Scheduling Model Behavior

EES 4760/5760

Agent-Based & Individual-Based Computational Modeling

Jonathan Gilligan

Class #17: Tuesday Mar. 14 2017

Mousetrap model

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14.nlogo

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14_v2.nlogo

Scheduling Actions:

- Representing time:
  - Discrete (`tick`)
  - Continuous (`tick-advance`)
- Execution order
  - Synchronous
  - Asynchronous
    - Random order
    - Determined order

Repeating actions

- `repeat` repeats a certain number of times

```
repeat 5 [ wander ]
```

  or

```
repeat random count turtles [ wander ]
```

- `while` repeats as long as a condition is true

```
while not any? turtles-here [ wander ]
```

- `loop` repeats forever (until `stop` or `report`)

```
loop [
  wander
  if any? turtles-here [ stop ]
]
```

Discrete vs. continuous time

- Almost all models use discrete time:
  - `tick` advances tick counter by 1.
  - `ticks` is always an integer.
- Continuous time
  - `tick-advance 2.3`
- Things to think about:
  - When to tick?

```
to go
  ask patches [ do-patch-stuff ]
  ask turtles [ do-turtle-stuff ]

  tick
  if ticks > run-duration [stop]
end
```

```
to go
  tick
  if ticks > run-duration [stop]

  ask patches [ do-patch-stuff ]
  ask turtles [ do-turtle-stuff ]
end
```

Order of execution

- `ask`: Asks turtles in a random order.
  - ```
    ask turtles [do-sales]
    ```
- Suppose we wanted bigger turtles to act before the smaller ones?
  - ```
    foreach sort-on [size] turtles [ ask ? [do-sales] ]
    ```

Concurrent execution

- `ask-concurrent` (**not recommended**)

```
to go
  ask turtles [turtle-action]
end

to go-concurrent
  ask-concurrent turtles [turtle-action]
end

to turtle-action
  ask one-of patches with [pcolor = blue]
  [
    set patch-value patch-value + 1
    set pcolor red
  ]
end
```

- What is the difference between `go` and `go-concurrent`?

ask vs. ask-concurrent

```
ask patches [ set patch-value 0 ]
ask turtles [turtle-action]

to turtle-action
  ask one-of patches with [pcolor = blue]
  [
    set patch-value patch-value + 1
    set pcolor red
  ]
end
```

- Each turtle finishes everything in brackets before the next turtle starts
  1. First turtle checks `[pcolor] of patch 20 20`: it's blue
  2. First turtle increments `patch-value` (1)
  3. First turtle sets `pcolor` to red
  4. Second turtle checks `[pcolor] of patch 20 20`: it's red
  5. Second turtle checks another patch
- `[patch-value] of patch 20 20` is 1

ask vs. ask-concurrent

```
ask patches [ set patch-value 0 ]
ask-concurrent turtles [turtle-action]

to turtle-action
  ask one-of patches with [pcolor = blue]
  [
    set patch-value patch-value + 1
    set pcolor red
  ]
end
```

- Multiple turtles running at the same time, with no synchronization.
  1. First turtle checks `[pcolor] of patch 20 20`: it's blue
  2. Second turtle checks `[pcolor] of patch 20 20`: it's blue
  3. Second turtle increments `patch-value` (1)
  4. First turtle increments `patch-value` (2)
  5. First turtle sets `pcolor` to red
  6. Second turtle sets `pcolor` to red
- `[patch-value] of patch 20 20` is 2

Synchronous vs. asynchronous updating

- What is the difference?
- When would you want to use one or the other?
  - Business investor model?
  - Telemarketer model?
- How would you do *asynchronous* updating?
- How would you do *synchronous* updating?
  - Hidden state-variables (turtle can't see other turtle's hidden variables)
  - Two ways:
    1. Break submodel into two parts:
       1. Turtles sense and update hidden state-variables that others can't sense
       2. Update environment (including state-variables that others can sense)
    2. Make shadow copy of all state variables:
       1. Sensing sees originals, updates change shadow-copies
       2. Update the original (`set original shadow-copy`)

Mousetrap model

https://youtu.be/XIvHd76EdQ4

Mousetrap model

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14.nlogo

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14_v2.nlogo

Mousetrap model

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14.nlogo

https://ees4760.jonathangilligan.org/models/class_17/Mousetrap_Ch14_v2.nlogo

- Play with models
- Compare continuous updating with updating on ticks