

Modeling, Simulation & Software Applications

1

Context

- ❑ Algorithms --- precise/unambiguous sequence of steps as basis for all computation
- ❑ Hardware --- physical realization of algorithmic machines.
- ❑ Virtual Machines --- idea of presenting an interface that is simpler than the actual physical entity.
 - more pleasant, more human user friendly

What if computers were used to emulate other physical systems?

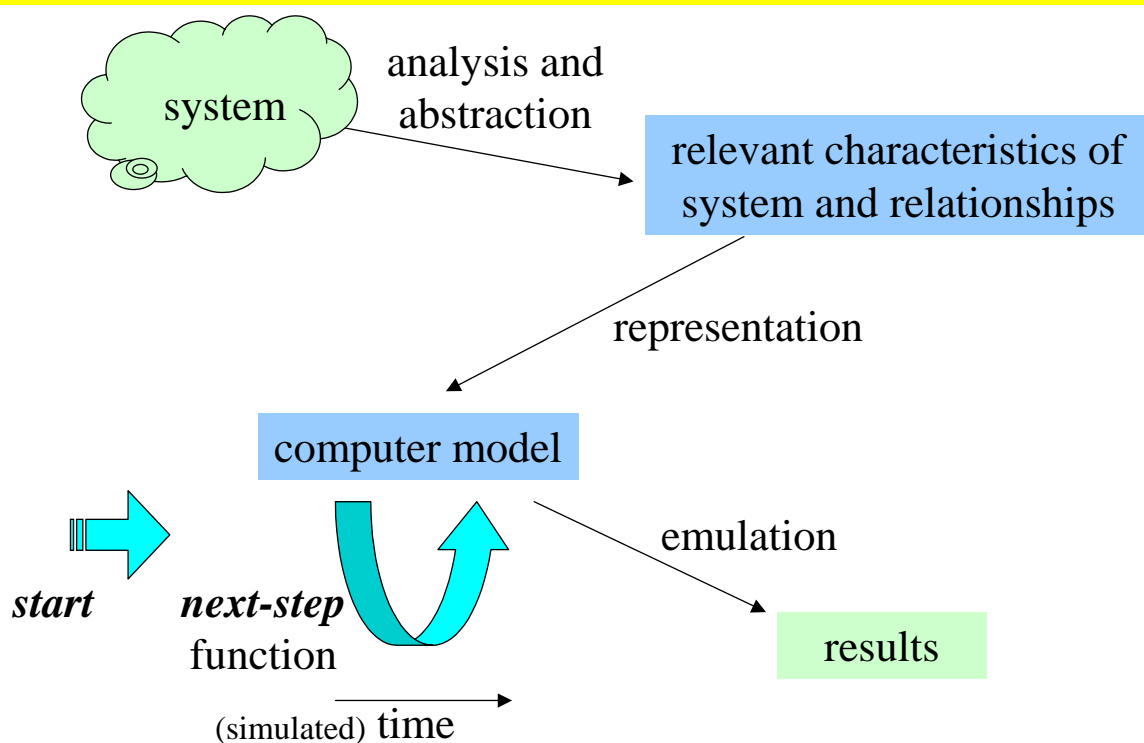
2

Scenarios for Emulation

- explore scenarios that may otherwise be physically impossible or too costly to build
 - Changi Airport terminal 4
 - size of waiting area?
 - number of check-in counters?
 - capacity of baggage handling facility?
 - effects of proposed ERP gantries
 - traffic volume?
 - average travelling speed?
 - predicting world population in year 2100
 - births, mortality, food, disease, natural disasters

3

Methodology



4

Car Race

□ Constraints:

- Max acceleration/deceleration 3m/s/s
- Max lateral acceleration 9m/s/s (i.e. $v^2/r < 9$)



What is the strategy for winning race?

5

State of Car

- position
- velocity
- direction
- skidded?
- completed?
- time since started

6

Representation

□ position	x position y position
□ velocity	x component of velocity y component of velocity
□ direction	x component of direction y component of direction
□ skidded?	false/true
□ completed?	driving/crashed/completed
□ time since started	0, 1, 2 time (in seconds)

7

Next-step

□ `drive(state, acceleration, turnRadius) → state`

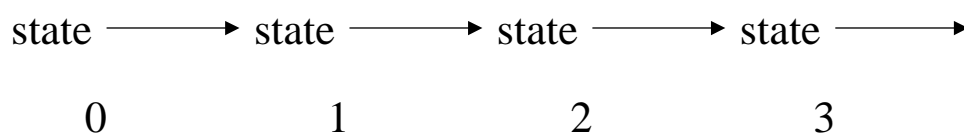
Increment simulated clock

Find new position

Update velocity/direction

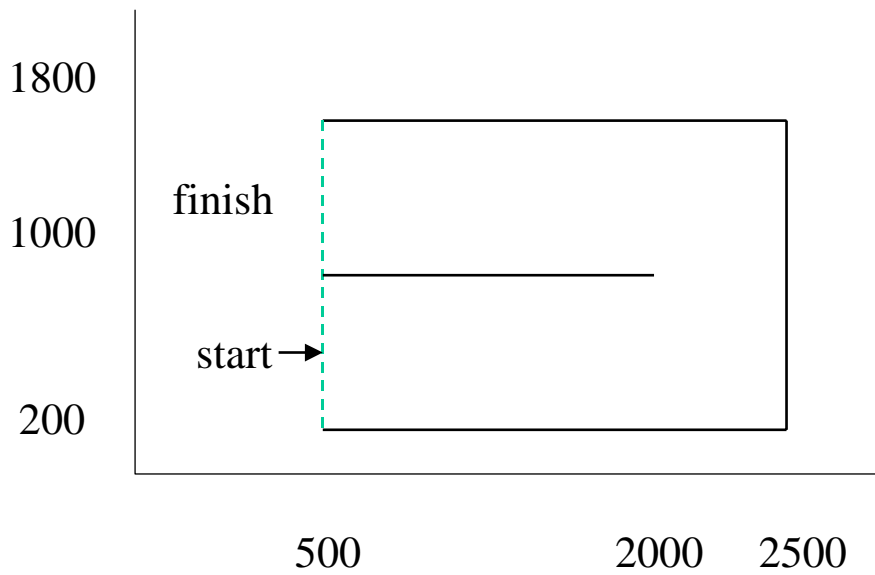
Has car skidded?

Has car crashed?



8

Actual Instance



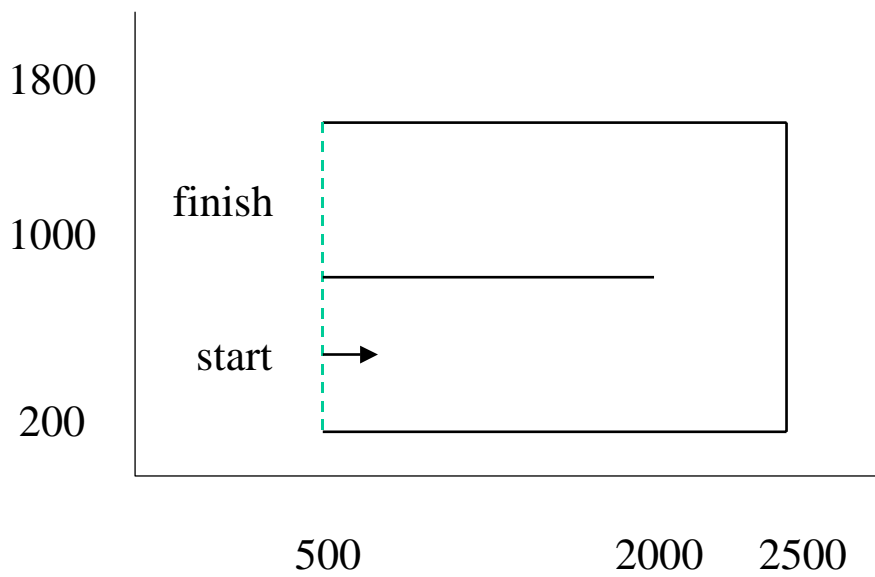
9

Starting out

```
a := 3;  
turnr := 0;  
while (time <= 10) & not completed do  
begin  
    state := drive(state, a, turnr);  
end
```

10

Starting out



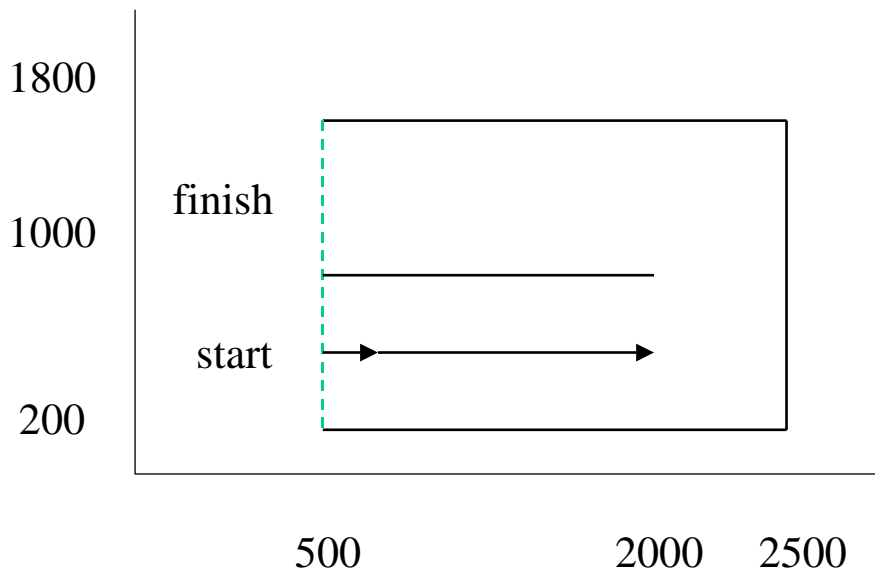
11

Constant speed

```
a := 0;  
turnr := 0;  
while (x <= 2000) & not completed do  
begin  
  state := drive(state, a, turnr);  
end
```

12

Constant Speed



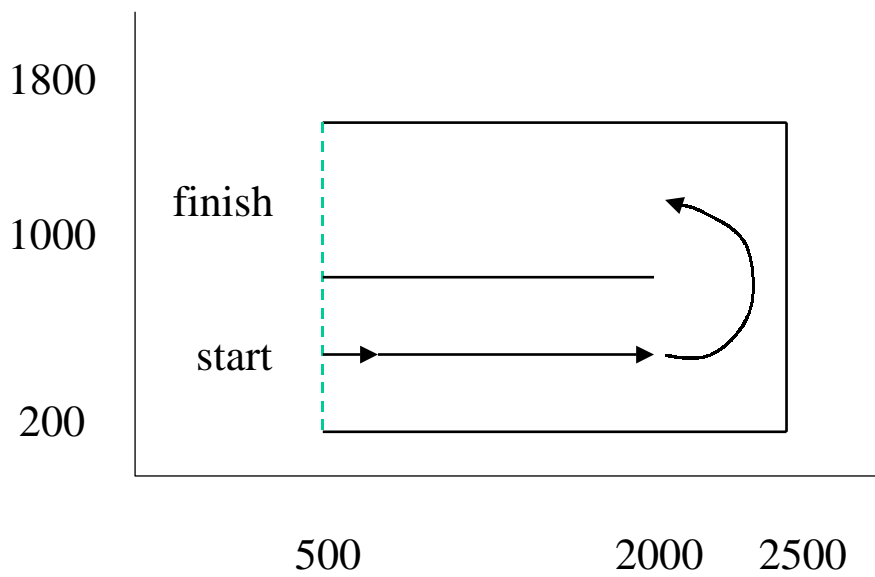
13

Turning

```
a := 0;  
turnr := -400;  
while (x > 2000) & not completed do  
begin  
    state := drive(state, a, turnr);  
end
```

14

Turning



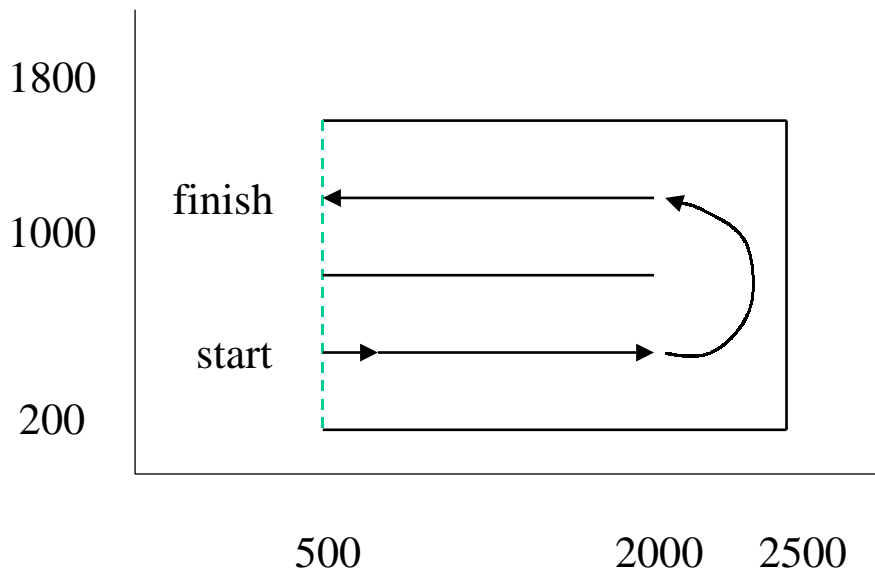
15

Speeding back

```
a := 3;  
turnr := 0;  
while not completed do  
begin  
  state := drive(state, a, turnr);  
end
```

16

Speeding back



17

Discrete-event simulation

□ Scenario:

- How many counters to open at fast food store?
- More counters involve more manpower and more slack during non-peak periods
- Long queues might cause potential customers to eat elsewhere.

18

Model of Customer

Customer arrives:

If (too crowded)

Leave;

Elapse(decide_time);

Wait for free counter;

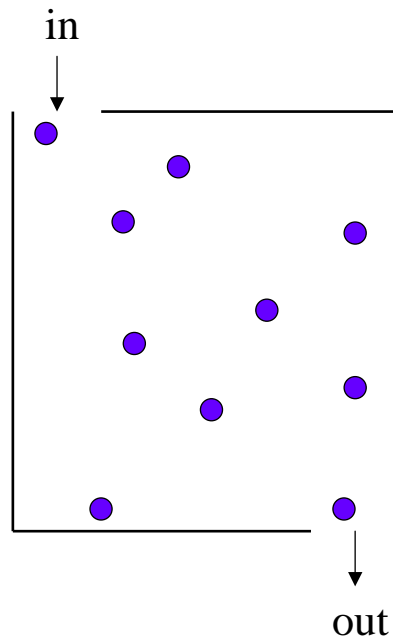
Elapse(order_time);

Wait for free table;

Elapse(eat_time);

Elapse(chat_time);

Leave;



19

Limited Resources

- Counters and Tables have associated queues
- Entity is suspended while on the queue and resumed when resource are available
 - Entity will resume another waiting for a resource when it has finished

20

Time

- ❑ Time is simulated so that simulation can be completed faster than what it actually takes
- ❑ Time need not be linear
- ❑ An event queue or calendar is maintained so that time can be advanced to the next event.

21

Modeling

- ❑ The results of a simulation are only as good as our understanding of such system and the accuracy of our model.
 - Some models are too complex and must be simplified
 - Are results of simplified models sufficient?

22

Related Areas

- ❑ Combat simulators
- ❑ Flight simulators
 - Too expensive to use real systems?
- ❑ Word processors
- ❑ CAD systems
 - Visualization of finished product?

23

Computer Applications

- ❑ Payroll system
- ❑ Inventory system
- ❑ Banking system
 - Accurate models of real world system
 - Inputs from real events in the world
- ❑ Simulation of popular control components in User-interfaces
 - radio buttons in choices
 - dials and sliders in ranges

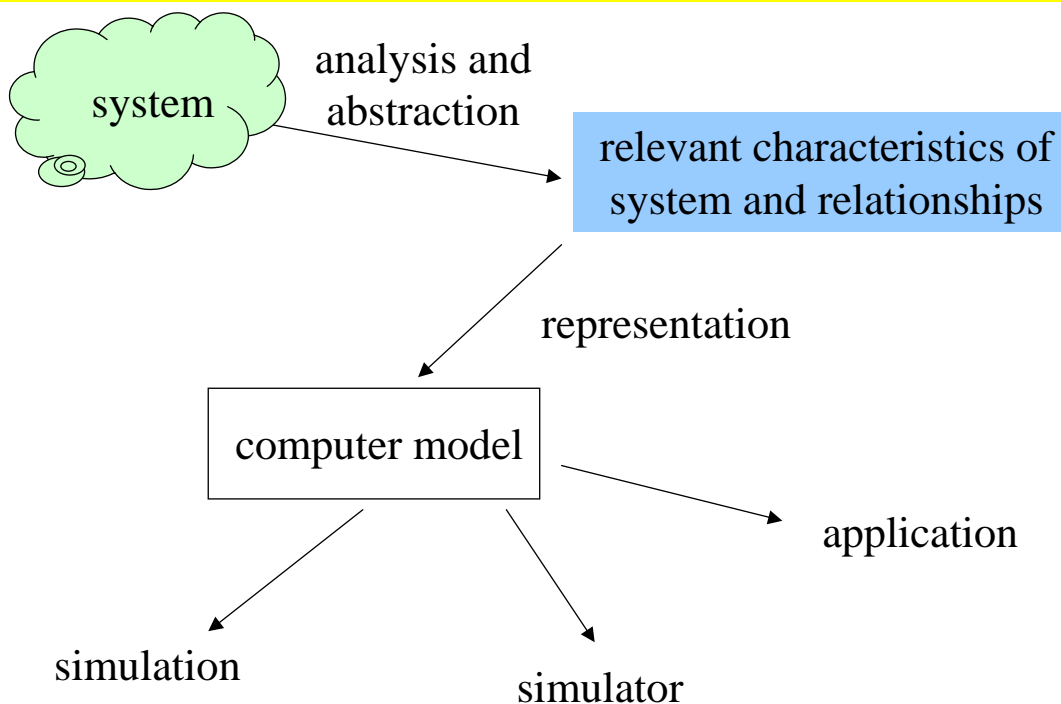
24

Real-time Systems

- Navigation system in missile
- Nuclear reactor control
 - time-critical, embedded computers
 - monitors environment to provide almost immediate responses
 - responses in turn affect environment

25

Summary



26