Discrete Event Simulation

Xiangyang Zhang
TRLab, Queen's University

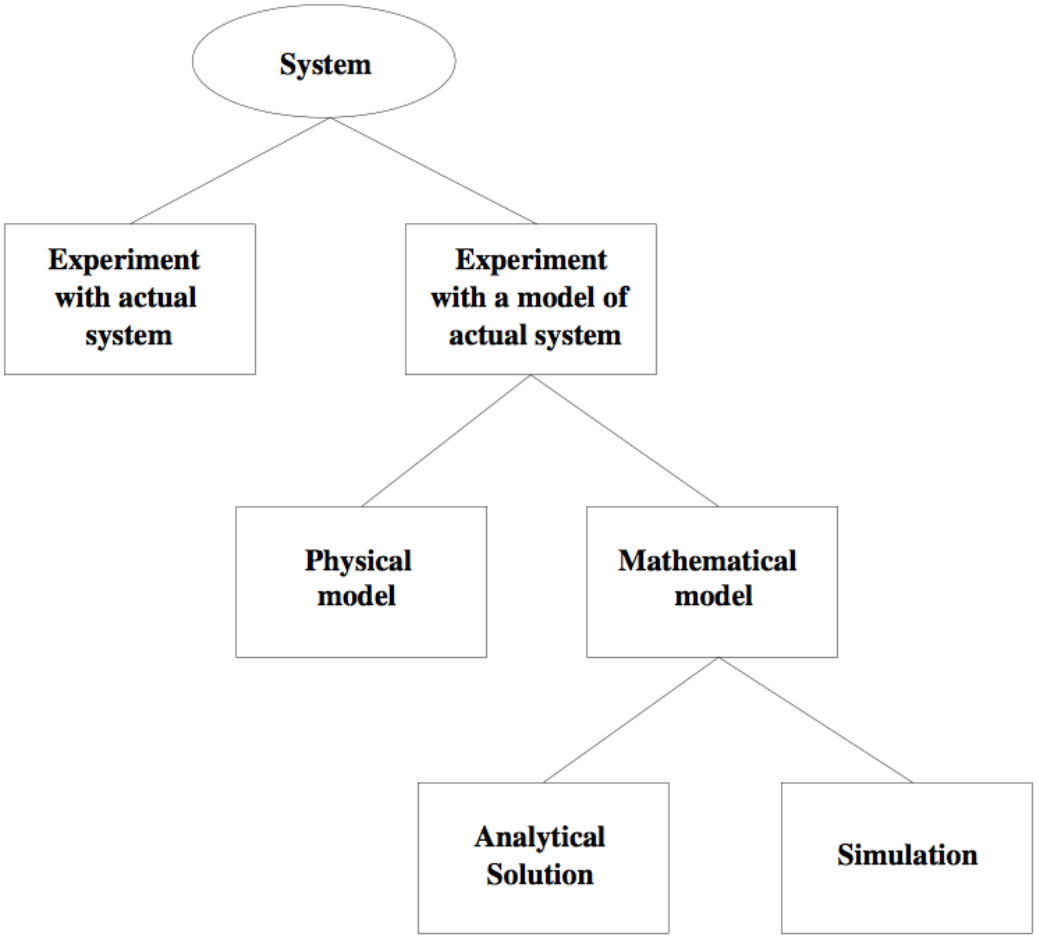
CONTENTS

- Theory and Concepts
  - Banks et al., Discrete Event System Simulation, Prentice Hall, 2001
  - White and Ingalls, Introduction to Simulation, Proc. Winter Simulation Conf., 2009
- Practice
  - Various online resources and ns2

System and Model

- A system is a group of objects that are joined together in some regular interaction toward the accomplishment of some purpose
- A model is a representation of a real system for the purpose of studying the system
  - Models are simplified abstraction,
  - But contain enough details for the study objectives

Ways To Study A System



Simulation, Modeling & Analysis (3/e) by Law and Kelton, 2000, p. 4, Figure 1.1

Simulation

- A simulation is the imitation of the operation of a real-world process or system over time.
  - Analytical methods employ the deductive reasoning of math
  - Simulation is a numerical method
- Application areas of simulation
  - You name it !

What Is a System?

- System, environment and the boundary
  - An Input is an action of the environment on the system that causes changes in the system state.
- System and components
  - Entities and their attributes
  - State and state variables
  - Activity and event
- System outputs
  - Measured quantities which we want to known

Components in Details

- System state: a collection of variables necessary to describe the system at any time
- An entity is an object of interest in the system
  - A dynamic entity “moves” through the system
  - A static entity serves other entities
  - A resource serves dynamic entities and has a constrained capacity
- An attribute is a property of an entity
  - Should be considered as local values

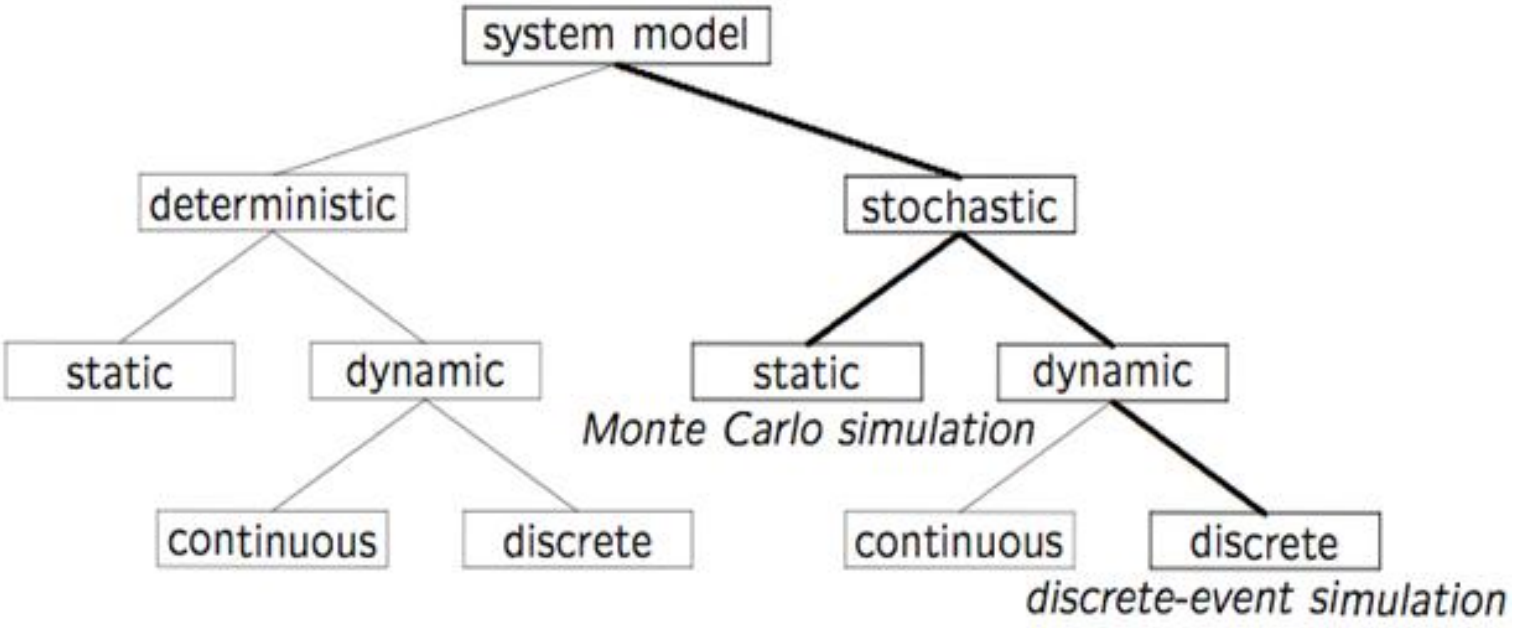
Components in Details, Cont'd

- An activity represents a time period of specified length
  - Delay activity: end time known when begin
  - Queue activity: end time unknown when begin
  - Logic activity: instantaneous
- An event is an instantaneous occurrence that may change the state of the system
- Activities and events can be internal or external
  - internal: occurs within the system
  - external: occurs in the environment

Types of Models

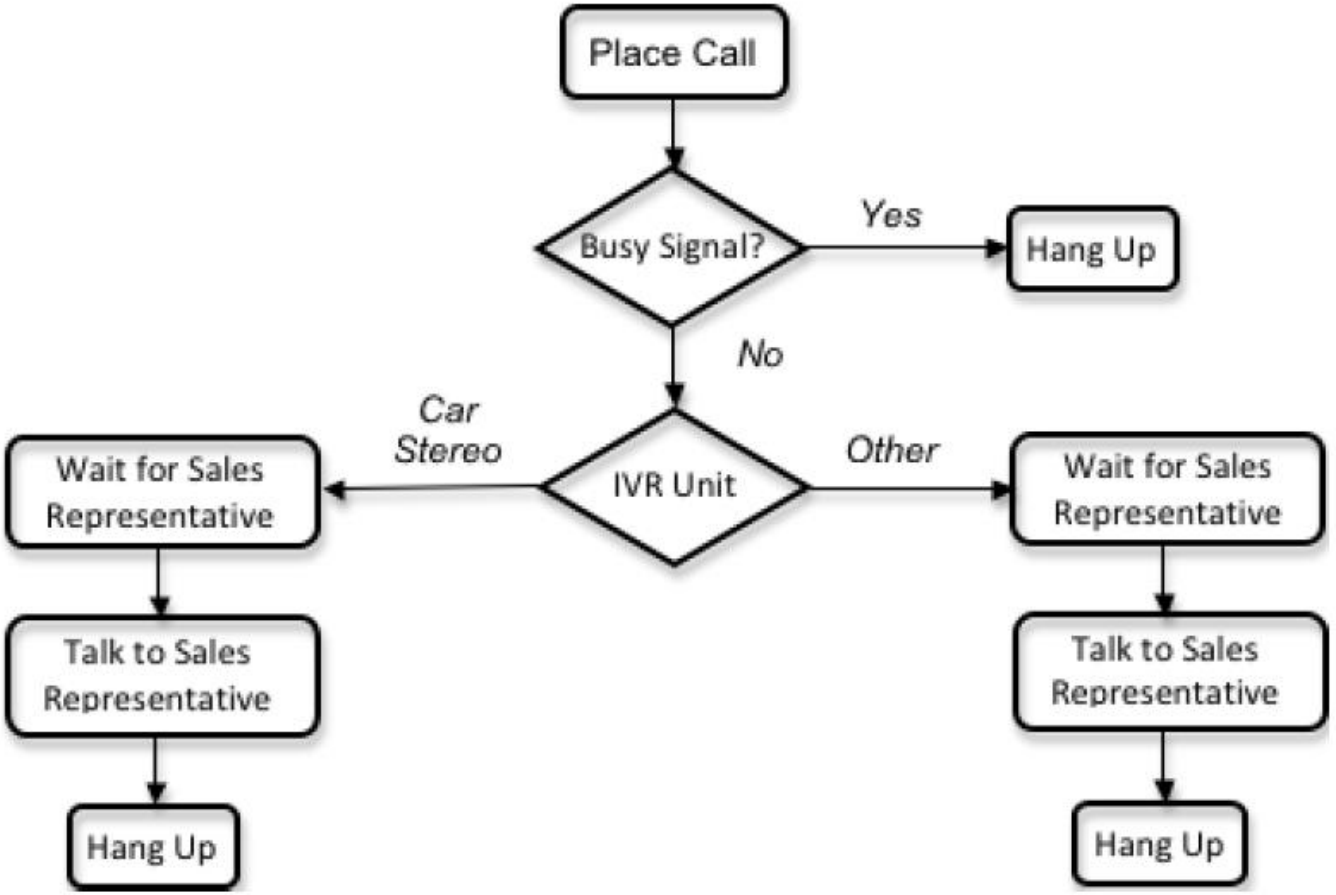
- Continuous vs. discrete
  - State variables change continuously over time or only at a discrete set of points in time
- Static vs. dynamic
  - Represent a system at a particular point in time or over a period of time
- Stochastic vs. deterministic
  - Have random variables as inputs or not

Model Taxonomy



Discrete Event Simulation, A First Course.
Steve Park and Larry Leemis, College of William and Mary

Example: Call Center



White and Ingalls, Introduction to Simulation, 2009

Matching the Concepts

- Telephone calls are dynamic entities with three attributes
  - productType (car stereo or others),
  - startTime (the time arrives at IVR)
  - BeginWait (the time joins the queue to begin waiting for the service)
- The system state has three variables
  - the number of calls in process on the IVR
  - the number of calls of car stereo type waiting for service
  - the number of calls of other types waiting for service
- Two resource
  - Sales representatives for the two types of products
  - Note that IVR has enough capacity and is not considered as a resource
- Two types of delay activities
  - calls receiving service from the two types of sales representatives
- Two types of queue activities
  - calls waiting in the two queues
- Several logic activities
  - E.g., IVR decides whether to accept a call or not
- Events ?

The Calendar at Noon

Table 1: The Calendar at Noon

Entity	Event	Event Time	Product_Type	Start_Time	Begin_Wait
34	Arrive at Call Center	12:01:40 PM	---	---	---
33	Complete Service IVR	12:02:40 PM	car-stereo	11:59:00 AM	---
10	Complete Service Car-Stereo	12:04:00 PM	car stereo	11:04:20 AM	11:07:20 AM
26	Complete Service Other	12:06:00 PM	other	11:43:20 AM	11:45:04 AM
27	Complete Service Other	12:07:00 PM	other	11:45:20 AM	11:46:16 AM
1	End replication	02:00:00 PM	---	---	---

Table 2: Car-Stereo Queue at Noon

Entity	Event	Event Time	Product_Type	Start_Time	Begin_Wait
11	Begin Service	---	car-stereo	11:17:30 AM	11:16:00 AM
24	Begin Service	---	car-stereo	11:37:20 AM	11:39:07 AM

Table 3: Other-Product Queue at Noon

Entity	Event	Event Time	Product_Type	Start_Time	Begin_Wait
28	Begin Service	---	other	11:47:40 AM	11:50:59 AM
30	Begin Service	---	other	11:51:40 AM	11:53: 12 AM
29	Begin Service	---	other	11:50:00 AM	11:53:47 AM
31	Begin Service	---	other	11:53:40 AM	11:55:28 AM
32	Begin Service	---	other	11:57: 46 AM	11:59:46 AM

White and Ingalls, Introduction to Simulation, 2009

Steps in Simulation

- Problem formulation
- Setting objectives
- Conceptual model: a high level idea
- Specification model: a document
- Computational model: a computer program
- Verified: program consistent to specification
- Validated: program consistent to the system

CONTENTS

- Theory and Concepts
  - Banks et al., Discrete Event System Simulation, Prentice Hall, 2001
  - White and Ingalls, Introduction to Simulation, Proc. Winter Simulation Conf., 2009
- Practice
  - Various online resources and ns2

Time and Execution

- Physical time: time in the physical system
  - Noon, December 31, 1999 to noon January 1, 2000
- Simulation time: time in the simulation
  - floating point values in interval [0.0, 24.0]
- Wall clock time: time of the execution
  - 9:00 to 9:15 AM on September 10, 1999
- Unpaced execution
  - independent simulation and wall clock time
- Paced execution
  - simulation time proportional to wall clock time

Fundamental Concepts

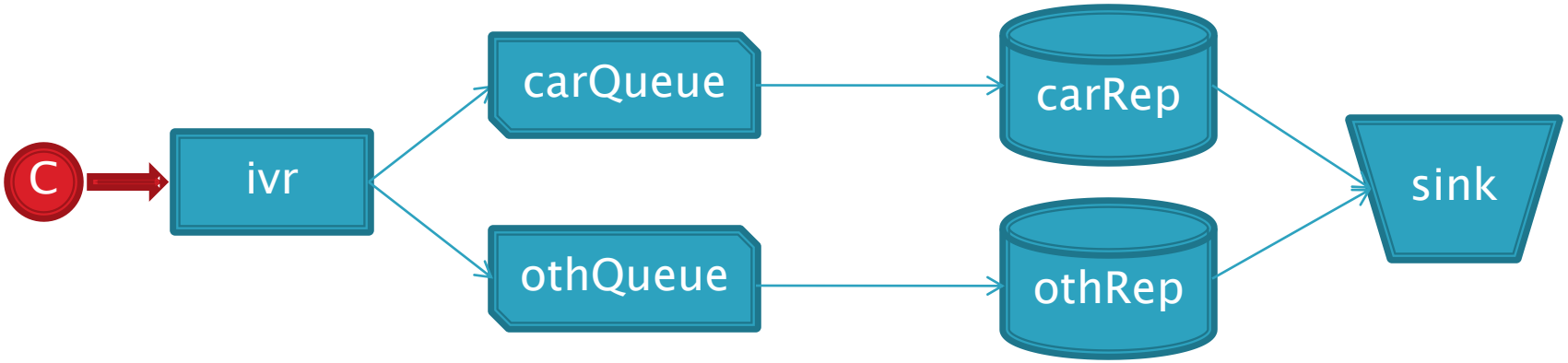
- System state
- Entity and attribute (object)
- Event
  - can modify state variables (state transition)
  - can schedule new events (insert them into eventQueue)
- Event queue, ordered by time
- Scheduler
  - update current time
  - invoke handler

```
while (true)
    if (! eQueue.empty()) {
        Event* e = eQueue.front();
        clock=e->time();
        e->handler()->handle(e);
    }
```

What On Earth Is an Event ?

- A logic activity is a event
- A delay activity is two events
  - firstEvent.handle() schedules secondEvent
- A queue activity is two events
  - secondEvent is scheduled by other means
- An external event (activity) is a input
  - AtEvent in ns2
  - start and stop are also inputs
- An internal event (activity) is the “meeting” of a dynamic entity with a static entity

Model for Call Center Example



- Dynamic entities: calls
- Static entities: five plus the sink
  - Assume a user takes a fixed time to dial one or two
  - Use a sink to clean up (e.g., write records)
- State variable: ivrCnt, carRepCnt, othRepCnt
  - Better to treat them as attributes of ivr, carRep, and othRep
- Inputs: (time, type) tuples in a file
  - Alternatively, we can treat them as events

Call Events

- Usually define the dynamic entity as the Event object in the program

  typedef struct {id; time; type; handler; sig} Event;

  - id: used to trace the history
  - time: scheduled time to happen
  - type: car stereo or others
  - handler: which static entity to handle the event
  - sig: how the handler should handle it

IVR

```
struct  Ivr {
    int cnt;
    const int Max;
    void handle(Event*);
    void decCnt();
} ivr;
```

Ivr::handle(Event* e)

- set handler to carQueue or othQueue according to type
- delay() returns a delay that the user dials 1 or 2
- can invoke sink.handle() instead of insert an event; notice that e->time is not updated

```
if (cnt < Max) {
    update event e={
        time+=delay(),
        handler=carQueue or othQueue,
        sig=ARRIVAL}
    eventQueue.insert(e.time, e)
    cnt++;
} else {
     update event e={
        handler=sink,
        sig=ARRIVAL}
     eventQueue.insert(e.time, e)
}
```

carQueue & othQueue

```
struct  Queue{
    list<Event*> queue;
    Rep* rep;
    void handle(Event*);
} carQueue, othQueue;
```

Queue::handle(Event* e)

- rep is initialized to carRep or othRep according to type
- DEPARTURE time is unknown, do not insert to eventQueue, put it somewhere else temporarily
- invoke rep to handle if the queue is previously empty

```
update event e={
    handler=rep,
    sig=ARRIVAL}
queue.insert(e);
if (queue.size()==1)
    e->rep.handle(e);
```

carRep & othRep

```
struct  Rep{
    int cnt;
    const int Max;
    Ivr *ivr;
    Queue* queue;
    void handle(Event*);
} carRep, othRep;
```

Queue::handle(Event* e)

- handle both ARRIVAL and DEPARTURE
- ivr, queue and Max are initialized according to the type
- delay() returns an interval to process the call

```
if (sig == ARRIVAL)
    while(cnt <Max && !queue.emtpy()) {
        cnt++; ivr->decCnt();
        e = queue.front(); queue.pop();
        update event e={
            time+=delay();
            handler=carRep;
            sig=DEPARTURE}
        eventQueue.insert(e);
    }
else if (sig==DEPARTURE) { cnt--;
    while(!queue.emtpy()) {
        cnt++; ivr.decCnt();
        e = queue.front(); queue.pop();
        update event e={
            handler=sink,
            sig=ARRIVAL
        eventQueue.insert(e);
    }
}
```

Summary

- An event queue is an ordered list of events,
- An event can be inserted into the event queue by another event or as inputs
- Scheduler run is a indefinite loop
- A logic activity can be
  - executed right away, or
  - scheduled as an event occurring at the current time
- A delay activity is two events, the second is scheduled by the first
- A queue activity is two events, the second needs to be temporarily stored somewhere, and is not put into the event queue by the first

Q&A

Exercise About ns2

- How to model a source that sends at rate B?
- How to model a point-to-point link LK with delay D and bandwidth B?
- How to model a CSMA/CD link?
- How to model a node that has an output buffer of size S and uses PQ scheduling?
- How to model the layer 2—4 processing of packets inside a node?
- How to model a routing protocol (e.g., RIP) on a network?