Ecological modelling with Simile

Lecture 4

Part A: Conditional submodels
Part B: Association submodels

Robert Muetzelfeldt
Jasper Taylor
Jonathan Massheder

www.simulistics.com

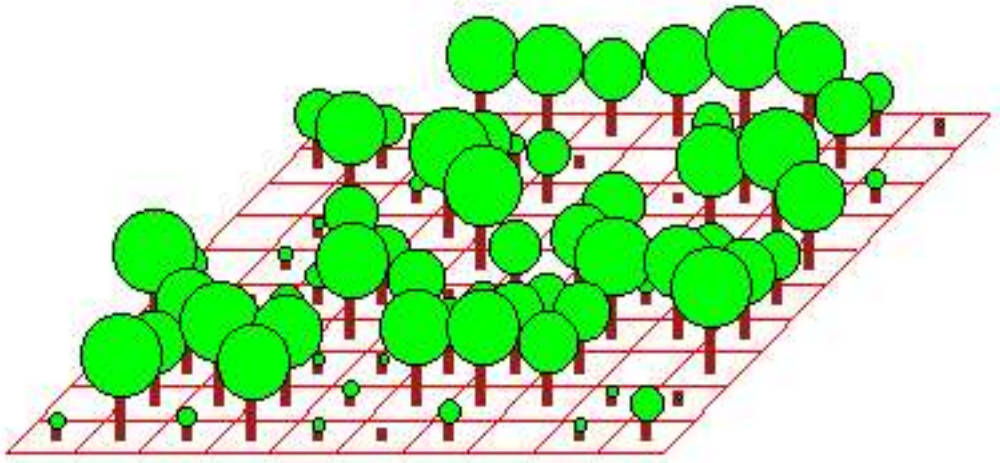
Feb 2006

Part A
Conditional submodels

What are conditional submodels used for?

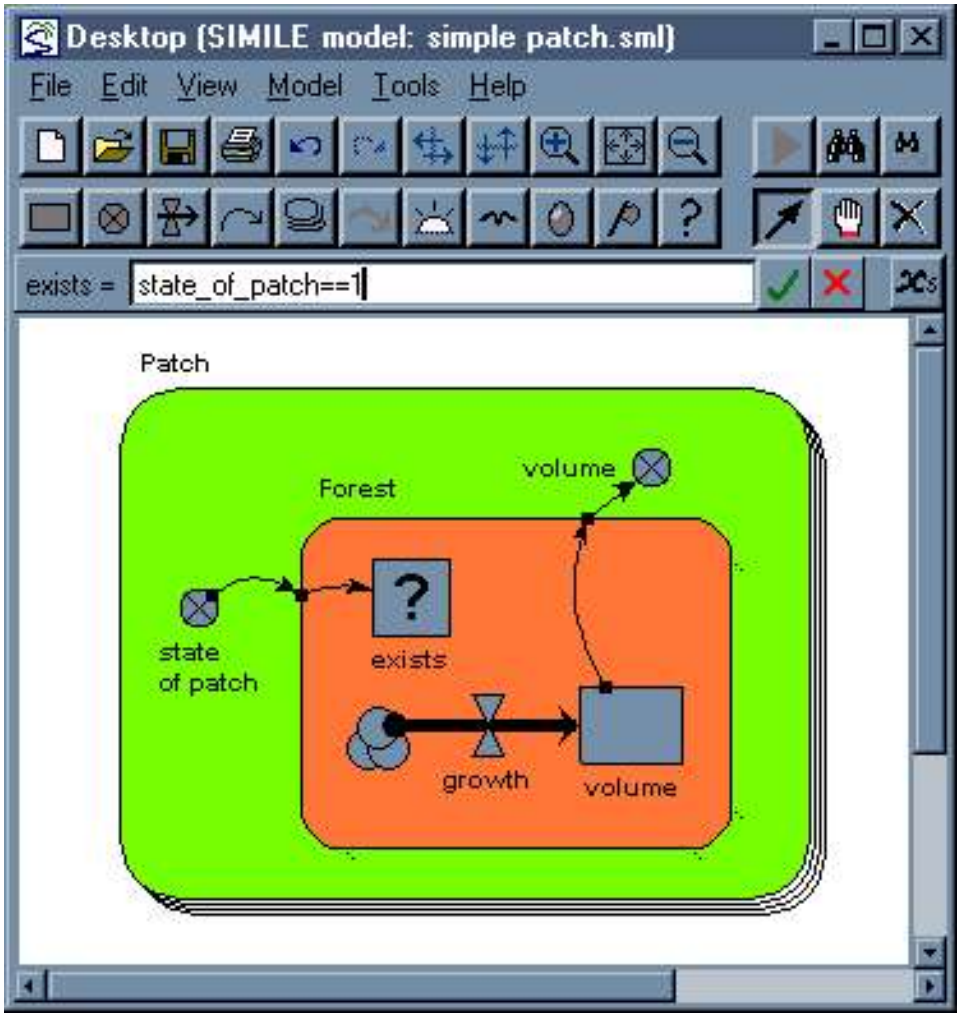
A particular subsystem that may or may not be present e.g.:

- In different instances (forest on land)
- at different times (seasonal plants)
- for different conditions (plankton in ocean)

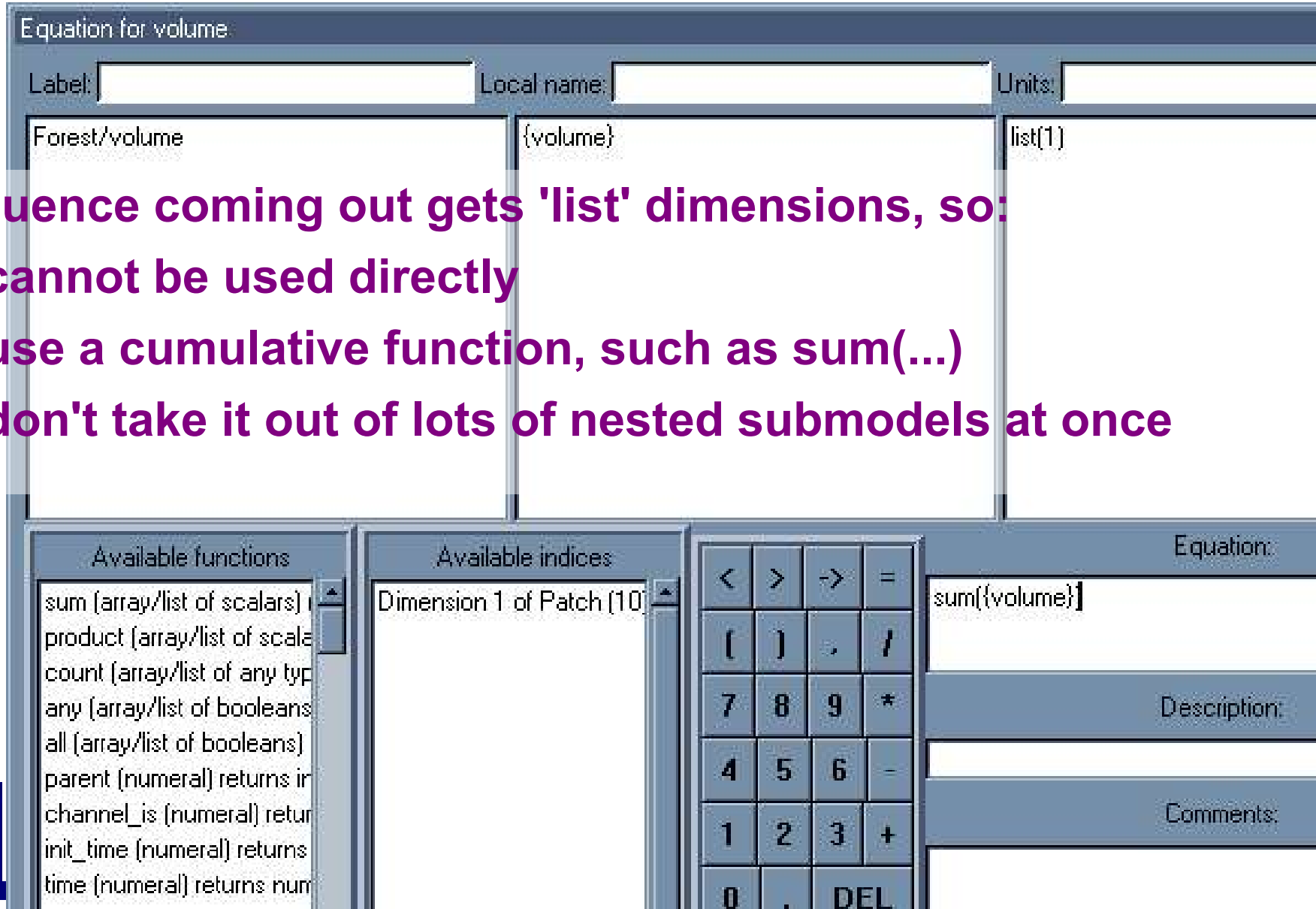


How do we set one up?

- A submodel with a ‘condition’ component
- Square box with ‘?’ on
- Equation is boolean, like what comes after ‘if’
- Submodel can have dimensions, but must not be a population
- Diagram has ‘…’ notation



Using variable-membership models



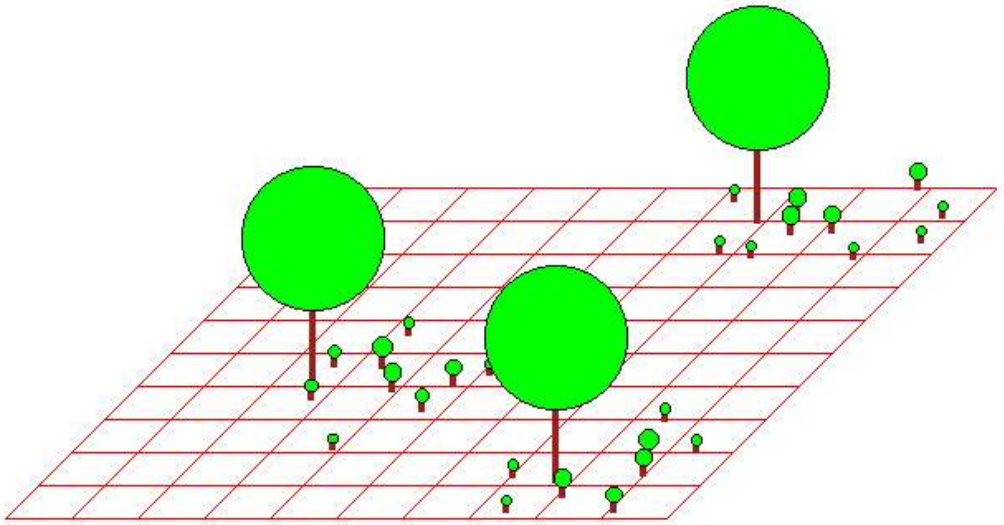
Influence coming out gets 'list' dimensions, so:

- cannot be used directly
- use a cumulative function, such as sum(...)
- don't take it out of lots of nested submodels at once

Part B
Association submodels

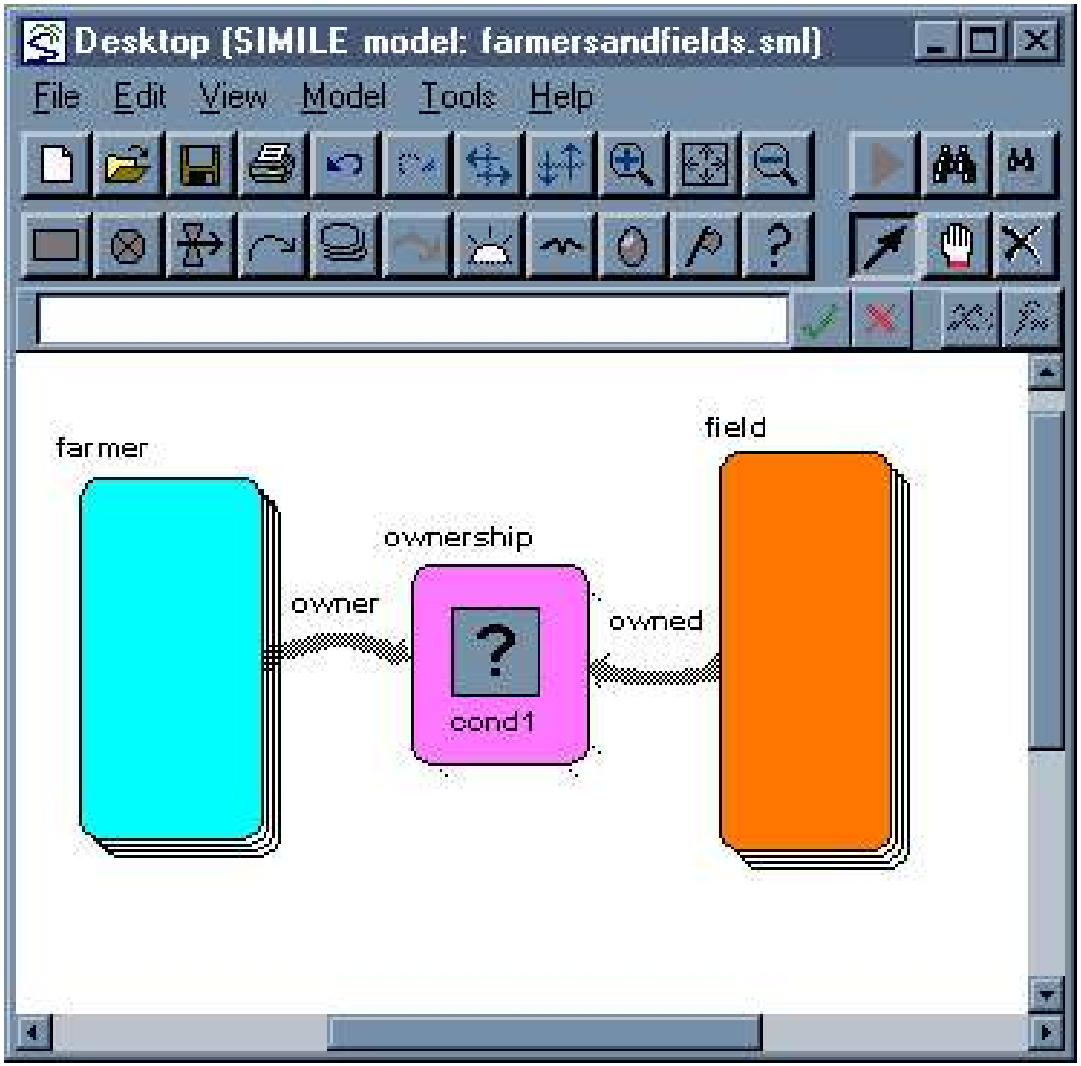
What are association submodels used for?

- Interactions that depend on proximity, similarity, reference
- Move values between instances in variable-membership submodels
- Speed up calculations of fixed interactions

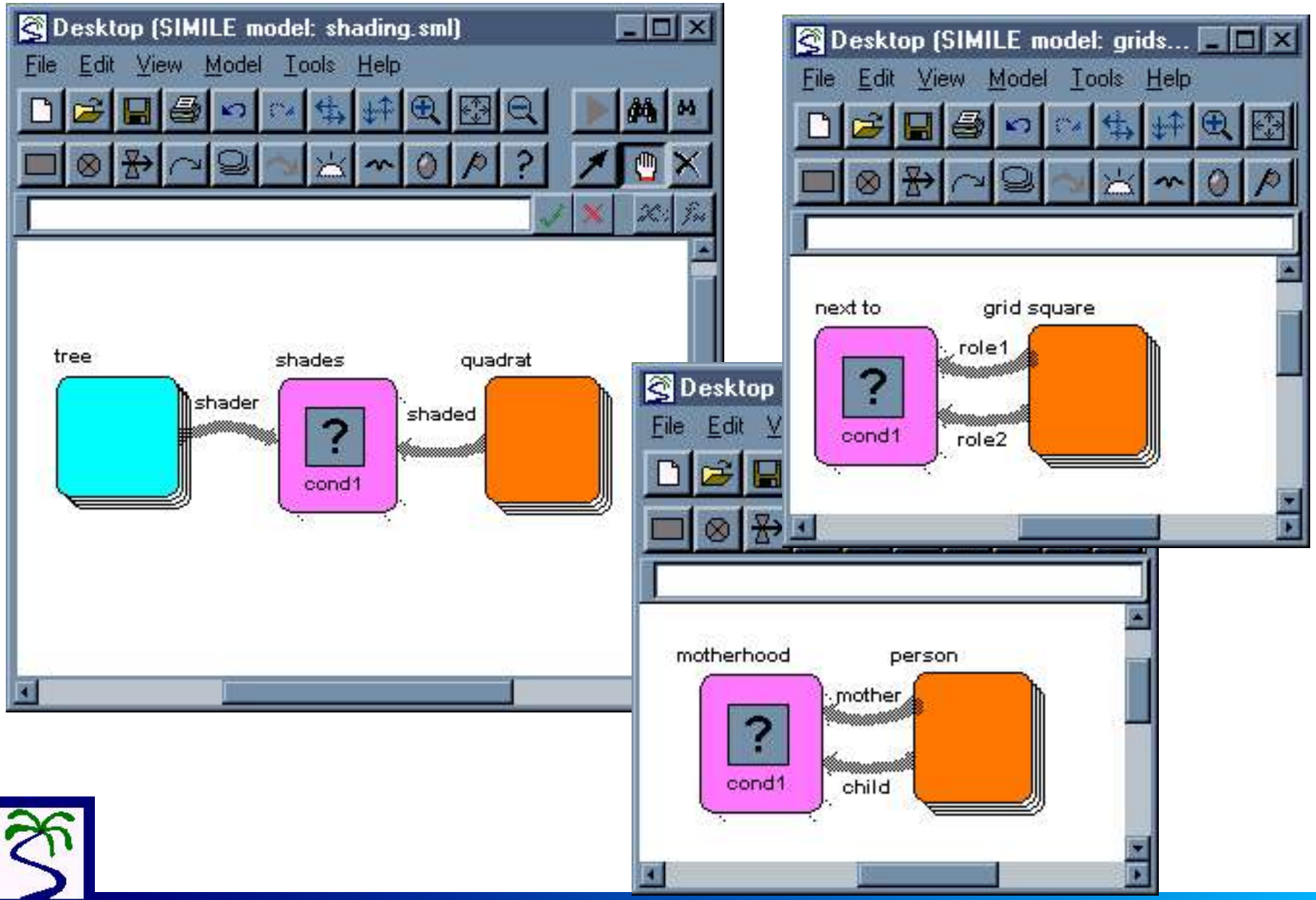


Examples of associations

- Neighbourhood, land use change
- Proximity:
- shading, seeding, grazing
- Fixed association: parenthood
- Social constructs: tenure, territory, mating



More examples of associations



Example 1: Field ownership

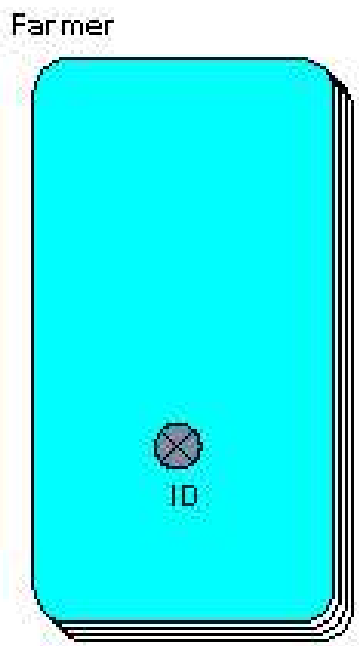
We want to model a collection of farmers owning a collection of fields. Each field is owned by one farmer; one farmer can own several fields.

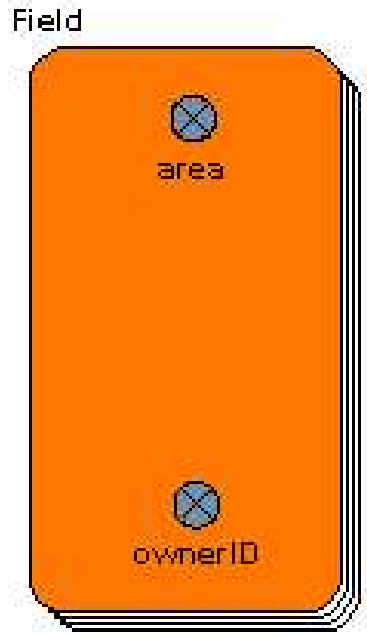
Information is to be transferred from a field to the farner that owns it: e.g. the area of the field, the yield obtained from it, etc.

Field ownership: step 1

Create the submodels representing the farmers and fields

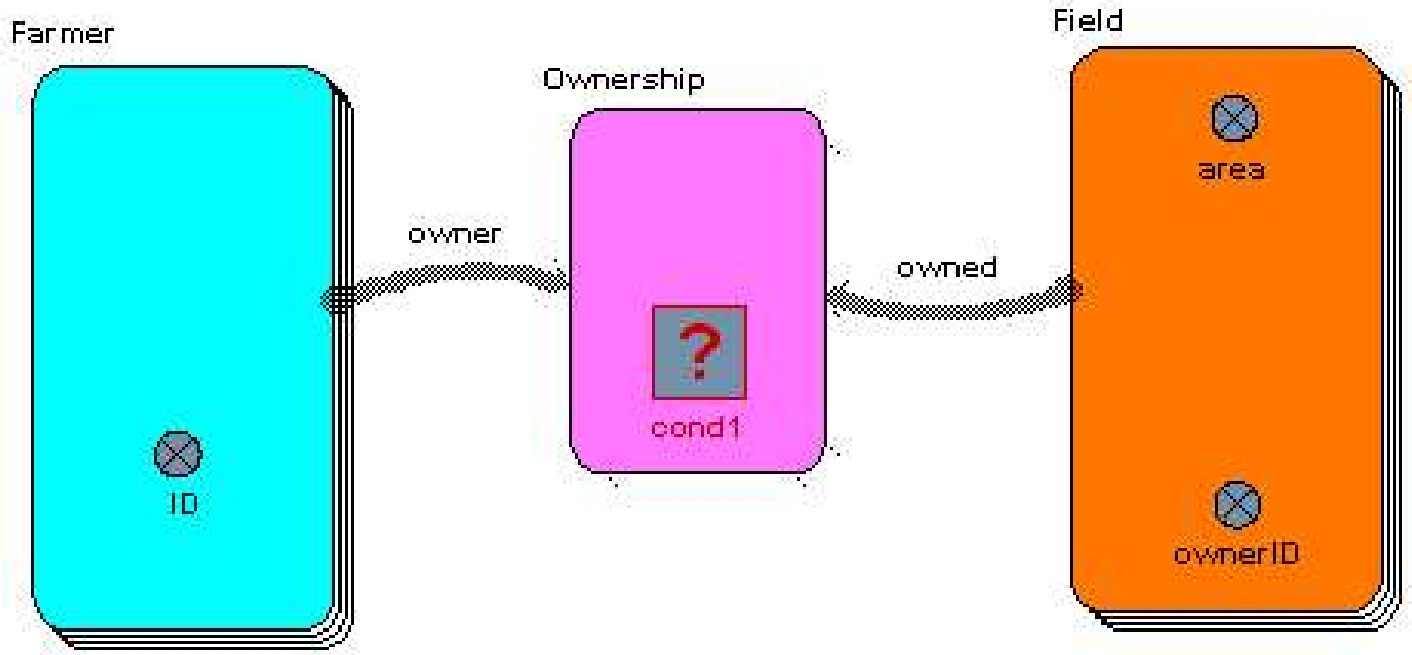
```
ID = index(1)
ownerID = int(rand_const(0,10))+1
area = 1
```





Field ownership: step 2

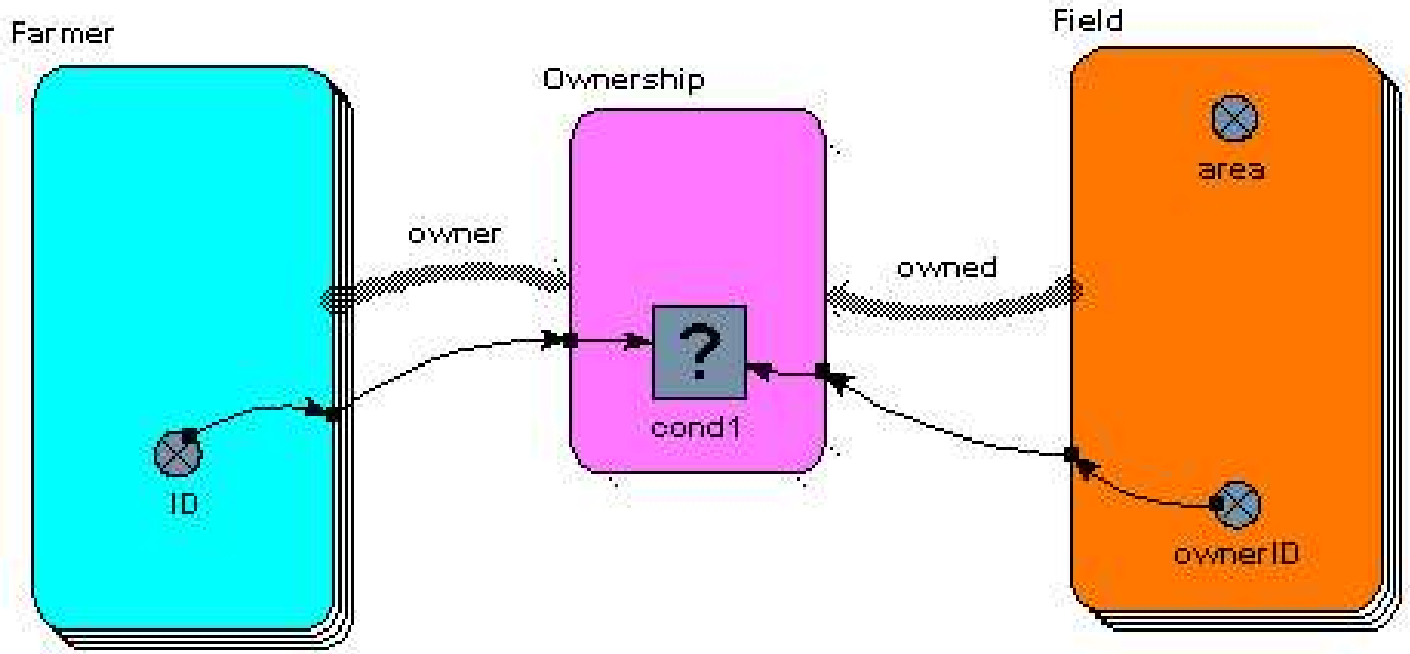
Add the ‘ownership’ submodel.



Field ownership: step 3

Create the association between farmers and fields

cond1 = ID == ownerID

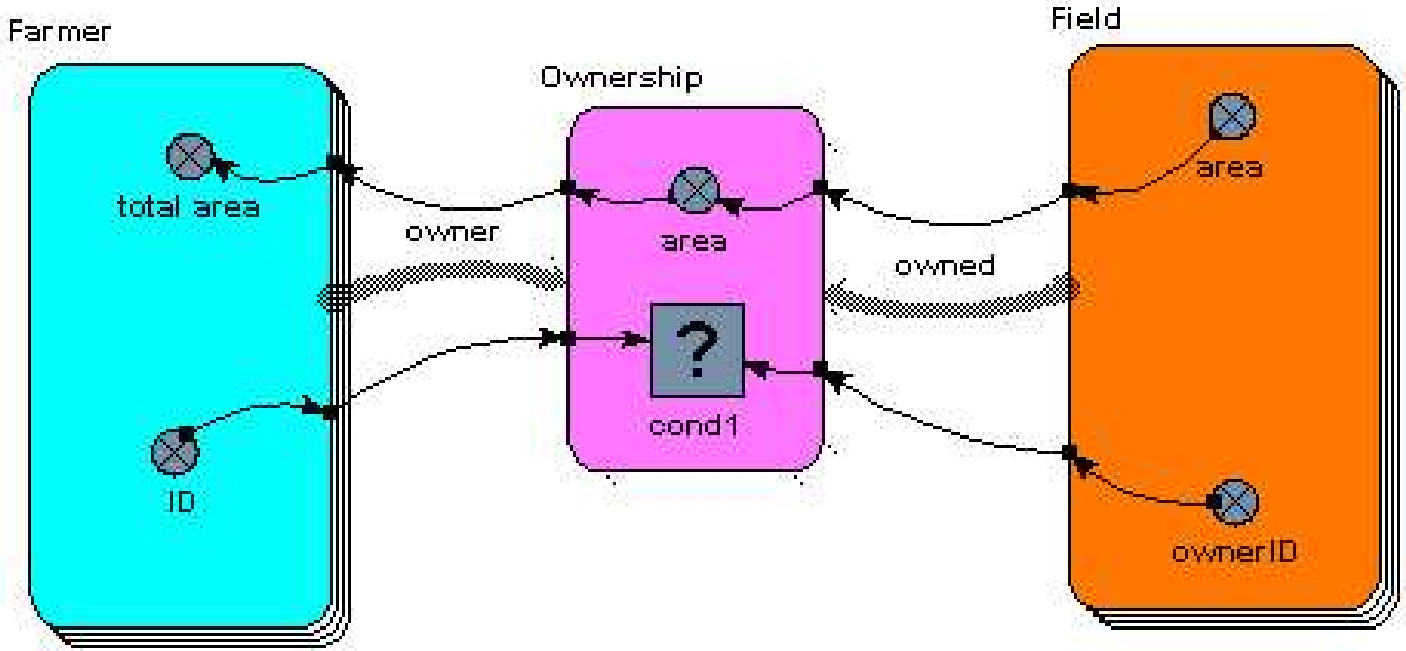


Field ownership: step 4

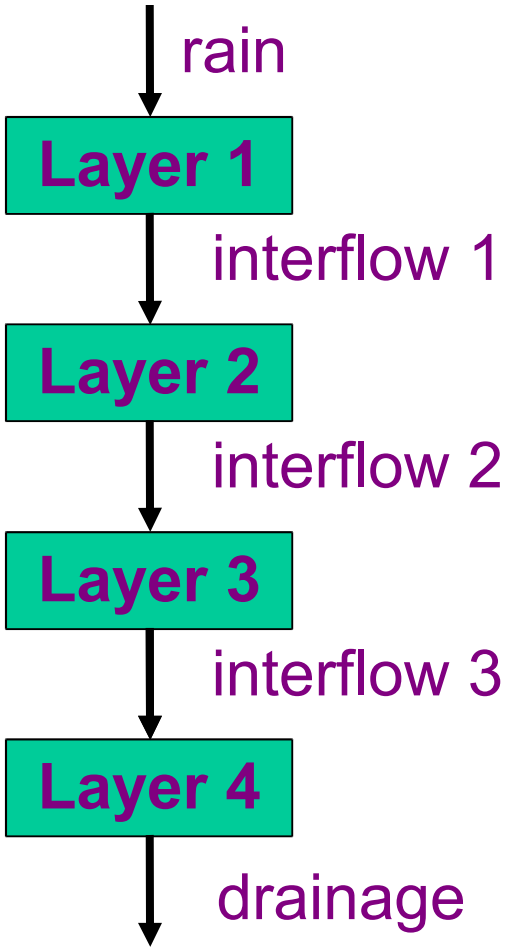
Work out the total area per farmer

Farmer/total area = `sum({area}) (to farmer in owner)`

Ownership/area = `Field/area (from Field in owned)`



Example 2: Water flow between soil layers



Problem: to simulate soil water dynamics

- Illustrates the use of Simile for 1-D spatial modelling
- Implements the concept of the 'above' association between Layers

Water flow between soil layers: step 1

1) Create the compartments

Layer 1

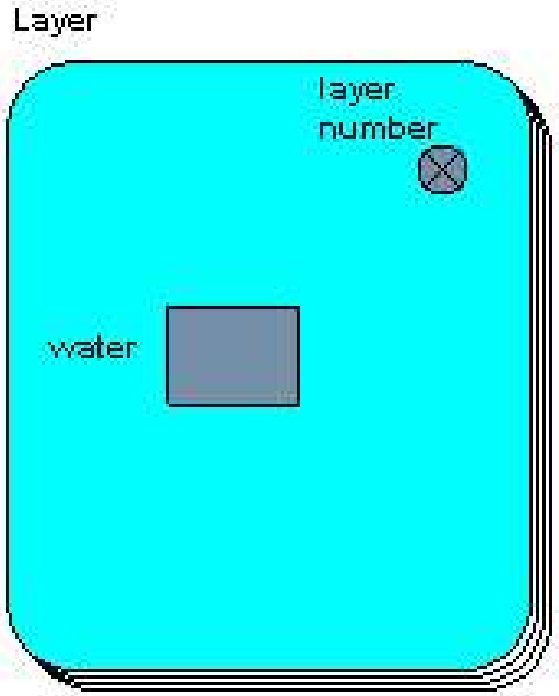
Layer 2

Layer 3

Layer 4

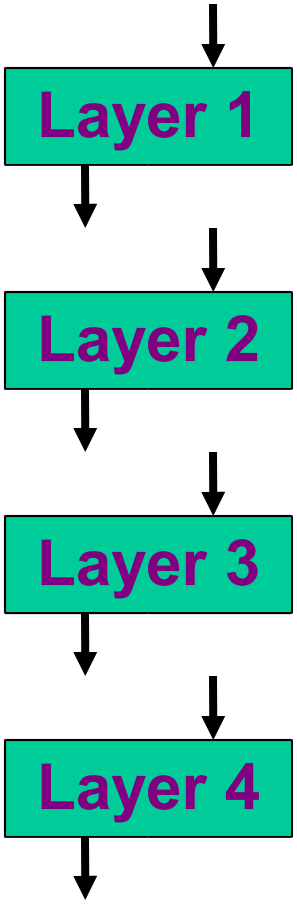
```
layer_number = index(1)

water = 0
```

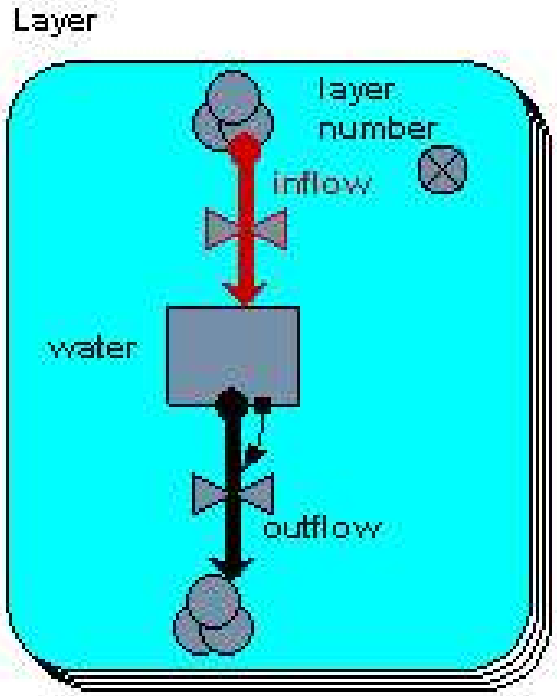


Water flow between soil layers: step 2

Add in the flows

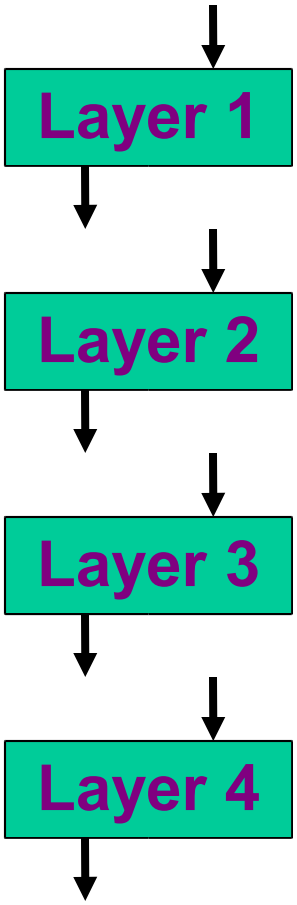


outflow = 0.2*water

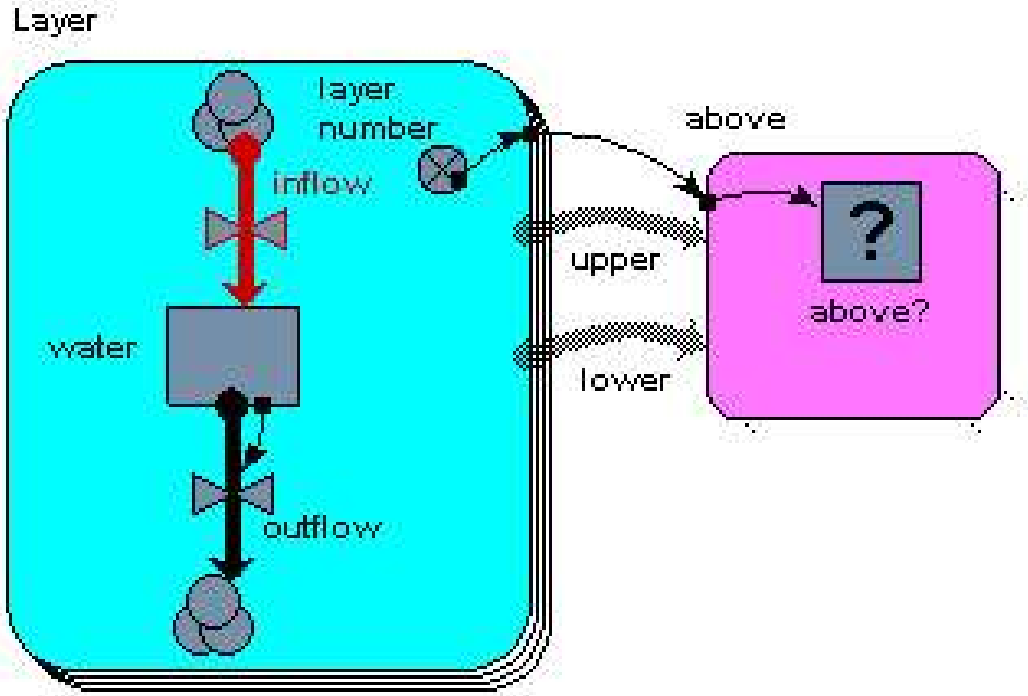


Water flow between soil layers: step 3

Create the 'above' association between layers

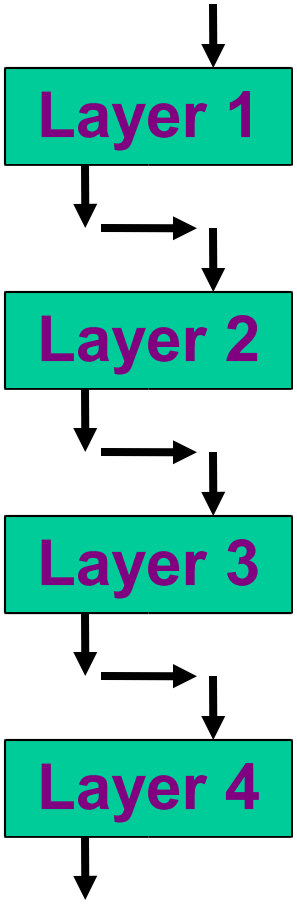


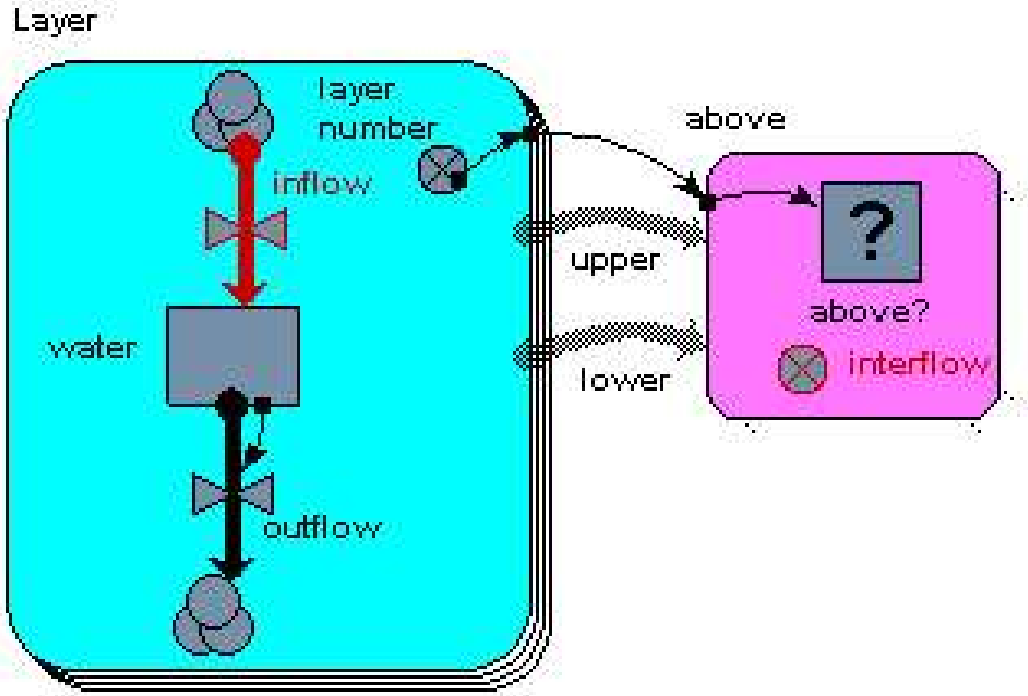
above? = `layer_number-layer_number_0`



Water flow between soil layers: step 4

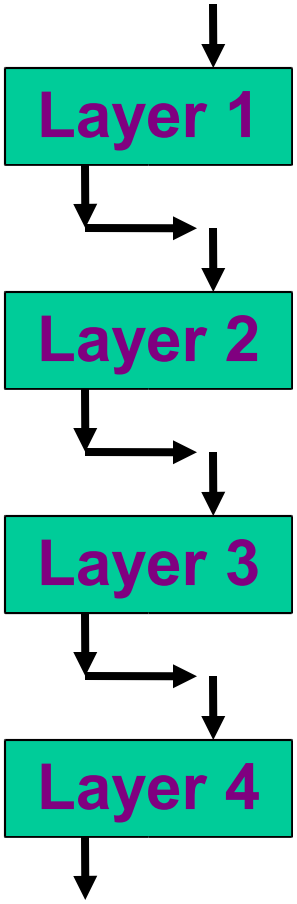
Create a variable to link the flows



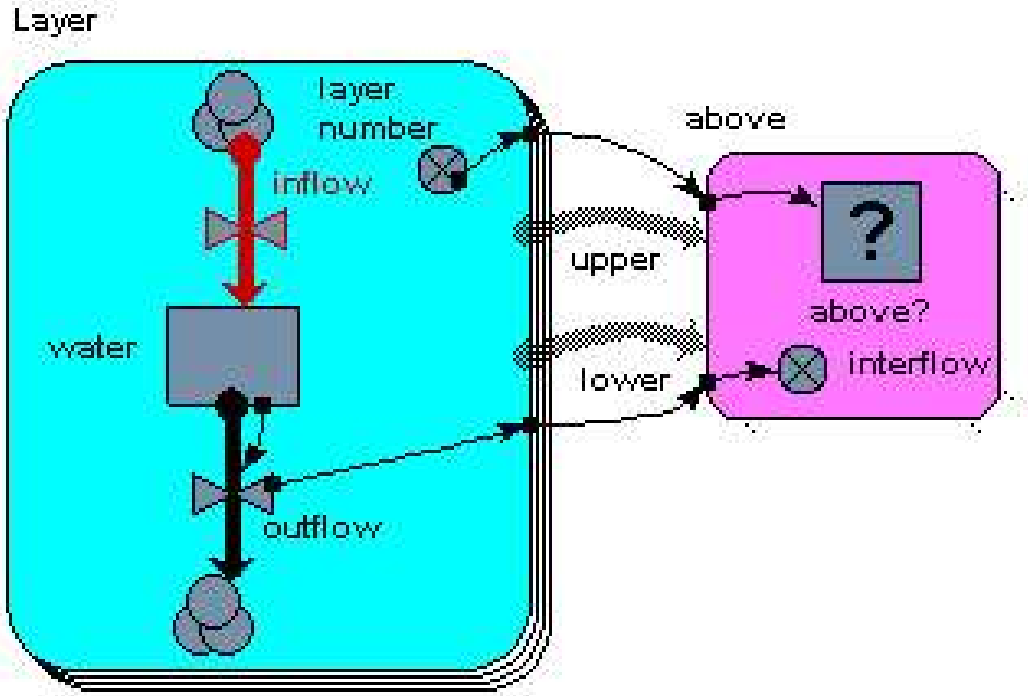


Water flow between soil layers: step 5

Link the variable to the outflow of the ‘upper’ level

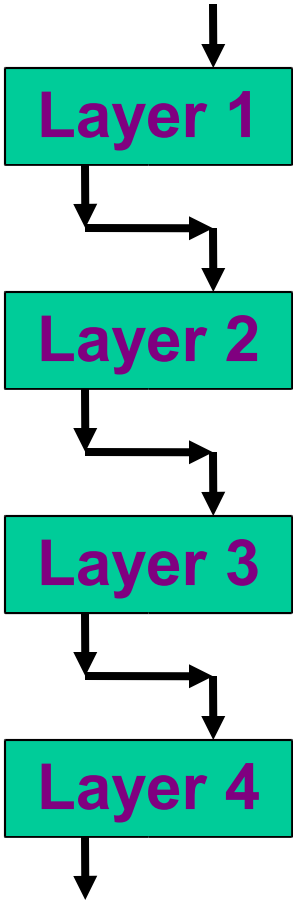



```
interflow = outflow (from Layer in upper)
```

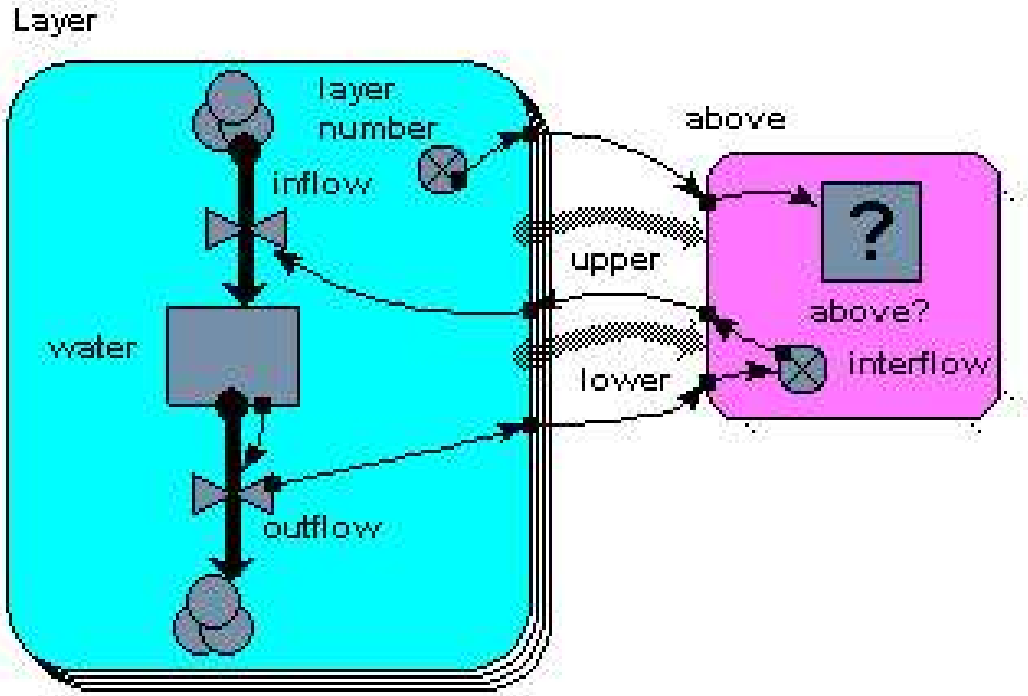


Water flow between soil layers: step 6

Link the variable to the inflow of the ‘lower’ level

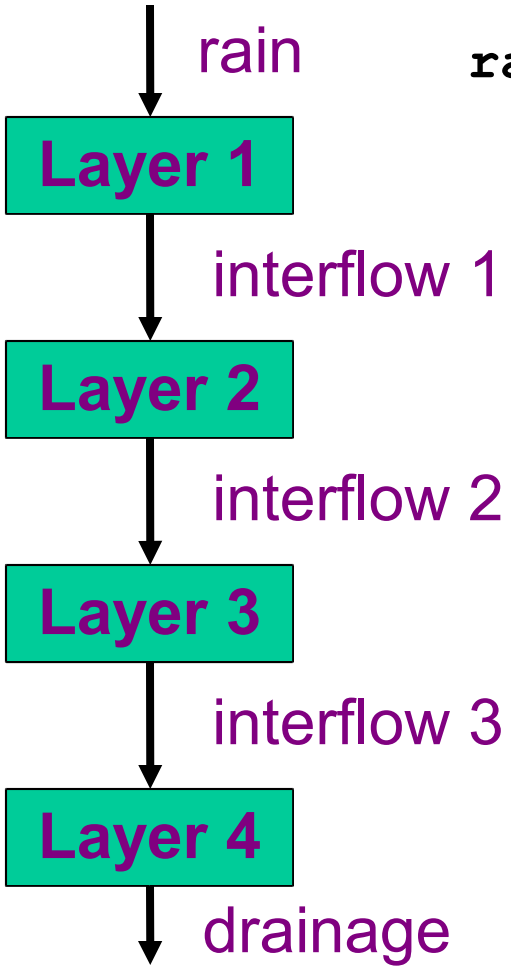


inflow = `interflow_0 (to Layer in lower)`

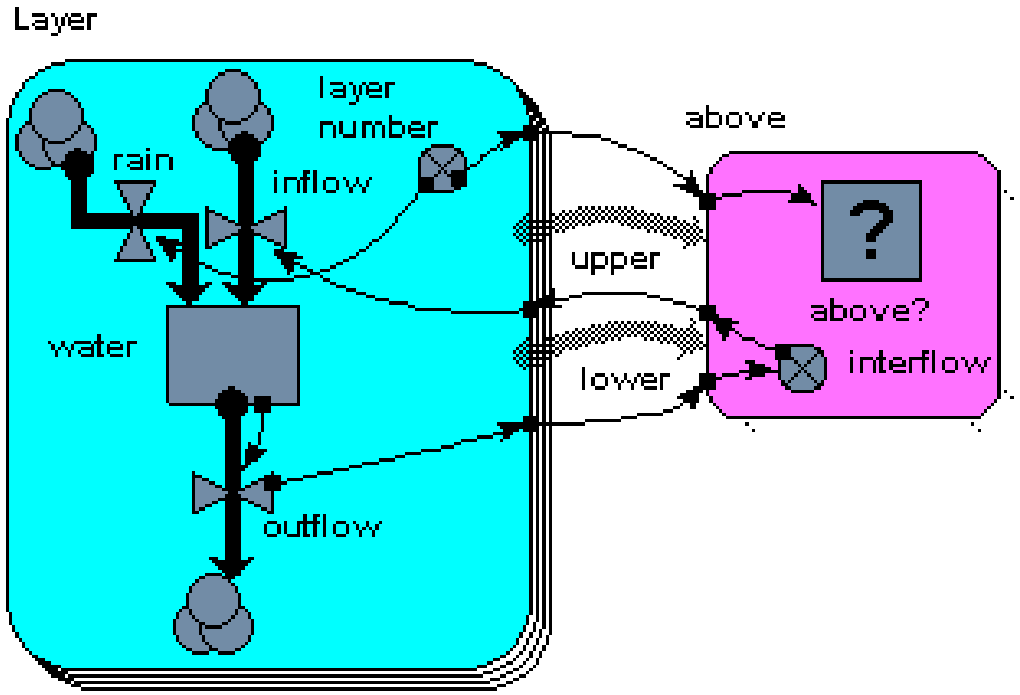


Water flow between soil layers: step 7

Add in the rain flow to the top layer




```
rain = if layer_number==1 and
       time(1)<10 then 10 else 0
```



Example 3: Land-use change

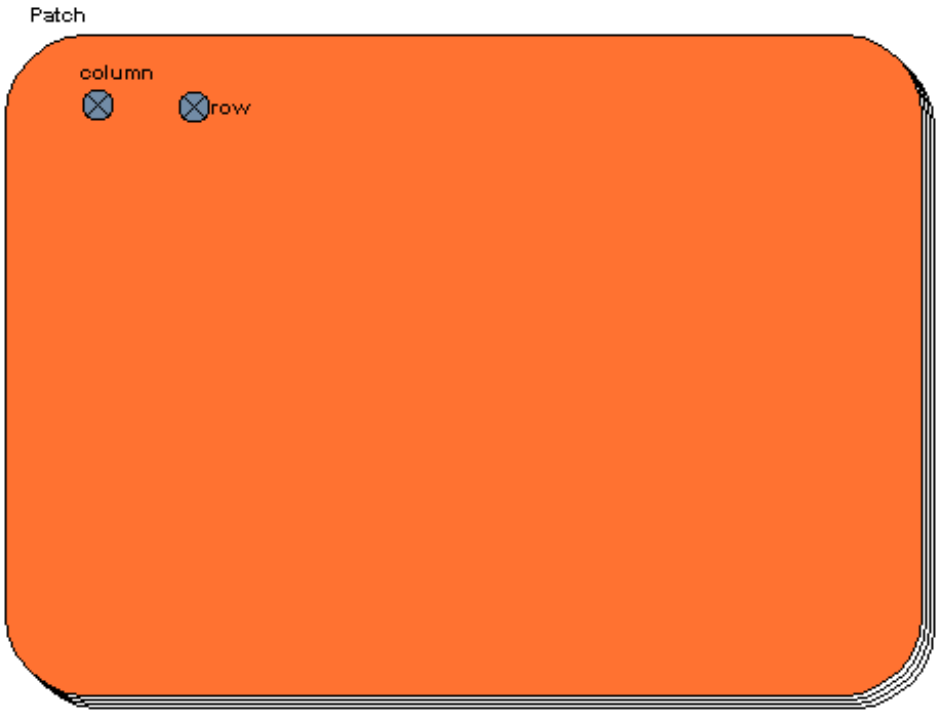
Problem: to simulate land-use change at the forest margin

- Illustrates the use of conditional submodels: the forest and crop submodels may or may not ‘exist’ in a patch
- Implements the concept of the ‘next-to’ association between patches
- Illustrates the use of Simile for spatial modelling

Land-use change: step 1

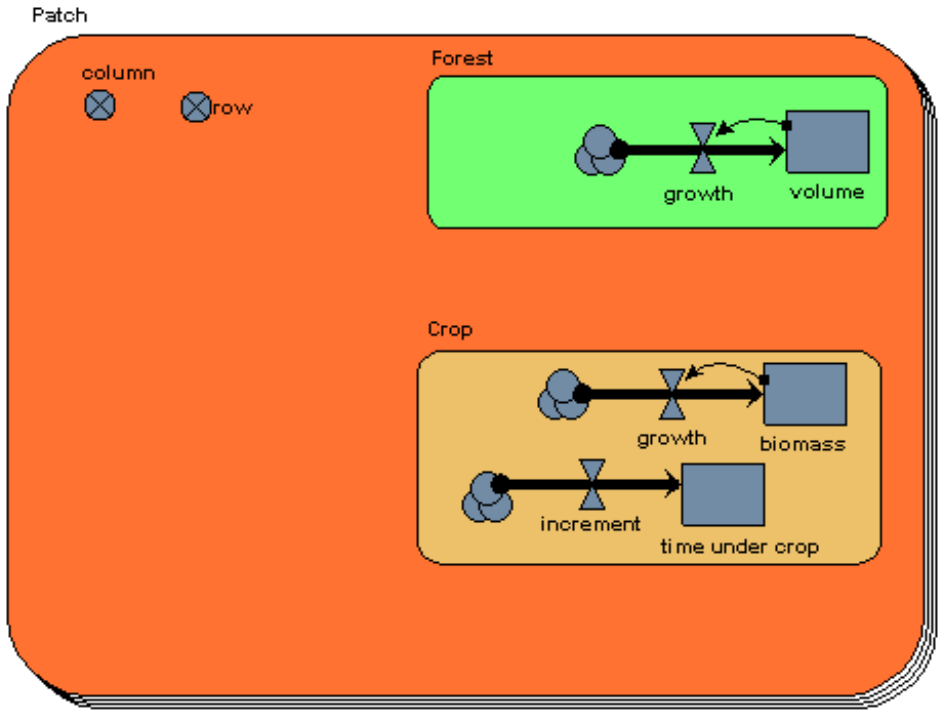
Create a multiple instance submodel so as each plot can have a different position

```
column = floor((index(1)-1)/10)+1
row = fmod(index(1)-1,10)+1
```



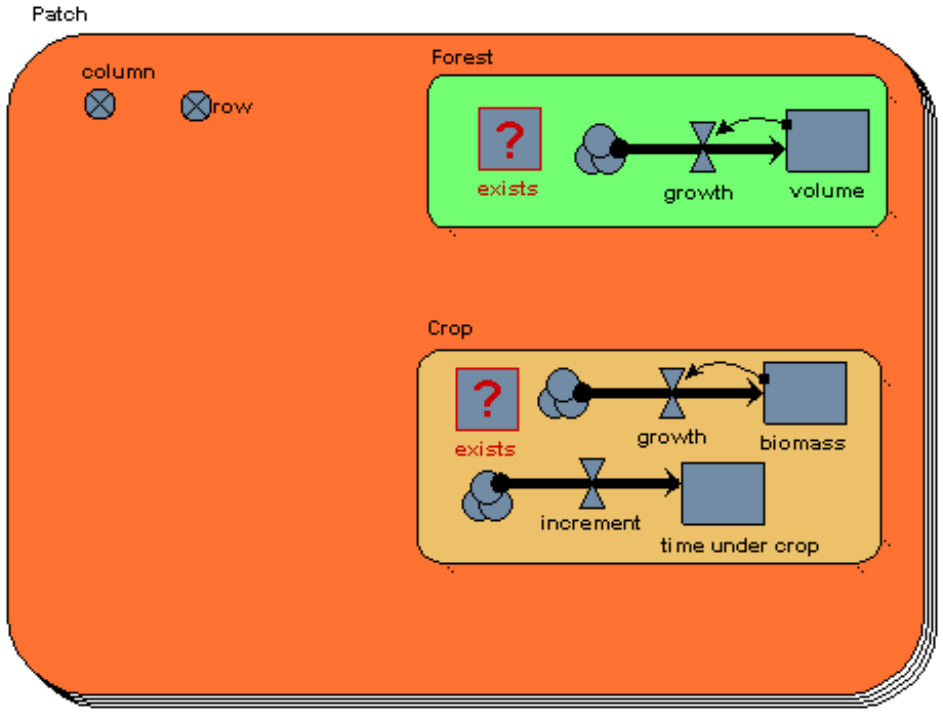
Land-use change: step 2

Specify the types of land that will be used



Land-use change: step 3

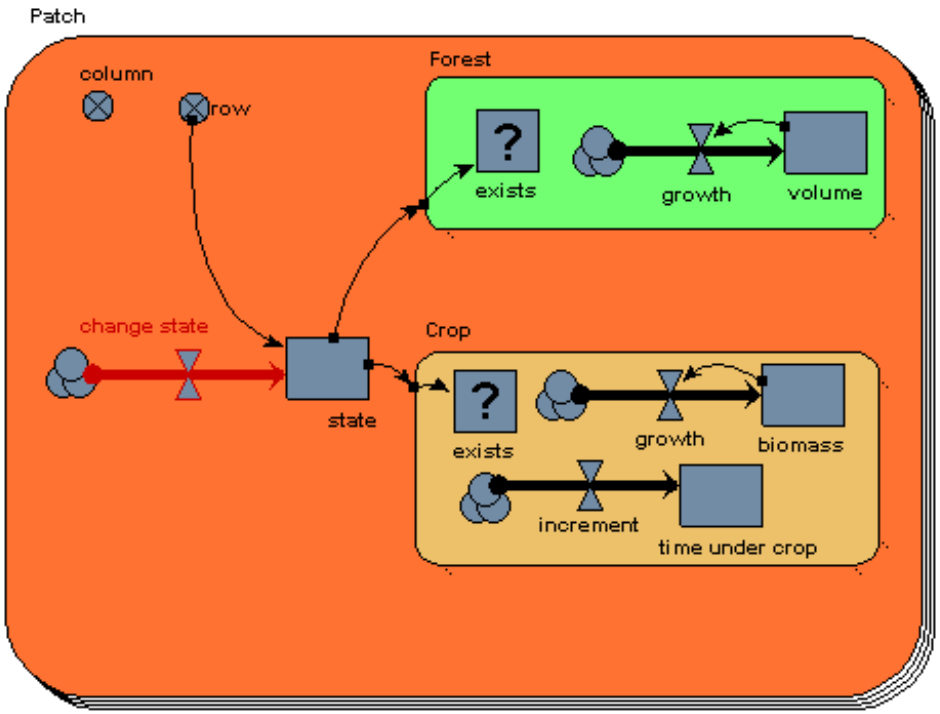
Add an existence condition to the 2 land types



Land-use change: step 4

Create a compartment that defines the state of a particular plot

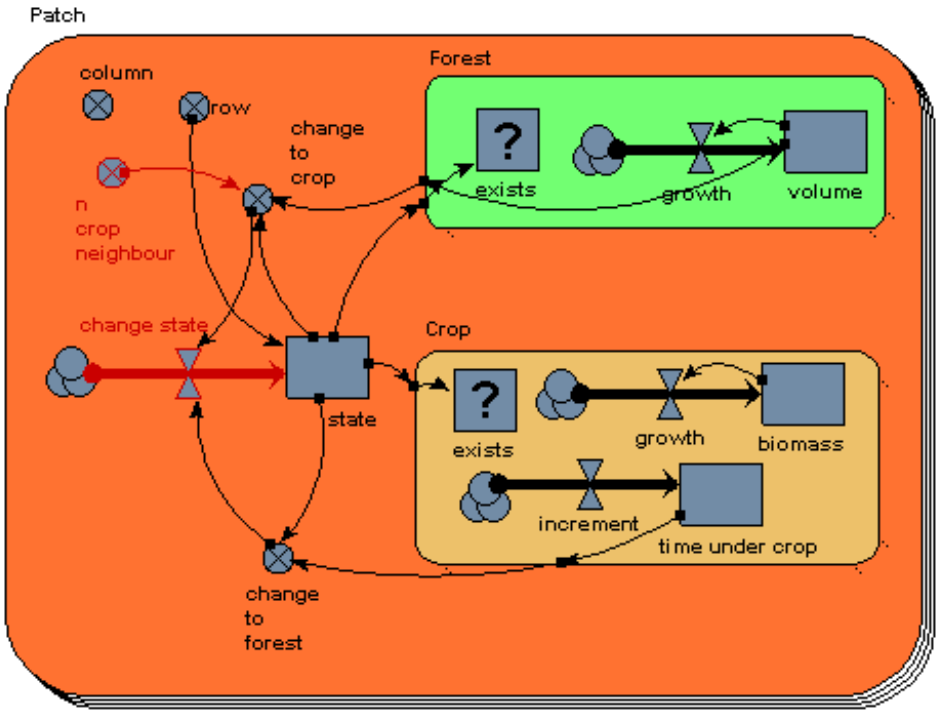
```
state = if row < 3 then 2 else 1
exists/Forest = state == 1
exists/Crop = state == 2
```



Land-use change: step 5

Set the conditions for a change of state

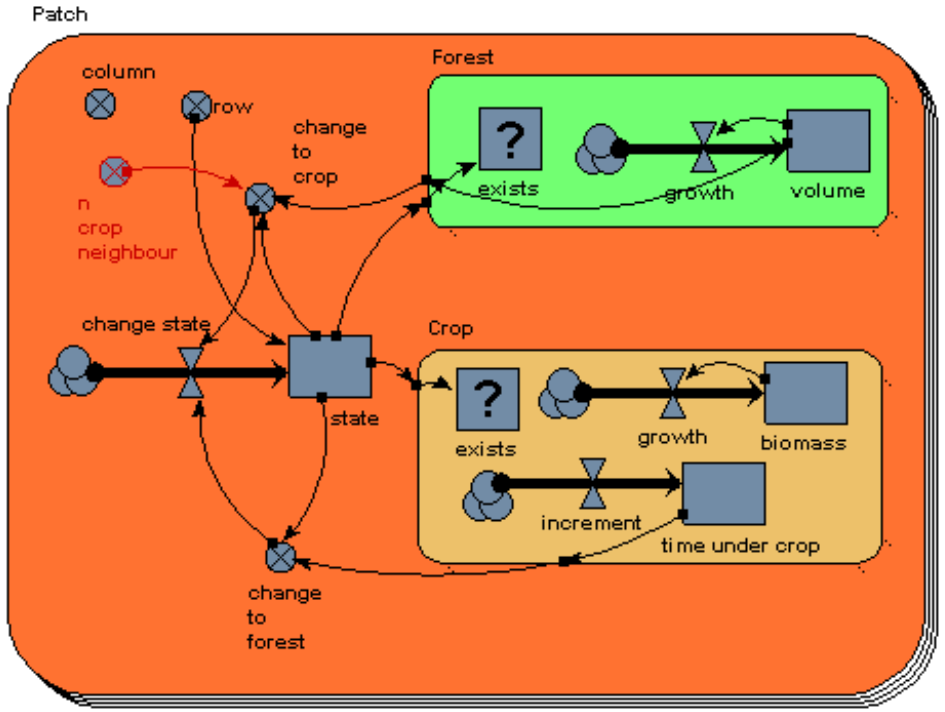
```
change_to_crop = if state==1 and
    sum({volume})>rand_var(250,400) and
    n_crop_neighbour > 1.9 then 1 else 0
change to forest = if state==2 and
    sum({time_under crop})>100 then 1 else 0
```



Land-use change: step 6

Make a way of changing the state of a plot

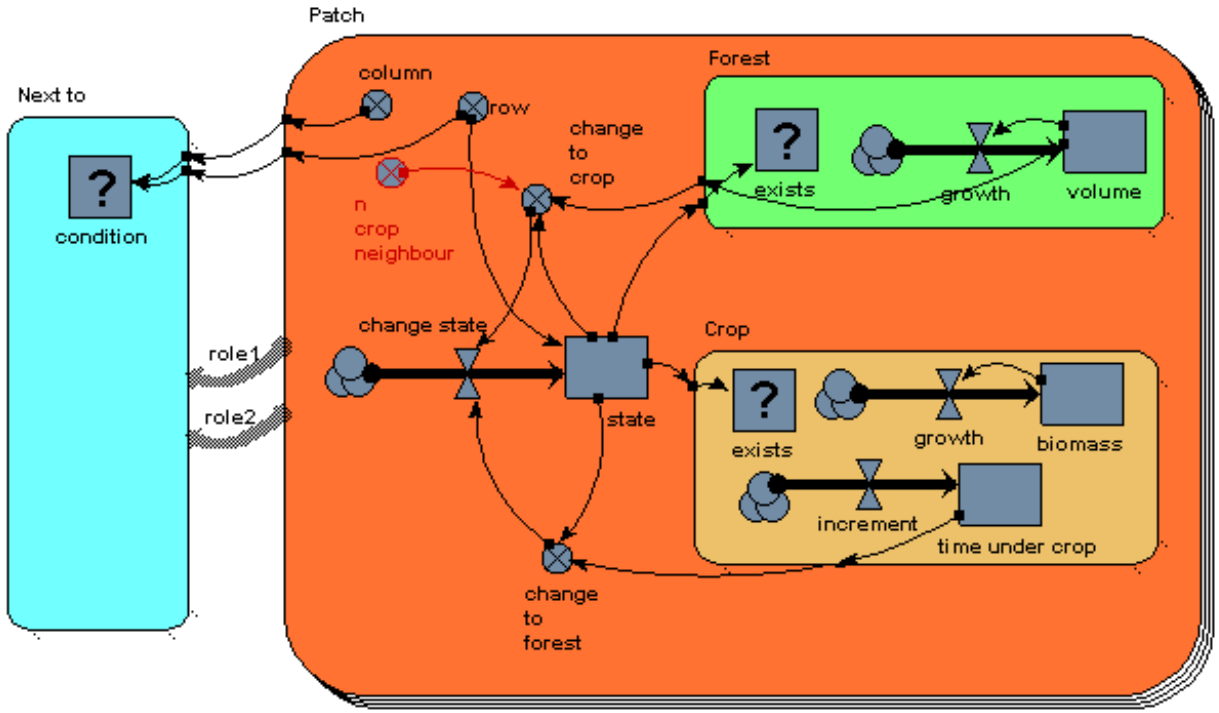
```
change_state = if change_to_crop == 1 then 1/dt(1)
   elseif change_to_forest == 1 then -1/dt(1) else 0
```



Land-use change: step 7

Add an association submodel that exists if any 2 plots of land are next to each other

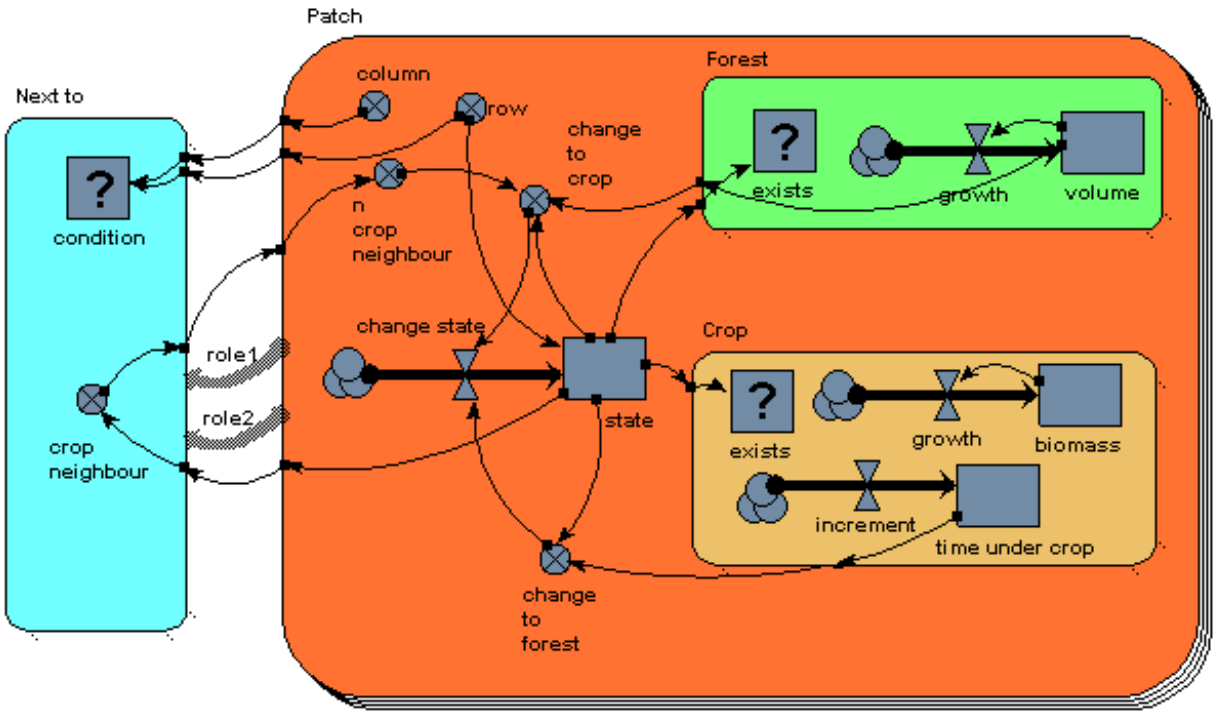
```
condition = not(column == column_0 and row == row_0)
    and abs(column - column_0) <1.5 and
    abs(row - row_0) <1.5
```



Land-use change: step 8

Complete the model by working out how many crop neighbours a particular plot has.

```
crop_neighbour = if state == 2 then 1 else 0
n_crop_neighbour = sum({crop neighbour_0})
```



Land-use change: results