runway is used for each landing aircraft
G = 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 + G;
    if( InTheAir > 0 )
      Schedule Landed event @ Now + R;
    else
      RunwayFree := True;
```

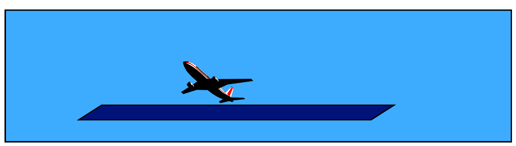

Departure Event

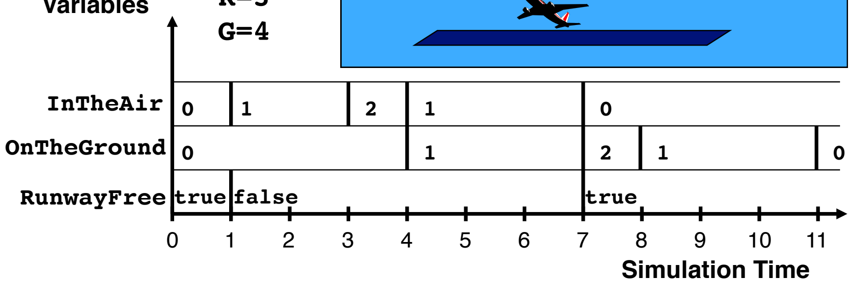
An aircraft on the ground departs for a new destination.

R = time runway is used for each landing aircraft
G = 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

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


Execution Example



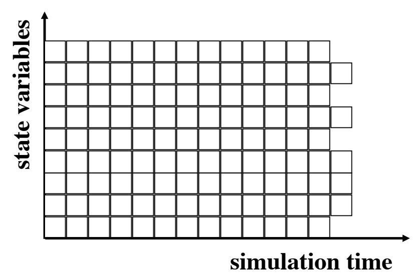


Processing:

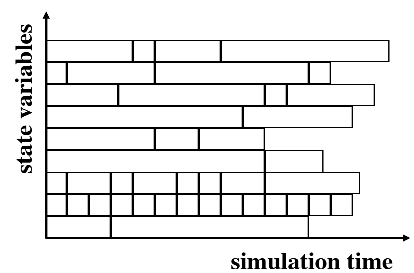
	Now=0	Arrival F1 (Now=1)	Arrival F2 (Now=3)	Land F1 (Now=4)	Land F2 (Now=7)	Depart F1 (Now=8)	Depart F2 (Now=11)
Time / Event	1 Arrival F1	3 Arrival F2	4 Land F1	7 Land F2	8 Depart F1	11 Depart F2	
Time / Event	3 Arrival F2	4 Land F1	7 Land F2	8 Depart F1	11 Depart F2		

Perspective: Time Stepped vs. Event Stepped

Goal: compute state of system over simulation time



time stepped execution



event driven execution

Summary

- Time
  - » Important to distinguish among simulation time, wallclock time, and time in the physical system
  - » Paced execution (e.g., immersive virtual environments) vs. unpaced execution (e.g., simulations to analyze systems)
- DES computation: sequence of event computations
  - » Modify state variables
  - » Schedule new events
- DES System = model + simulation executive
- Data structures
  - » Pending event list to hold unprocessed events
  - » State variables
  - » Simulation time clock variable
- Program (Code)
  - » Main event processing loop
  - » Event procedures
  - » Events processed in time stamp order