Module 7

SELES – building models

Case Study – Hands On

Andrew Fall

Landscape Systems Ecologist

Gowlland Technologies Ltd.

April 2024

Module 7 Objectives

What you can expect to learn from this module:

- Gain experience in building models in SELES
  - Incrementally increasing complexity
  - Simple habitat suitability index, fire, logging and roading models

➢ See SELES User Documentation: Part 3

Case Study

Let's assume that we have been tasked with building a habitat suitability index (HSI) model for Northern Goshawk in a given study area that can be projected over time subject to the following processes:

- Wildfire
- Logging
- Road building

- ➢ The focus is on the HSI and on outcomes due to interactions among the modelled processes (so the process models can be relatively simpe)

- ➢ This module is mostly hands-on
- ➢ All layers and models for the case study are included in the tutorial model files download in the CaseStudy folder

Case Study
description

Area: ~418,000 ha (mostly forested)

Topography: mountainous in south west; hilly to gentle in remainder

Mostly public land

Exercise: take a look at some of the layers in the CaseStudy\gisDat\grids folder (see the data dictionary included with the case study)

- *It is always a good idea to view and check every spatial input*

Case Study
Conceptual Model

Case Study

Northern Goshawk HSI

Suppose we are provided with the following conceptual model for a Northern Goshawk nest area HSI:

- *HSI* (0-1): = *species factor* * *crown closure factor* * *age & height factor* * *edge factor*, where
- *species factor* = weights by leading species:

➢ preference for AT, PL and SW

Species		Weight
Name	Code	
Cottonwood	AC	0.5
Trembling Aspen	AT	1.0
Subalpine Fir	BL	0.6
Mountain Hemlock	HM	0.5
Lodgepole Pine	PL	1.0
White Spruce	SW	0.8

➢ Note: this is a hypothetical model, and is not meant to represent current knowledge about nest area habitat requirements for Northe Goshawk in B.C.

Case Study

Northern Goshawk HSI

- *crown closure factor* = weights by crown closure in deciles (with range from 0 or totally open, to 10 or 100% closure):
  - ➢ Preference for partially open stands, but avoidance of very open stands

Crown closure class	Weight
≤ 2	0
3	0.3
4	0.6
5-7	1
≥ 8	0.8

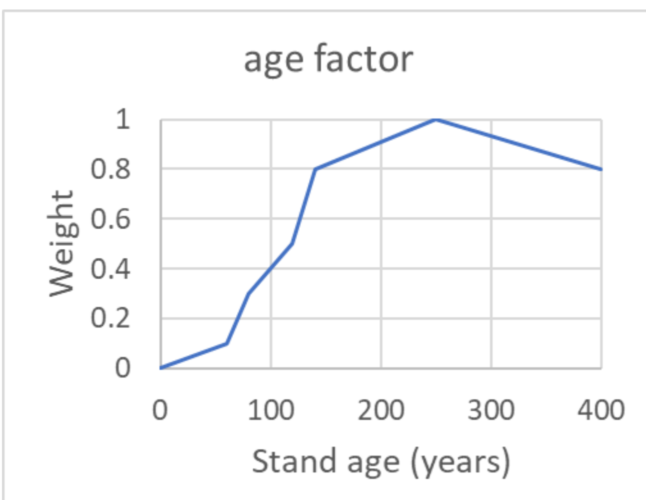
Case Study

Northern Goshawk HSI

- *age & height factor* = average of *age factor* and *height factor*, where:

- *age factor* = weights by age class
  - ➢ Preference for mature and old stands, avoidance of young and immature stands
  - ➢ Approximately logistic shape, if interpolate except at older ages

Age range (years)	Weight
≤ 60	0
61 – 80	0.1
81 – 100	0.3
101 – 120	0.5
121 – 140	0.8
141 – 250	1.0
≥ 250	0.8

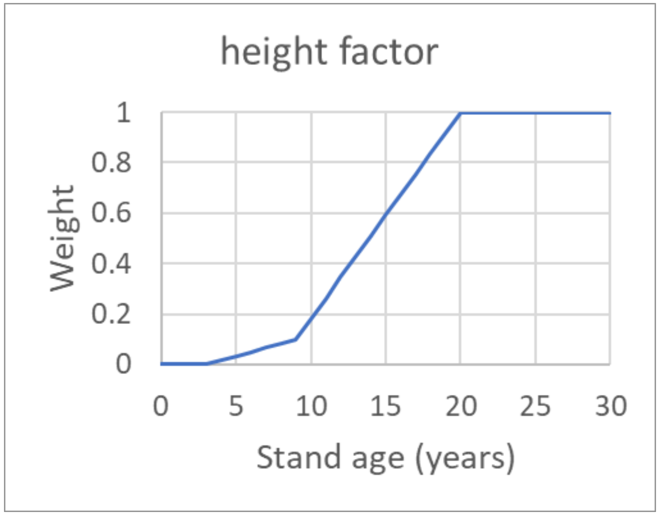


Case Study

Northern Goshawk HSI

- *height factor* = weights by height (in metres)
  - Preference increasing with stand height, avoidance of short stands
  - Approximately logistic shape
    (could be fitted to a continuous function)

Height range	Weight
≤ 3 m	0
3-9 m	(height-3) * 0.016667
9 - 20 m	0.1 + (height-9) * 0.0818182
> 20 m	1.0



Case Study
Northern Goshawk HSI

- *edge factor* = weight related to presence of hard edges, where

Hard edge	Weight
yes	0.7
no	1.0

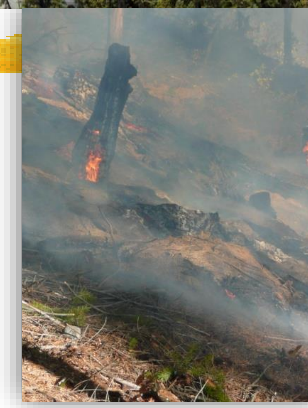
- A hard edge is defined as defined as forest that is at least 10 m tall and that has at least one adjacent neighbour that is either non-forest or < 20 m tall *and* more than 10m shorter than the focal cell
  - ➢ The 25% criteria usually this means 2 or more neighbours that meet the criteria
  - ➢ Adjacent neighbours must share a boundary of at least 100 m (i.e. just the cardinal neighbours, not diagonals)
  - ➢ Reduced preference at hard edges

Case Study
Succession

Objective: model stand-scale success with the following characteristics

- *Focus on stand aging*
- *Consider species changes at a later time*
- *Assume that leading species may change, but that there is no afforestation or deforestation (i.e. areas with a leading species will continue to have a leading species)*

Case Study
Wildfire



Objective: model a fire regime with the following characteristics

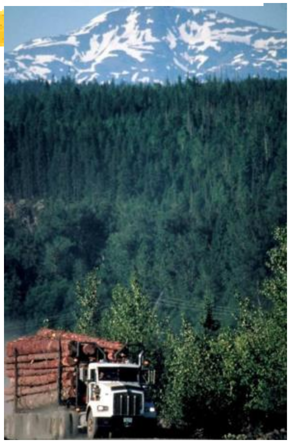
- *Parameters*
  - *Mean Number of Fires per yYar (exponential distribution)*
  - *Mean Fire Size (exponential distribution)*
  - ➢ *Fire Cycle (rotation) = Forest Size / (Mean Number of Fires per Year * Mean Fire Size)*
  - ➢ *Top-down fire model (disturbance rate as an input, not an emergent output)*

- Ignitions:
  - Highest on southwest facing slopes and lowest on northeast facing slopes

- Spread
  - Preference to spread highest on southwest facing slopes and lowest on northeast facing slopes
  - Preference to spread higher uphill than downhill

- Options to consider: ignition/spread preference by species, rate of spread vs. preference, fire shape controls

Case Study
Logging



Objective: model timber harvesting with the following characteristics

- *Parameters*
  - *Mean Number of Fires per yYar (exponential distribution)*
  - *Mean Fire Size (exponential distribution)*
  - *Fire Cycle (rotation) = Forest Size / (Mean Number of Fires per Year * Mean Fire Size)*
  - *Top-down fire model (disturbance rate as an input, not an emergent output)*

- Ignitions:
  - Highest on southwest facing slopes and lowest on northeast facing slopes

- Spread
  - Preference to spread highest on southwest facing slopes and lowest on northeast facing slopes
  - Preference to spread higher uphill than downhill

- Options to consider: ignition/spread preference by species, rate of spread vs. preference, fire shape controls

Version 1:
Basic Nest Area Habitat Suitability Index
(v1_HSI)

Case Study Model version 1

basic habitat suitability index (HSI)

Start with a simple model

- It is always a good idea to get a simple model running first, then incrementally add complexity, and ensure the model runs after each step

The case study provides a simple start point: version 1

Open and read the version 1 files (in LSEditor), starting with the scenario script:

- Scenarios\BaseScenarios_v1.scn
  - Creates two script variables ($modelVersion$ used to load the .sel file, and $gisData$ used for the path to the input GeoTiff files)
  - Opens three layers (StudyArea, Spp1 and StandAge)
  - Sets model dimensions using the StudyArea layer and loads the v1 CaseStudy.sel file
  - Runs a simulation for 1 time step

Case Study Model version 1

basic HSI: state space

Next open and read the version 1 state-space configuration (.sel) file:

- v1_HSI\CaseStudy.sel
    - Loads one landscape event (GoshawkHSI.lse)
    - Declares 3 spatial constants (for the loaded layers)
    - Declares 2 global constants:
        - CellWidth = CELL WIDTH(StudyArea), which is the width (in metres) of a grid cell
        - HaPerCell = (CellWidth^2)/10000, which is the area (in hectares) of a grid cell
        - ➢ These two constants are commonly used to ensure models are scalable to different resolutions and for reporting
    - Declares 2 spatial variables: NestAreaHSI with a range from 0 to 100, and NestAreaHSIClass with a range from 0 to 3
        - ➢ Note: if spatial variables are assigned an initial state, it is assume to be 0

Case Study Model version 1

basic HSI: new HSI event

Open and read the version 1 landscape event (.lse) file:

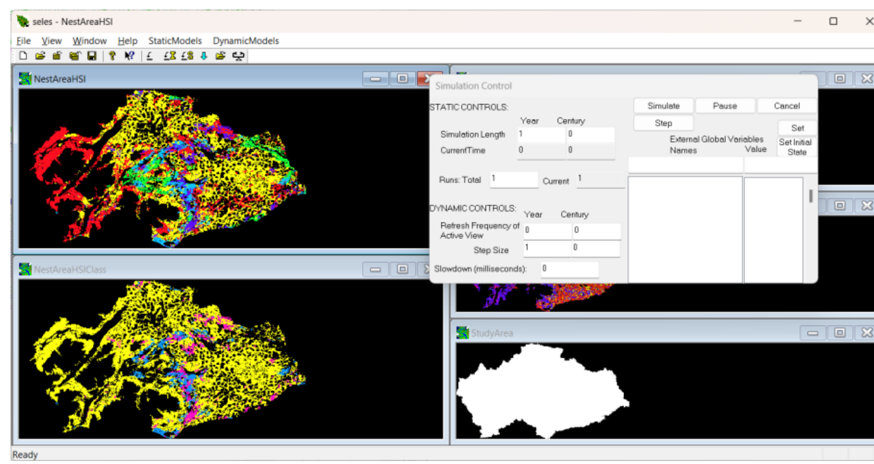
- v1_HSI\GoshawkHSI.lse
  - Definitions: include the relevant portion of the state space
  - First focus on the main expressions of the properties:
    - *ReturnTime = 0:* run at time 0 then stop (a *ReturnTime* of 0 on startup schedules the event instance for time 0, while after startup does not schedule anything)
      - That is, this event is static (does not run over time)
    - *EventLocation*: all cells in the study area that have a leading species (i.e. forested cells). The *NumClusters* and *ProbInit* properties are not defined, so initiate in all cells of the *EventLocation*.
    - *Transitions = TRUE*: occur and stop (because there are no spread properties)
  - Then focus on the state changes associated with properties
    - AgeRating: is a temporary variable assigned a weight from 0 to 1 based on the value of the StandAge layer in the current cell
    - NestAreaHSI: is assigned AgeRating multiplied by 100 (to scale from 0 to 100) and rounded to the nearest integer
    - NestAreaHSIClass: is assigned a classification of NestAreaHSI into quartiles (0-3)

Case Study Model version 1

basic HSI: simulation

Now run the model: open the version 1 scenario script in SELES:

- Scenarios\BaseScenario_v1.scn
  - This will run the model for one step, and produce the NestAreaHSI and NestAreaHSIClass output layers
  - This should look something like this:



Exercises:

a) Explore the grids (e.g. zoom/pan) to compare HSI result values with the input age layer

b) Calculate the average HSI value (e.g. use a Value Model)

c) Create a frequency histogram of the HSI and HSI class layers

d) Create a model report (DynamicModels menu: Model Report) and look at the result (selesModelReport.txt)

➢ There was no change of directory in the script, so the working director is Scenarios

Version 2:
Refined Nest Area Habitat Suitability Index
(v2_HSI2)

Case Study Model version 2
refined HSI

Incrementally add refinements and complexity

- Always make sure a model runs after adding significant complexity as errors tha become apparent are most likely due to the most recent set of changes
  (what constitutes "*significant*" will depend on your level of confidence)

Go back to the HSI model description and think about what factor you would add next.

Exercise: try to revise the version 1 model to add the elements for the full HSI

Then continue with the slides for an explanation of the steps from version 1 to version 2

Case Study Model version 2
refined HSI: script changes

Open and compare the version 1 and 2 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v1.scn and Scenarios\BaseScenarios_v2.scn
- The revised and new aspects of v2 are:
  - The script variable $modelVersion$ is set to v2_HSI2
  - Two additional layers are opened (StandHeight10 and CrownClosureClass)

- The units in the stand height layer are 1/10th of a metre. In most cases that need decimal values, it is more convenient to used a fixed precision integer representation (rather than a full floating point grid).
  - In this case, values represent metres to 1 decimal place (decimeter), which is done by multiplying heights in metres by 10 (e.g. a value o 245 represents 24.5 metres)
  - Care must be taken when reading values (to divide by the scaling factor, 10 in this case) and writing values (to multiply by the scaling factor)
  - Advantages include simpler control over level of precision, ability to display the grid more easily (because there are a finite number of potential values)

Case Study Model version 2

refined HSI: state space changes

Open and compare the version 1 and 2 state-space configuration (.sel) files:

- v1_HSI\CaseStudy.sel and v2_HSI2\CaseStudy.sel
- The revised and new aspects of v2 are:
  - Declares a legend: SppLegend = ..\gisData\grids\Spp1.tif
    - This extracts the legend from the Spp1.tif GeoTiff file and uses it to (a) define a global constant for each legend entry (e.g. PL will be created as a global constant with value 5) and (b) define a global constant SppLegend that can be use to ouput legend labels (instead of values) in landscape events (not used in the Case Study)
  - Declares 2 additional spatial constants (for the 2 new loaded layers)
  - Declares an additional global constant, created by loading a table input file:
    - SpeciesRatings = ..\inputFiles\GoshawkSpp.txt
    - ➢ Table values must be numbers, but can include global constants, which get converted to their values on loading (e.g. PL is converted to 5)
    - ➢ The file has two columns, one for species and one for rating (weight)
      - ➢ By default, the first column is interpreted as the row index, so SpeciesRatings is created as a 1-dimensional array with rows 0 to 6
  - Declares 1 additional spatial variable: HardEdge with a range from 0 to 1

Case Study Model version 2

refined HSI: HSI event changes

Open and compare the version 1 and 2 landscape event (.lse) files:

- v1_HSI\ GoshawkHSI.lse and v2_HSI2\ GoshawkHSI.lse
- The revised and new aspects of v2 are:
  - Definitions: include the new state space elements
    - Array global constants and variables have square brackets after their names
  - No changes to the main expressions of the properties
  - State changes in the *Transitions Consequent* expressions
    - SppRating: is a temporary variable assigned the weight in the SpeciesRating input table, indexed using the leading species layer Spp1 (simple lookup)
    - HeightRating: is a temporary variable assigned a weight from 0 to 1 based on the stand height h in the current cell
    - CCRating: is a temporary variable assigned a weight from 0 to 1 based on the value in the CrownClosureClass layer in the current cell
    - EdgeRating: is a temporary variable assigned a weight from 0 to 1 based on the whether or not the cell is considered a "*hard edge*" (see next slide)
    - NestAreaHSI: is assigned the product of SppRating, CCRating, mean of AgeRating and HeightRating, and EdgeRating multiplied by 100 (to scale from 0 to 100) and rounded to the nearest integer

Case Study Model version 2

refined HSI: identifying local "hard edges"

A hard edge is define in this model as:

Forest that is at least 10 m tall and that has at least one adjacent neighbour that is either non-forest or < 20 m tall and more than 10m shorter than the focal cell

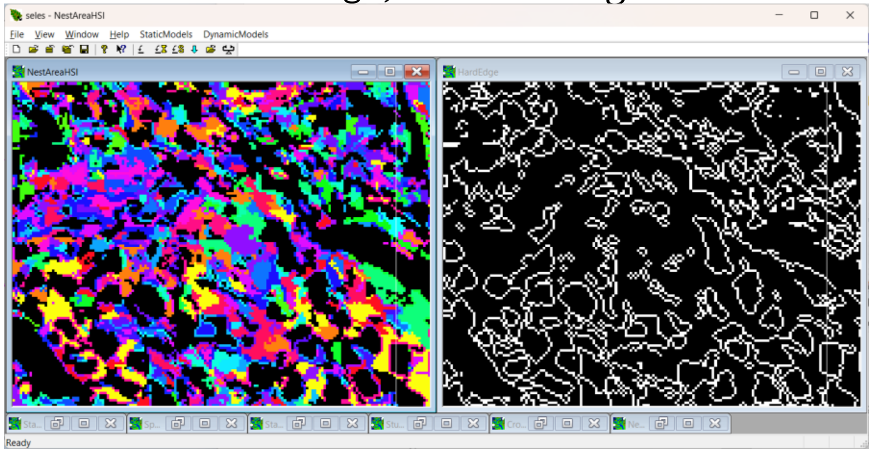
- To implement this requires assessing each of the neighbours of a focal cell
  - ➢ A simple method to do this in SELES is to use a "*REGION CENTRED*" expression to visit the adjacent neighbours of a focal cell

- The GowhawkHSI landscape implements this as follows
  - Neighbours are only assessed for focal cells with height $\geq$ 10 metres
    - ➢ Visiting neighbourhoods can be computationally expensive – keep it trim
  - *OVER REGION CENTRED(1,1)* defines the region of cells that are a minimum and maximum of 1 cell away from the focal cell (i.e. just the 4 cardinal neighbours)
  - The decision keeps within the study area (avoid edge effects)
  - The region expression creates a context for each neighbour – spatial variables refer to the value at the neighbour (but temporary variable `h` is still the height of the focal cell)
  - Increment temporary variable `nEdge` if the hard edge definition is met
  - After visiting the neighbours, set the focal cell as a hard edge if `nEdge` > 0

Case Study Model version 2
refined HSI: simulation

Now run the model: open the version 2 scenario script in SELES:

- Scenarios\BaseScenario_v2.scn
  - This will run the model for one step, and produce the NestAreaHSI, NestAreaHSIClass and HardEdge output layers
  - Minimizing all windows except NestAreaHSI and HardEdge, and zooming in a bit looks something like this:



Exercises:

a) Explore the grids (e.g. zoom/pan) to compare HSI result values with the input layers, and the result from v1

b) Calculate the average HSI value (e.g. use a Value Model)

c) Create a frequency histogram of the HSI and HSI class layers

Version 3:
Projecting the Nest Area Habitat Suitability Index
(v3_aging)

Case Study Model version 3
projecting the HSI

The attributes available for current conditions are a subset of those that can be projected over time

- Habitat models designed to be projected over time must be limited to attributes available via stand projection

For the Case Study, we will assume that the stand attribute that can be project is stand age

- Stand height projections are commonly available from stand models (via tables indexed by stand age, stand projectivity and other attributes)
- Crown closure and species succession is not available from all stand models
- We will assume that it is reasonable to assume for now that lead species is static, but that stand height and crown closure attributes are not available for projection

Exercise: try to revise the version 2 model to allow the HSI to be projected

Then continue with the slides for an explanation of the steps from v2 to v3

Case Study Model version 3

projecting the HSI: script changes

Open and compare the version 2 and 3 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v2.scn and Scenarios\BaseScenarios_v3.scn
- The revised and new aspects of v3 are:
  - The script variable $modelVersion$ is set to v3_aging
  - Two layers were removed (StandHeight10 and CrownClosureClass)
  - The layer names for two input layers were change (initialAge and initialSpp)
  - Some layers are minimized to focus the views on the dynamic layers
    - Minimize Static minimizes layers configured as *Spatial Constants* in the .sel file
    - Minimize Initial State minimizes layers used as the initial state for *Spatial Variables* in the .sel file
    - Tile re-tiles the remaining raster views
  - Run a simulation for 300 time steps (years) instead of just 1

Case Study Model version 3

projecting the HSI: state space changes

Open and compare the version 2 and 3 state-space configuration (.sel) files:

- v2_HSI2\CaseStudy.sel and v3_aging\CaseStudy.sel
- The revised and new aspects of v3 are:
  - Loads a second landscape event: Succession.lse
  - Declares an external script variable: $gisData$ for use in the .sel file
  - Removes 2 spatial constants (StandHeight10 and CrownClosureClass)
  - Changes 2 spatial constants to spatial variables (StandAge and Spp1), with initial state respectively initialStandAge and initialSpp1
  - Declares 2 additional global constants:
    - MaxStandAge = 450, used for the upper bound on the StandAge layer
    - MaxSpp = ROWS(SppLegend) - 1, used for the upper bound on the Spp layer
    - ➢ ROWS(SppLegend) is the number of rows in the SppLegend legend array
    - ➢ Since array indexing in SELES starts at 0, this is reduced by 1 to get the correct maximum value for potential leading species
  - Declares 1 additional global variable: BaseTimestep with initial value 1
  - Sets the default frequency for refreshing the display views to 1 time step

Case Study Model version 3

projecting the HSI: new succession event

Open and read the version 3 stand aging landscape event (.lse) file:

- v3_aging\Succession.lse
  - Definitions: include the relevant portion of the state space (not all state variables)
  - First focus on the main expressions of the properties:
    - *ReturnTime* = BaseTimestep*:* schedule and run each BaseTimestep years (default BaseTimestep = 1, which is an annual time step)
    - *EventLocation*: all cells in the study area that have a leading species (i.e. forested cells).
    - *Transitions = TRUE*: occur and stop (because there are no spread properties)
  - Then focus on the state changes associated with properties
    - StandAge is incrementd in the focal cell by BaseTimestep, but isn't allowed to exceed MaxStandAge

➢ This is a very simple "succession" model that only model stand aging (but could be expanded later to also modify leading species)

Case Study Model version 3
projecting the HSI: HSI event changes

Open and compare the version 2 and 3 landscape event (.lse) files:

- v2_HSI2\ GoshawkHSI.lse and v3_aging\ GoshawkHSI.lse
- The revised and new aspects of v3 are:
  - Definitions: include the new state space elements (and exclude the removed state)
    - Array global constants and variables have square brackets after their names
  - Changes to the main expressions of the properties:
    - *ReturnTime* = BaseTimestep: schedule and run each BaseTimestep years (default BaseTimestep = 1, which is an annual time step)
    - This changes the model from static to dynamic (although the HSI is a response)
    - *EventLocation*: the keyword "*STATIC*" causes the *EventLocation* to be calculated once at startup for efficiency (but the decision must actually be static)
  - State changes in the *Transitions Consequent* expressions
    - HeightRating and CCRating: removed
    - Hard edge calculation: same as in v2, except use stand age / 2 as a surrogate for stand height (assumes an average height growth of 0.5 m/year (a site index of 25)
    - NestAreaHSI: is assigned the product of SppRating, AgeRating and EdgeRating multiplied by 100 (to scale from 0 to 100) and rounded to the nearest integer

Case Study Model version 3

projecting the HSI: simulation

Now run the model: open the version 3 scenario script in SELES:

- Scenarios\BaseScenario_v3.scn
  - This will run the model for 300 years

Exercises:

a) Run for 1 step (with BaseTimstep set to 1) and compare HSI result with the result from v2
   - What is the likely cause of the largest discrepancies?
   - What might be done to deal with these discrepancies?

b) What improvements could be made to the HSI model, and how might they be implemented?

c) Create a model report (DynamicModels menu: Model Report) and look at the result (selesModelReport.txt)

- ➢ There was no change of directory in the script, so the working director is Scenarios
- ➢ Draw a dependency table for the main spatial variables (usually variables involved in the most important feedbacks of a model), and compare with the table on the next slide

Case Study Model version 3

state dependency table

	Spatial variables			
Landscape events	Species	Stand Age	Nest Area HSI	
GoshawkHSI	↲	↲	↱	
Succession (aging)	↲	↲↑		

Version 4:
Adding Basic Fire
(v4_fire)

Case Study Model version 4
basic wildfire

We will next incorporate wildfire into our model

- As with the HSI model, start with a simple fire model and add complexity once it is running

Case Study Model version 4
basic wildfire: script changes

Open and compare the version 3 and 4 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v3.scn and Scenarios\BaseScenarios_v4.scn
- The revised and new aspects of v4 are:
  - The script variable $modelVersion$ is set to v4_fire

Case Study Model version 4

basic wildfire: state space changes

Open and compare the version 3 and 4 state-space configuration (.sel) files:

- v3_aging\CaseStudy.sel and v4_fire\CaseStudy.sel
- The revised and new aspects of v4 are:
    - Loads a third landscape event: Fire.lse
    - Declares 1 additional global constants:
        - MaxTimeSinceDisturbance = 400
    - Declares 3 additional spatial variables:
        - OldForest: binary layer (range of 0 to 1)
        - Burnt: with a range from 0 to MaxTimeSinceDisturbance
        - Ignition: with a range from 0 to MaxTimeSinceDisturbance
    - Declares 3 additional global variables:
        - MeanFiresPerYear: with initial value 10
        - MeanFireSize: with initial value 100
        - AreaBurned: with initial value 0
        - ➢ Note: SELES does not know whether a global variable is a parameter, an output, for tracking, etc. (and use may vary among models), but documentation is useful to help read the intent of a variable

Case Study Model version 4
basic wildfire: succession event changes

Open and compare the version 3 and 4 stand aging landscape event (.lse) files:

- v3_aging\ Succession.lse and v4_fire\ Succession.lse
- The revised and new aspects of v4 are:
  - Definitions: include the new OldForest layer
  - State changes in the *InitialState Preliminary* expressions:
    - The main expression is the default value (1), so this property is included only for the *Preliminary* context
    - An *OVER REGION WHOLE MAP* expression is used to set up a spatial context for each cell in the study area that has a species
    - In each of those contexts, the OldForest binary layer is set to cells with age ≥ 250
    - This is done to set up a more the initial state for the OldForest layer, which is more complex than a single value
  - State changes in the *Transitions Consequent* expressions
    - The OldForest binary layer is updated based on the StandAge in that location and at that time

Case Study Model version 4
basic wildfire: new fire event

Open and read the version 4 fire landscape event (.lse) file:

- v4_fire\Fire.lse
  - Definitions: include the relevant portion of the state space plus one cluster variable FireExtent
  - First focus on the main expressions of the properties:
    - *ReturnTime:* BaseTimestep except on the first time (*Time EQ 0)* which uses 0.5. This is one way to ensure that events are sequenced in a time step as desired (in this case after the Succession event, which is right on the time step).
    - *EventLocation*: all cells in the study area that have a leading species (i.e. forested cells). Same as the other events.
    - *NumClusters*: selected from a negative exponential distribution with a mean of MeanFiresPerYear * BaseTimestep. The "FLOOR" function ensures that the result is an integer.
    - *ProbInit* = 1: All cells in the *EventLocation* have an equal likelihood of selection
    - *Transitions*: occur (continue) if the FireExtent cluster variable > 0, and the Burnt layer < MaxTimeSinceDisturbance (i.e. not burned this time step). FireExtent is set in *ProbInit* to a random number drawn from a negative exponential distribution with a mean MeanFireSize/HaPerCell (dividing by HaPerCell converts to cells)

Case Study Model version 4

basic wildfire: new fire event

- v4_fire\Fire.lse continued…
    - *SpreadTime = 0.001:* Spread an equal rate for all cells on the fire front. The small number ensures that all fires will go out before the end of the step.
    - *SpreadLocation*: cardinal neighbours in the study area.
    - *NumRecipients*: selected from a uniform distribution between 1 and 3 (rounded to the nearest integer). That is spread to 1, 2 or 3 of the neighbours.
    - Since *SpreadProb* is not specified, there is no preference among neighbours.
  - State changes in the *ReturnTime Consequent* expressions (start of each step)
    - The Burnt and Ignition layers are decremented using *OVER REGION* expressions
    - Reset the AreaBurned global variable to 0
  - State changes in the *ProbInit Consequent* expressions (initiating cells)
    - Set the FireExtent cluster variable to a random number drawn from a negative exponential distribution using the MeanFireSize parameter.
    - Set the Ignition layer to MaxTimeSinceDisturbance where and when a fire starts
  - State changes in the *Transitions Consequent* expressions (context of "burning")
    - Decrement the FireExtent cluster variable, increment the AreaBurned global variable, set StandAge to 0 and set the Burnt layer to MaxTimeSinceDisturbance.

Case Study Model version 4

basic wildfire: simulation

Now run the model: open the version 4 scenario script in SELES:

- Scenarios\BaseScenario_v4.scn
  - This will run the model for 300 years

Exercises:

a) Run for 1 step (with BaseTimstep set to 1) and compare HSI result with the result from v2
   - What is the likely cause of the largest discrepancies?
   - What might be done to deal with these discrepancies?

b) Create a model report (DynamicModels menu: Model Report) and look at the result (selesModelReport.txt)

- ➢ There was no change of directory in the script, so the working director is Scenarios
- ➢ Draw a dependency table for the main spatial variables (usually variables involved in the most important feedbacks of a model), and compare with the table on the next slide

Case Study Model version 4

state dependency table

Landscape events	Spatial variables: Species	Stand Age	Nest Area HSI	Burnt	
GoshawkHSI	↲	↲	↱		
Succession (aging)	↲	↲↱		↲↱	
Fire	↲	↱			

Version 5:
Refining the Fire Model
(v5_fire2)

Case Study Model version 5
refined top-down wildfire

We will next incorporate wildfire into our model

- As with the HSI model, start with a simple model and add complexity once it is running

Incrementally add refinements and complexity

Go back to the fire model description and think about what factor you would add next. The model describes a top-down approach to fire modelling (i.e. expected area to burn each time period is an input not an output)

Exercise: review the version 4 model and think about how it might be refined to model preference spread uphill and on southwest facing slopes

Case Study Model version 5

refined wildfire: script changes

Open and compare the version 4 and 5 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v4.scn and Scenarios\BaseScenarios_v5.scn
- The revised and new aspects of v5 are:
  - The script variable $modelVersion$ is set to v5_fire2
  - Two additional layers are opened (Elevation and Aspect)

Case Study Model version 5
refined wildfire: state space changes

Open and compare the version 4 and 5 state-space configuration (.sel) files:

- v4_fire\CaseStudy.sel and v5_fire2\CaseStudy.sel
- The revised and new aspects of v5 are:
  - Declares 2 additional spatial constants (for the 2 new loaded layers)
  - Removes 1 spatial variable (Ignition)

➢ When developing a model, it is just as important to consider what can be removed as what can be added

Case Study Model version 5

refined wildfire: fire event changes

Open and compare the version 4 and 5 fire landscape event (.lse) files:

- v4_fire\ Fire.lse and v5_fire2\ Fire.lse
- The revised and new aspects of v5 are:
  - Definitions: include the new Elevation and Aspect layers, and removes the Ignition layer
  - *ProbInit* property:
    - Relative probability of selection is a function of aspect that is highest (1.0) on southwest-facing aspects (225 degrees) and lowest (0.5) on northeast-facing aspects (45 degrees)
  - *NumRecipients* property:
    - Revision to the random selection of the number of neighbours to which to spread (this controls the shape of the fire patches)
    - Selection is from a normal distribution with a mean of 2 and a standard deviation of 1 (with a min of 1 and max of 8)
  - *SpreadProb* property:
    - Relative probability of selecting a neighbour for spread is a function of aspect and elevation (with factors combed as a geometric mean)
    - The aspect factor is the same as in *ProbInit*
    - The elevation factor compared the elevation of the potential recipient with that of

Case Study Model version 5

refined wildfire: simulation

Now run the model: open the version 5 scenario script in SELES:

- Scenarios\BaseScenario_v5.scn
  - This will run the model for 300 years

Exercises:

a) What are some different spreading options that can be used to produce complex shapes (i.e. that have complex final perimeters and unburned patches within the perimeter)?

b) Read and run the top-down and bottom-up fire models in models\SimpleFireModel, and identify the key differences between all three fire models.

c) What improvements could be made to the fire model, and how might they be implemented?

Version 6:
Adding Simple Timber Harvesting
(v6_logging)

Case Study Model version 6

simple logging

We will next incorporate timber harvesting into our model

- As with the HSI model, start with a simple logging model and add complexity once it is running

There is a often a fair amount in common between landscape-scale natural disturbance and timber harvesting models. The main differences are often in the level of control applied to timber harvesting models.

Case Study Model version 6

simple logging: script changes

Open and compare the version 5 and 6 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v5.scn and Scenarios\BaseScenarios_v6.scn
- The revised and new aspects of v6 are:
  - The script variable $modelVersion$ is set to v6_logging
  - One additional layers is opened (ForestStatus). This is layer identifies which forested cells are potentially available for logging (a simple timber harvesting land base)
  - Four parameter global variables are set in the script (AreaToCurPerYear, MinHarvestAge, BlockSizeMin and BlockSizeMax)
  - The script changes the working directory after loading the model (.sel file) but before running it (note use of the $modelVersion$ script variable for the folder of the model version to load, as well as the output folder
    - ➢ cwd ..\oOutput\$modelVersion$ minimizes layers configured as *Spatial Constants* in the .sel file
    - ➢ This model will introduce some output files, so it is good practice to move to an appropriately named folder before running a simulation

Case Study Model version 6

simple logging: state space changes

Open and compare the version 5 and 6 state-space configuration (.sel) files:

- v5_fire2\CaseStudy.sel and v6_logging\CaseStudy.sel
- The revised and new aspects of v6 are:
  - Loads two new landscape events: Logging.lse and ReportResults.lse
  - Declares 1 new legend: ForestStatusLegend from the ForestStatus.tif GeoTiff (this includes legend entries NonProductive, ProductiveExcluded and Productive)
  - Declares 1 additional spatial constant: ForestStatus
  - Declares 1 additional global constant: MaxBlockId = 1000
  - Declares 2 additional spatial variables:
    - Logged: with a range from 0 to MaxTimeSinceDisturbance
    - BlockId: with a range from 0 to MaxBlockId
  - Declares 8 additional global variables:
    - Logging parameters: AreaToCutPerYear, MinHarvestAge, GreenupYears, BlockSizeMin and BlockSizeMax
    - Tracking variables: AreaHarvested, TotalHSIbyClass and AreaOldForest

Case Study Model version 6
simple logging: succession event changes

Open and compare the version 5 and 6 stand aging landscape event (.lse) files:

- v5_fire2\ Succession.lse and v6_logging\ Succession.lse
- The revised and new aspects of v6 are:
  - Definitions: include the new AreaOldForest global variable
  - In the *ReturnTime Consequent* expressions:
    - Reset AreaOldForest at the start of each step
  - In the *Transitions Consequent* expressions
    - Increment AreaOldForest for cells that are at least 250 years old

Case Study Model version 6

simple logging: HSI event changes

Open and compare the version 5 and 6 Nest Area HIS landscape event (.lse) files:

- v5_fire2\ GoshawkHSI.lse and v6_logging\ GoshawkHSI.lse
- The revised and new aspects of v6 are:
  - Definitions: include the new TotAreabyClass global array variable
  - In the *ReturnTime Consequent* expressions:
    - Reset TotAreaByClass at the start of each step
  - In the *Transitions Consequent* expressions
    - Increment TotAreaByClass by the cell size for the HIS class for the current cell

Case Study Model version 6

simple logging: new logging event

Open and read the version 6 timber harvesting landscape event (.lse) file:

- v6_logging\Logging.lse
  - Definitions: include the relevant portion of the state space plus one local variable (currBlockId), one event variable (HarvestTarget) and one cluster variable (BlockExtent)
  - First focus on the main expressions of the properties (and dependent expressions):
    - *InitialState = 1*: default (single instance on start up)
    - *ReturnTime:* BaseTimestep except on the first time (*Time EQ 0)* which uses 0.7 to sequence this event within each time step after fire
    - *EventLocation*: all cells in the study area that have a Productive ForestStatus
    - *NumClusters = WHILE HarvestTarget > 0*: continues to initiate new clusters (blocks) as long as the condition holds. *HarvestTarget*
    - *ProbInit* = p: Selects cells from the *EventLocation* with relative probability p, calculated as a product based StandAge and neighbours (buffer):
      - The 1st factor tests if StandAge meets the minimum age threshold
      - The 2nd factor tests if the cell is adjacent to any cells below GreenupYears (using an *OVER REGION CENTRED* expression to visit the 8 adjacent cells)
      - The 3rd factor increases with StandAge (from 1 at MinHarvestAge)
      - The ORDERED keyword is used to select block initiations in order of p

Case Study Model version 6
simple logging: new logging event

- v6_logging\Logging.lse continued…
  - *Transitions = isAvailable*: occur (log cell) only if the isAvailable temporary variable is TRUE, calculated as a series of conditions (all of which must hold)
    - BlockExtent (cluster variable) and HarvestTarget (event variable) must both be positive
    - StandAge must be at least the minimum harvest age threshold
    - The cell must not be adjacent to any cells below GreenupYears (other than those in the same block)
  - *SpreadTime =-1:* Spread *immediately* (i.e. before initiating any subsequent blocks), at an equal rate for all cells on the block front. Immediate spread is any spread time < 0, and causes cluster initiation to pause until a cluster finished spreading (the default is to spread clusters in parallel). This is commonly used when *NumClusters* uses the *WHILE* option.
  - *SpreadLocation*: cardinal neighbours in the productive forest.
  - Since neither *NumRecipients nor SpreadProb* are specified, spread will be to all neighbours (resulting in simple blocks shapes).

Case Study Model version 6

simple logging: new logging event

- v6_logging\Logging.lse continued…
  - State changes in the *InitialState Preliminary* expressions (on simulation startup)
    - Set the currBlockId local variable to 0 (this will increment for each new block)
  - State changes in the *ReturnTime Consequent* expressions (start of each step)
    - Reset the AreaHarvested global variable to 0
  - State changes in the *EventLocation Consequent* expressions (in productive forest cells)
    - Decrement the Logging spatial variable (decrement since it is the reverse of time since logging but don't allow to go below 0)
  - State changes in the *NumClusters Preliminary* expressions (non-spatial – before initiation)
    - Set the HarvestTarget cluster variable to the AreaToCutPerYear parameter times BaseTimestep
  - State changes in the *NumClusters Consequent* expressions (in *initiating* cells – same as the *ProbInit Consequent* expressions)
    - Set the BlockExtent cluster variable to a random number drawn from a uniform distribution between the MinBlockSize and MaxBlockSize parameters
    - Increment the currBlockId local variable (but cycle to 1 when MaxBlockId is reached)

Case Study Model version 6
simple logging: new logging event

- v6_logging\Logging.lse continued…
  - State changes in the *Transitions Consequent* expressions (context of “logging”)
    - Decrement the BlockExtent cluster variable and HarvestTarget event variable
    - Increment the AreaHarvested global variable
    - Set StandAge to 0
    - Set the Logged layer to MaxTimeSinceDisturbance
    - Set the BlockId layer to currBlockId

Case Study Model version 6
simple logging: new reporting event

Open and read the version 6 *report results* landscape event (.lse) file:

- v6_logging\ReportResults.lse
  - Definitions: include the relevant portion of the state space plus:
    - One *Output* variable ReportFile with output directed to the annualReport.txt file
  - First focus on the main expressions of the properties:
    - *ReturnTime:* BaseTimestep except on the first time (*Time EQ 0)* which uses 0.99 to sequence this event near the end of a time step (after all process models have run)
    - *NumClusters = 0*: no initiations
  - The only other expressions are in *ReturnTime Consequent* expressions:
    - Output a record to the ReportFile with the Run number (built-in variable), simulation time (built-in variable), area burned, area harvested, area in each HSI class and the area of old forest.

  - The purpose of this model is to output information to a file at the end of each step.
  - It does not make any state changes, and so assumes that the variables output have been set by other event processes.

Case Study Model version 6

simple logging: simulation

Now run the model: open the version 6 scenario script in SELES:

- Scenarios\BaseScenario_v6.scn
  - This will run the model for 300 years
  - View the output report (in annualReport.txt)

Exercises:

a) Try changing parameters (either on the Simulate dialog or in the scenario script) and run.
b) Try running without fire
c) What improvements could be made to logging, and how might they be implemented?
d) Create a model report (DynamicModels menu: Model Report) and look at the result (selesModelReport.txt)

- The script changed the working directory, so the state space report file will be in the output folder
- Draw a dependency table for the main spatial variables (usually variables involved in the most important feedbacks of a model), and compare with the table on the next slide

Case Study Model version 6

state dependency table

Landscape events	Spatial variables					
	Species	Stand Age	Nest Area HSI	Burnt	Logged	
GoshawkHSI	↲	↲	⤴			
Succession (aging)	↲	⤴				
Fire	↲	⤴		⤴		
Logging		⤴			⤴	

Version 7:
Adding Simple Road Building
(v7_roads)

Case Study Model version 7

simple roading

We will next incorporate road building and access limitations for logging into our model

- As with the HSI model, start with a simple road building model and add complexity once it is running

Roads are linear features, and special care must be taken when modelling linear features in grid-based models (especially when two linear features cross, such as roads crossing streams)

New modelled roads will be built as simple straight lines that connect to the input road network or to other modelled roads

The logging will be modified to respond to the road access

Case Study Model version 7

simple roading: script changes

Open and compare the version 6 and 7 scenario script files (in LSEditor):

- Scenarios\BaseScenarios_v6.scn and Scenarios\BaseScenarios_v7.scn
- The revised and new aspects of v7 are:
  - The script variable $modelVersion$ is set to v7_roads
  - Three additional layers are opened (initialRoadState, initialDist2Road and initialNearestRoadLoc) that identify current existing/planned future roads, the distance to existing roads and the cell location of the nearest road
  - Two parameter global variables are set in the script (MinNoEffectDist2Rd and MaxDist2Rd)

Case Study Model version 7

simple roading: state space changes

Open and compare the version 6 and 7 state-space configuration (.sel) files:

- v6_logging\CaseStudy.sel and v7_roads\CaseStudy.sel
- The revised and new aspects of v7 are:
  - Loads 1 new landscape event: Roading.lse
  - Declares 2 additional global constants:
    - MaxInitialDist2Road = MAX(initialDist2Road)
    - MaxHarvestableCells = NUM>2(ForestStatus), which counts the number of cells larger than x (2 in this case) – that is the potentially available productive cells
  - Declares 3 additional spatial variables:
    - RoadState: range 0 to 2 and initial state initialRoadState
    - Dist2Road: range -1 to MaxInitialDist2Rd and initial state initialDist2Road
    - NeareastRoadLoc: range 0 to NumCells and initial state initialNearestRoadLoc
  - Declares 8 additional global variables:
    - Parameters for road effects on logging: MinNoEffectDist2Rd and MaxDist2Rd
    - Tracking variables: BlkLandingLoc (a 1-dimensional array with indexes from 0 to NumHarvestableCells - 1) , nBlocks and KmRoadsBuilt

Case Study Model version 7

simple roading: new roading event

Open and read the version 6 road building landscape event (.lse) file:

- v7_roads\Roading.lse
  - Definitions: include the relevant portion of the state space plus one active cell variable (currNearestRoadLoc)
  - First focus on the main expressions of the properties (and dependent expressions):
    - *ReturnTime:* BaseTimestep except on the first time (*Time EQ 0)* which uses 0.8 to sequence this event within each time step after logging
    - *EventLocation*: uses a *REGION LOCATION LIST*, which is simply an array of cell locations and a variable to indicate the number – see the logging event changes
    - *ProbInit*: Uses the *ORDERED* keyword to process in sorted order of increasing distance from road (the sort order is highest value to lowest value of *ProbInit*)

➢ Initiate in each identified block landing, in order from closest to further from road

Case Study Model version 7
simple roading: new roading event

- v7_roads\ Roading.lse continued…
  - *Transitions*: occur (continue to update the distance surface) if the current distance to road is less than the recorded distance (Dist2Road) and also less than the MaxDist2Rd parameter (for efficiency: no need to update beyond this parameter)
    - The currDist2Road temporary variable is calculated as the straight-line distance from the current cell to the nearest road location (the DISTANCE function returns vales in cell units, so is multiplied by CellWidth)
  - *SpreadTime =-1:* Spread *immediately* (i.e. before initiating any subsequent road) at an equal rate for all cells on the expanding distance to roads front.
  - *SpreadLocation*: cardinal neighbours in the study area.
  - *SpreadProb*: default 1 (spread to all neigoburs in the *SpreadLocation*

- ➢ Spreading updates the distance to road surface after building each new segment
- ➢ It limits spread to MaxDist2Rd to avoid having to spread over the entire landscape
- ➢ This can be made more efficient (and more complex) by spreading from new road segments in parallel – landings are processed in order of increasing distance to road to allow new roads to join other new roads closer to the existing network

Case Study Model version 7

simple roading: new roading event

- v7_roads\ Roading.lse continued…
  - State changes in the *ReturnTime Consequent* expressions (start of each step)
    - Reset the KmRoadsBuilt global variable to 0
  - State changes in the *ProbInit Consequent* expressions (in *initiating* cells)
    - Set the currNearestRoadLoc active cell variable to the current location (which is assumed to be a block road landing site)
    - Increment the KmRoadsBuilt global variable by the straight-line distance from the landing cell (current cell) to the nearest existing road location (converted from cell units to km)
    - Use an OVER REGION VECTOR region expression to select the cells from the landing to the nearest road location
      - In each cell along that line, set the NearestRoadLoc layer to the location of that cell (i.e. assume every cell along the line is a road)

Case Study Model version 7

simple roading: new roading event

- v7_roads\ Roading.lse continued…
  - State changes in the *Transitions Consequent* expressions (context of "reducing distance to road")
    - Set the Dist2Road layer to the currDist2Road temporary variable
    - ➢ Temporary variables in *Preliminary* and *Consequent* contexts are separate, so currDist2Road must be recalculated (or declared as an active cell variable)
    - Set the NearestRoadLoc layer to the currNearestRoadLoc active cell variable
  - State changes in the *SpreadProb Consequent* expressions (after spreading out to a cell that is closer to the new road)
    - Note that active cell variables are new variables for the newly activated recipient
    - If the NearestRoadLoc layer has the value of the current cell, then this cell is on the new road segment (created by the OVER REGION VECTOR expression)
      - Set the currNearestRoadLoc active cell variable to the current location (because this cell has a road)
    - Otherwise, this cell is not on the road network
      - Set the currNearestRoadLoc active cell variable to value from the spreading cell (accesses in a recipient context using the SOURCE keyword)

Case Study Model version 7

simple roading: logging event changes

Open and compare the version 6 and 7 stand aging landscape event (.lse) files:

- v6_logging\ Logging.lse and v7_roads\ Logging.lse
- The revised and new aspects of v7 are:
  - Definitions:
    - include the new MaxDist2Rd and MinNoEffectDist2Rd global variables
    - include the new Dist2Road layer
    - declare a new RdDistEffectSlope local variable
    - Declare 1 new cluster variable: currBlockSize
  - In the *InitialState Preliminary* expressions:
    - Set the initial value of RdDistEffectSlope based on the MaxDist2Rd and MinNoEffectDist2Rd parameters
  - In the *ReturnTime Consequent* expressions:
    - Reset nBlocks at the start of each step
  - In the *NumClusters Consequent* expressions
    - Set currBlockSize to 0 (value when a block is initiated)

Case Study Model version 7

simple roading: logging event changes

- v7_roads\Logging.lse changes continued…
  - In the *ProbInit Preliminary* expressions
    - Added a 4th factor to test whether the cell is too far from roads (> MaxDist2Rd)
    - Added a 5th factor to decrease preference further from roads (linear decrease from 1 at MinNoEffectDist2Rd to 0 at MaxDist2Rd)
  - In the *Transitions Consequent* expressions (context of logging)
    - For the 1st cell of the block: set up to build a new road from the current cell to the nearest road (add the cell location to the BlkLandingLoc array and increment nBlocks)
    - Increment the currBlockSize cluster variable (used to identify if a cell is the 1st cell of a block, currBlockSize is 0, or a subsequent cell reached by spread)

Case Study Model version 7

simple roading: reporting event changes

Open and compare the version 6 and 7 reporting landscape event (.lse) files:

- v6_logging\ ReportResults.lse and v7_roads\ ReportResults.lse
- The revised and new aspects of v7 are:
  - Definitions: include the new KmRoadsBuilt global variable
  - In the *ReturnTime Consequent* expressions:
    - Include KmRoadsBuilt in the reported record

Case Study Model version 7

simple roading: simulation

Now run the model: open the version 7 scenario script in SELES:

- Scenarios\BaseScenario_v7.scn
  - This will run the model for 300 years
  - View the output report (in annualReport.txt)

Exercises:

a) Try changing parameters (either on the Simulate dialog or in the scenario script) and run.

b) Create a model report (DynamicModels menu: Model Report) and look at the result (selesModelReport.txt)

c) What improvements could be made to roading, and how might they be implemented?

- ➢ Note that even though the roading event does not directly change StandAge, it influences the location of logging (by modifying Dist2Road, which influences the location of logging) and hence influences StandAge indirectly
- ➢ Conversely, StandAge influences the location of logging, which modifies BlkLandingLoc, which affects where roads are placed

Case Study Model version 7

state dependency table

Landscape events	Spatial variables: Species	Stand Age	Nest Area HSI	Burnt	Logged	BlkLandingLoc (global var)	Dist2Road	NearestRoadLoc
GoshawkHSI	↵	↵	↵↑					
Succession (aging)	↵	↵↑						
Fire	↵	↑		↵↑				
Logging		↵↑			↵↑	↑	↵	
Roading						↵	↵↑	↵↑

Case Study Model
take-aways

- Very complex, emergent behaviour is possible by combining relatively simple landscape event process models

- A clear, written conceptual model is important to guide implementation, keep objectives in focus, and help awareness of the modelling gap
  - The conceptual model may evolve as model implementation progresses

- The main focus of any model should be on its underlying behaviour (how it navigates through space and time).
  - Once the correct contexts are established, specifying state changes is more straight-forward (even if some expressions an become complicated)

- Feedback between process sub-models is often primarily through changes to spatial state – dependency diagrams can help understand feedback loops

- Models can become complex and complicated quickly – build them incrementally, and focus on the most important aspects to include