Discrete Event Simulation

Event-Oriented Simulation

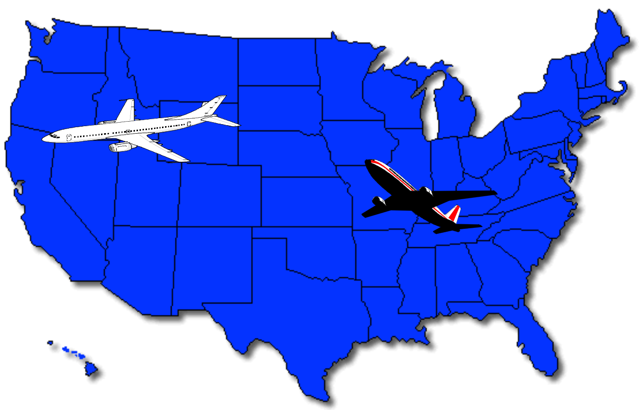
Richard M. Fujimoto
Professor

Computational Science and Engineering Division
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0765, USA

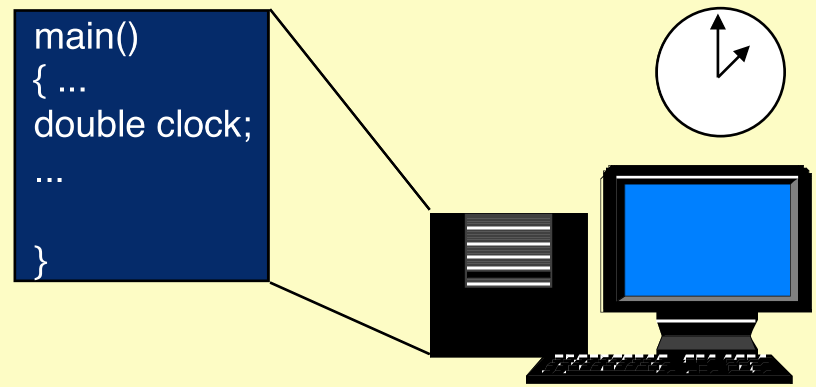
http://www.cc.gatech.edu/~fujimoto/

Time

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



physical system



simulation

- *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

Paced vs. Unpaced Execution

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., 2x wallclock 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.

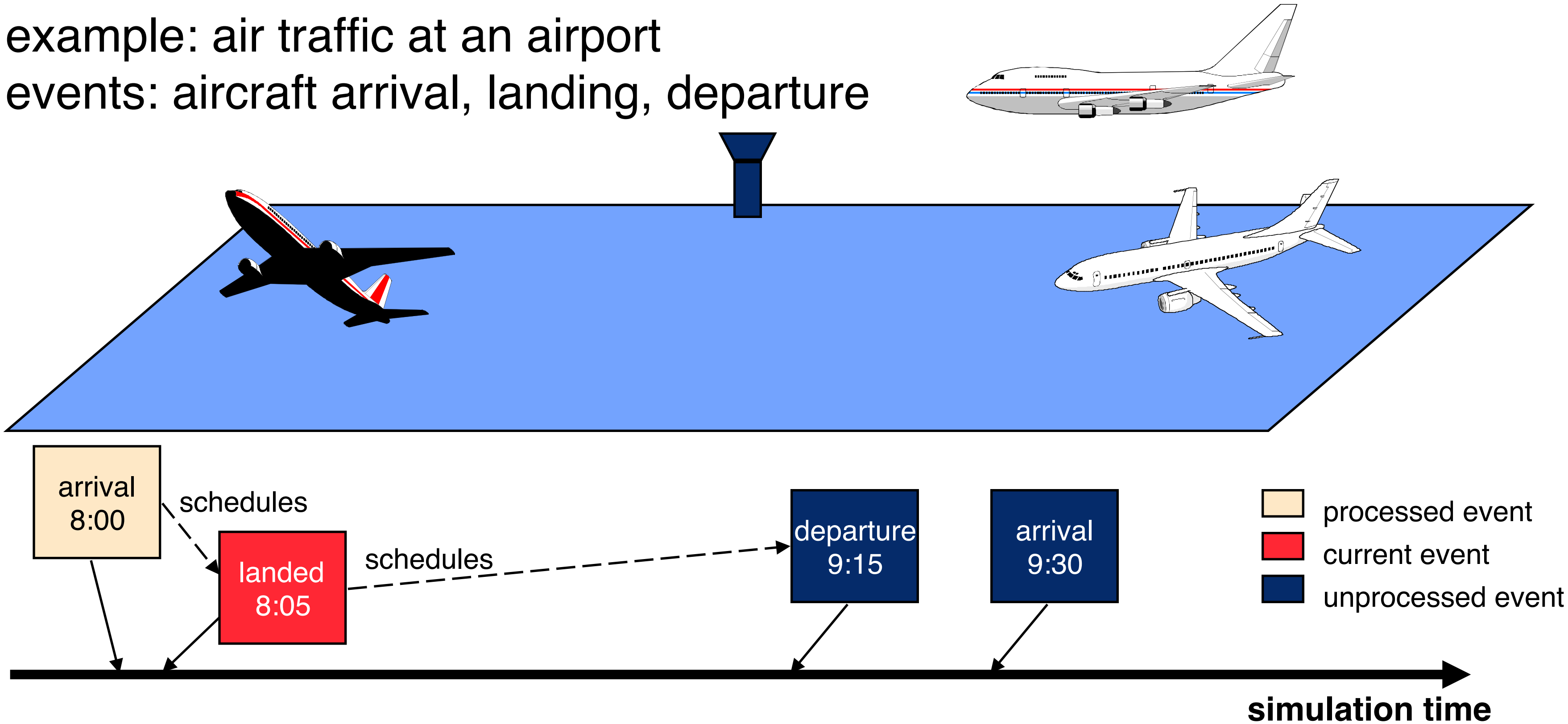
Fundamental concepts:
- system state (state variables)
- state transitions (events)

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

Event handler procedures

state variables

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

```
Arrival
  Event
{
    …
}

Landed
  Event
{
    …
}

Departure
  Event
{
    …
}
```

Simulation application

Simulation executive

Event processing loop

Now = 8:45

Pending Event List (PEL)

9:00

9:16

10:10

```
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 queueing

Single runway for incoming aircraft, ignore departure queueing

- `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;
```

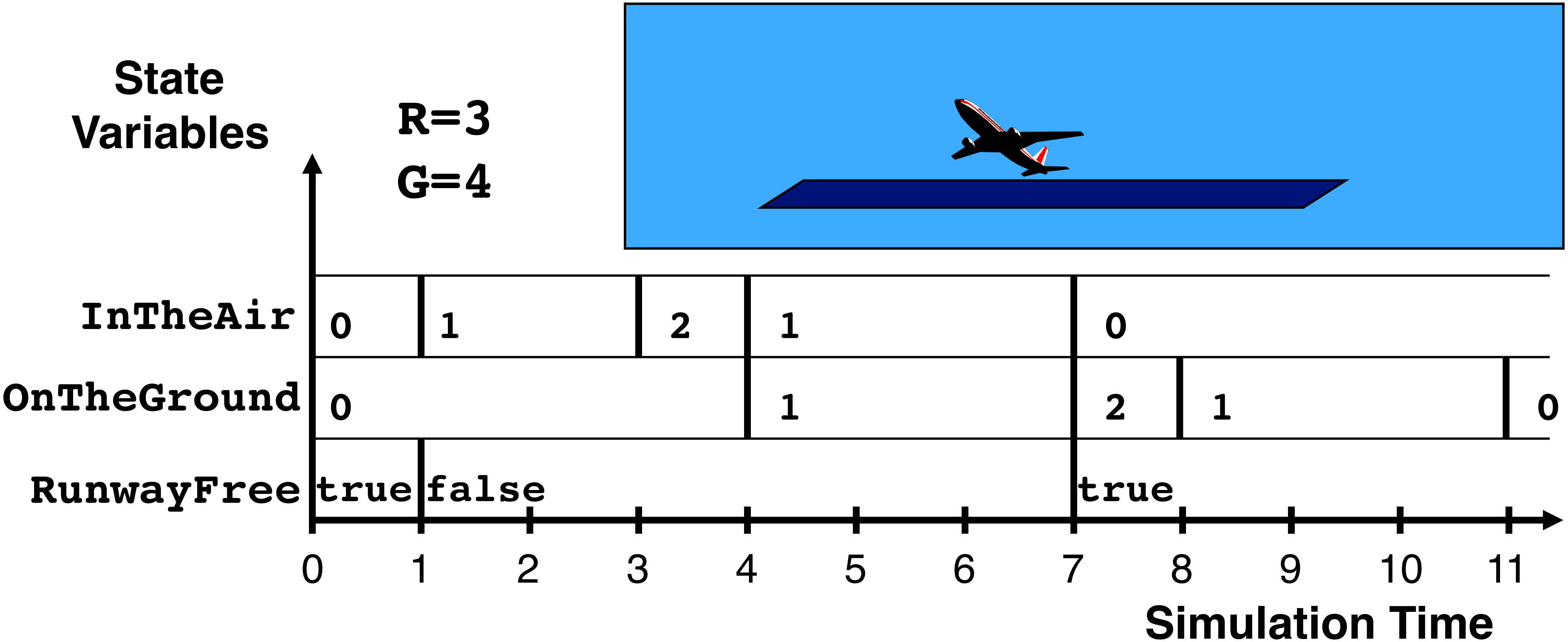
Departure Event

An aircraft now 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:

Arrival F1

Time	Event
1	Arrival F1
3	Arrival F2

Now=0

Arrival F2

Time	Event
3	Arrival F2
4	Landed F1

Now=1

Landed F1

Time	Event
4	Landed F1

Now=3

Landed F2

Time	Event
7	Landed F2
8	Depart F1

Now=4

Depart F1

Time	Event
8	Depart F1
11	Depart F2

Now=7

Depart F2

Time	Event
11	Depart F2

Now=8

Time	Event

Now=11

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