Module 5

SELES – Reading Models

Understanding spatio-temporal state spaces and landscape events

Andrew Fall

Landscape Systems Ecologist

Gowlland Technologies Ltd.

April 2024

Module 5 Objectives

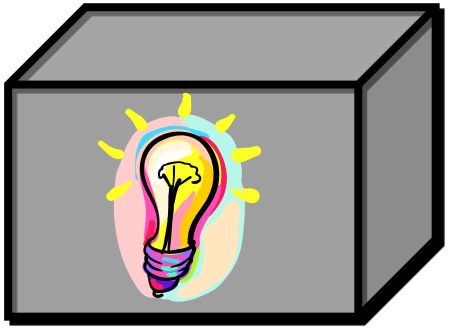
What you can expect to learn from this module:

- How to read and understand an existing SELES model
  - How to create and read a SELES state-space report
  - How to read a state-space configuration (.sel) file
  - How to read a landscape event (.lse) file
  - How to make basic changes to existing models

➢ See SELES User Documentation: Part 3

SELES modelling paradigm

- Understanding models: opening up Pandora's box
  - model state-space
  - spatio-temporal contexts
  - landscape events
  - base expressions
- Goal: make a black box transparent



State Space

- Set of all model variables
  + range of possible values

- Model report (DynamicModels menu)
  - create state-space report for a loaded model
  - list of all state shared state variables
    - » plus general causal links with events/agents

  - Useful for:
    - » understanding structure of a model
    - » debugging a model

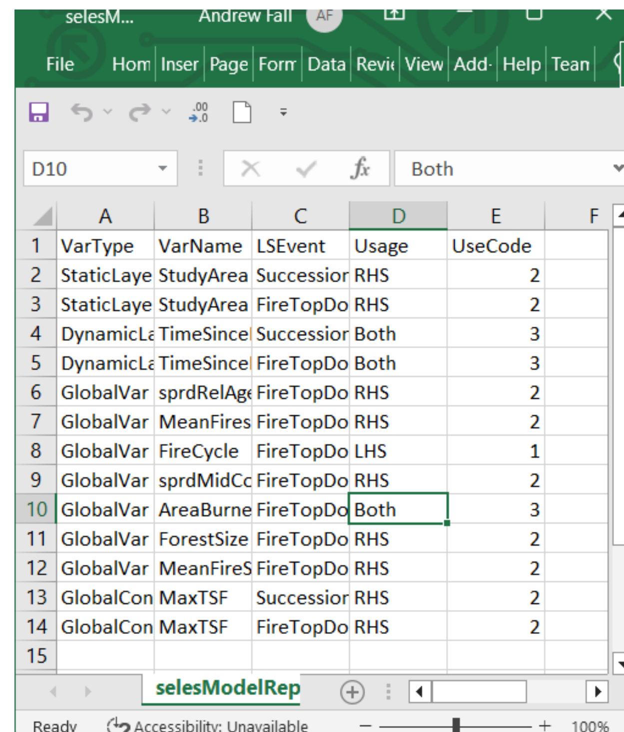
Hands-on

state space of the simple top-down fire model

- Start SELES and open the FireTopDown.scn scenario script
- Select DynamicModels menu: Model Report
  - ➢ Doesn't seem to do anything, but SELES output the model report in the current working directory (models\SimpleFireModel\Scenarios)

- In Excel open selesModelReport.txt and selesModelReport2.txt

➢ the latter should look like this:



	A	B	C	D	E
1	VarType	VarName	LSEvent	Usage	UseCode
2	StaticLaye	StudyArea	Successior	RHS	2
3	StaticLaye	StudyArea	FireTopDo	RHS	2
4	DynamicLa	TimeSince	Successior	Both	3
5	DynamicLa	TimeSince	FireTopDo	Both	3
6	GlobalVar	sprdRelAge	FireTopDo	RHS	2
7	GlobalVar	MeanFires	FireTopDo	RHS	2
8	GlobalVar	FireCycle	FireTopDo	LHS	1
9	GlobalVar	sprdMidCc	FireTopDo	RHS	2
10	GlobalVar	AreaBurne	FireTopDo	Both	3
11	GlobalVar	ForestSize	FireTopDo	RHS	2
12	GlobalVar	MeanFireS	FireTopDo	RHS	2
13	GlobalCon	MaxTSF	Successior	RHS	2
14	GlobalCon	MaxTSF	FireTopDo	RHS	2
15					

Hands-on

state space of the simple top-down fire model

These files show the following state-space info (in different formats):

- 2 landscape events: Succession.lse and FireTopDown.lse
- 1 static (spatial) layers: StudyArea
  - used on the right-hand side (RHS) of expressions in both landscape events (i.e. those events depend on StudyArea but don't change it)
- 1 dynamic layer: TimeSinceFire (initialized using initialTimeSinceFire)
  - used on the RHS and left-hand side (LHS) of expressions in both landscape events (i.e. those events depend on TimeSinceFire and modify it)
- 7 global variables:
  - Parameters are used only on the RHS: there 5 parameters to the FireTopDown.lse event (MeanFiresPerYear, MeanFireSize, ForestSize, sprdRelAgeExp, sprdMidComplexShp).
  - Tracking and output variables are modified on the LHS: there are two output variables from the FireTopDown.lse event (FireCycle, AreaBurned)
- 1 global constant: MaxTSF used by both events

Hands-on

dependency table of the simple top-down fire model

State-space info can be summarized as a table where arrows indicate if a process uses a variable (points to the event) and/or modifies it (points to the variable). This shows the main feedbacks in a model (in this case, the TimeSinceFire layer)

Process \ State	StudyArea	TimeSinceFire	MeanFiresPerYear	MeanFireSize	ForestSize	sprdRelAgeExp	sprdMidComplexShp	FireCycle	AreaBurned
Succession	←	← ↑							
FireTopDown	←	← ↑	←	←	←	←	←	↑	← ↑

Model Configuration Language (.sel files)

General

- Declarative: defines structural configuration
- Blocks (subsections):
  - can appear in any order
  - can appear more than once
- Case insensitive (except for variable and constant names)

Procedural vs. Declarative Languages

- Procedural languages state *how* to achieve a particular result
  - Advantages: possibly faster
  - Disadvantages: black boxes

- Declarative languages state the desired result
  - Classic example: logic.
  - Advantages: more transparent

Model Configuration (.sel) Files
guidelines for reading

- Keep in mind that a .sel file just declares the main state variables (but isn't a script of commands)
  - It is read and processed once when loaded by a scenario script (or when reloaded on the user interface)
- The order of declarations only matters if a declaration depends on another one that must precede it in the file
  - e.g. bounds of a spatial variable may depend on a global constant
- Think of each section as a set (e.g. the set of landscape events, the set of global constants, …)
- **Use LSEditor**

Model Configuration (.sel) Files
syntax

- Specifies landscape events to include
- Links variables in landscape events to global variables and rasters
- Sets up output of raster layers
- First line: Seles Model

Setting Time Unit

Set "meaning" of a single time unit and a meta-unit

(plus, optionally, default simulation length)

- Time in SELES is a real value, and spread rates and return times can be any positive increment (and may vary); but modellers attribute meaning to 1 time unit

Examples:

Time Units: Day Year 365.25 3652.5 // default length is 10 years

Time Units: Year Century 100 100 // default length is 100 years

Time Units: Step kiloStep 1000 100 // default length is 100 steps

Loading Landscape Events

Load landscape events

- Events are initialized at the start of a simulation in the order listed; thereafter depends on the event queue

Example:

Landscape Events:

Succession.lse

Logging.lse

Fire.lse

Global Constants and Variables

- single values, or 1 or 2 dimensional arrays
- can be specified directly or read from a text file
- single-value variables automatically show on user interface

Examples:

Global Constants:

MaxStandAge = 540

CellWidth = CELL WIDTH(Ecoregion) // in metres

HaPerCell = (CellWidth^2)/10000 // in hectares

Volume = Volume.txt // from a table

Global Constants and Variables

Examples:

Global Variables:

usePatchSizeDist = FALSE

GreenupYears = *Expr* // function of other variables and constants

NRL = 24700, 8500, 5300 // 1-dimensional are with 3 values

HarvestProfile[NumSppCodes] = 0 // 1-dimensional array

SumHarvest[NumUnits, 2] = 0 // 2-dimensional array

Constant Names

- Can appear in input files

- Special constants:

  MAX(viewName)

  MIN(viewName)

  CELL WITDH(viewName)

  ROWS(arrayName)

  COLS(arrayName)

Spatial Constants

– layers that do not normally change dynamically
– raster views must exist

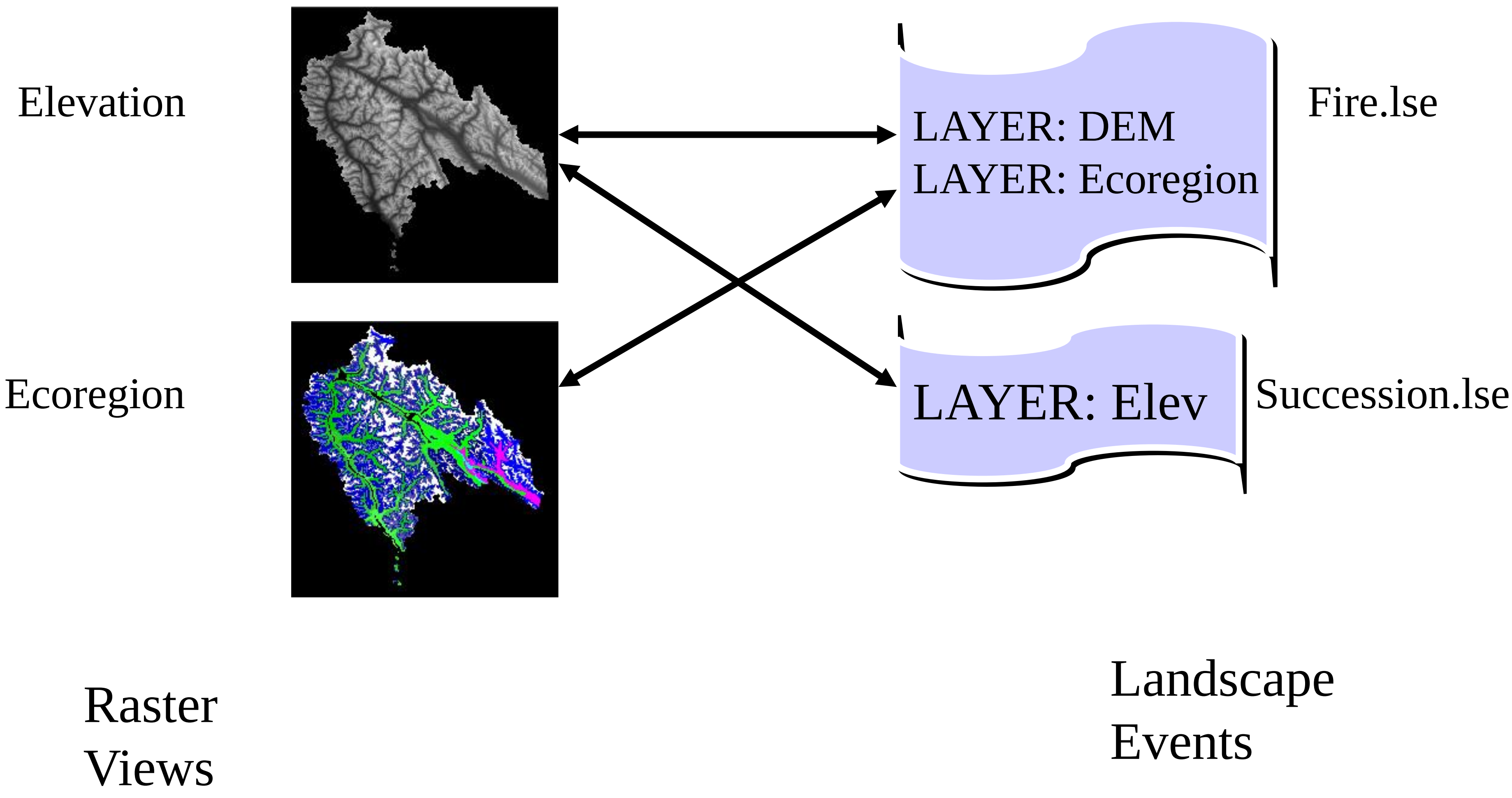
Example:

Spatial Constants:

Elev = Elevation

Ecoregion

Raster-Variable 1-to-Many Mapping



Spatial Variables

– If view doesn't exist, will be created

Examples:

Spatial Variables:

StandAge <- initialAge

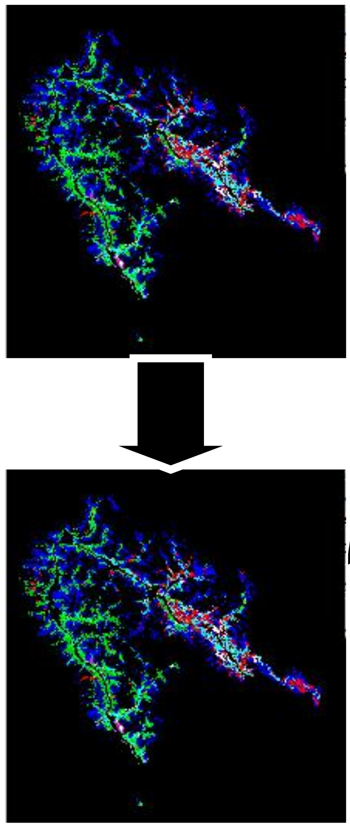
Burnt[MaxBound] <- 0

Species[0,MaxSpp] <- initialSpp

Layer Variables: Initial and Dynamic States

initalSpp

Input View (initial state)



Species

Output View (dynamic state)

Raster Views

LAYER: Species

Landscape Events

Importing Script Variables

Example:

External variables:

$HarvestFile$ = Harvest.txt

- specifies a script variable that can be used in .sel file (e.g. in place of a fixed file name or value)
- can provide a default value (right hand side)

Display Refresh Frequency

– Currently: can only specify for all raster views
– Can change on user interface

Example:

Output Frequency: 365.25

Saving Dynamic Time Series

RasterName Frequency BaseFilename Type

- Actual filename: *BaseFileName_#Run_#Seq* where *Run* is the simulation run and *Seq* is the sequence number
- if frequency is negative, no output will occur

Example:

Output Model Frequency:

StandAge Freq: 10 Filename: Age.tif Type: GEOTIFF

StandAge Freq: SpReportingFreq Directory:grids Type: GEOTIFF

Legends

Define vectors of labels

Examples:

Legends:

SppLegend = cats\spp

LULegend = grids\lu.tif

SoilTypes = soilType.txt

MgmtType = {1:ClearCut, 2:VR}

Macros

Define vectors (1-dimensional arrays) of expressions

- run a function when indexed instead of just looking up a value

Example:

```
Macros:
  testMacro = macro1.ce
```

Hands-on

state space of the simple top-down fire model

- Start LSEdior and open the FireTopDown.sel state-space configuration a file
- Keywords are shown in blue
- Comments are shown in green

- The order of sections doesn't matter, except if a variable depends on other state-space elements, those must be declared first
- There can be more than one of each section (additive)

- The main state-space is global or spatial constants and variables
- The process models (landscape event files) are usually listed near the top

Exercises
state-space

- Read through .sel examples
  a) Game of life: GameOfLife.sel (in models\GameOfLife\models)
  b) Simple bottom-up fire model: FireModelBottomUp.sel (in models\SimpleFireModel\models)
  c) Case study version 7 model: Model.sel (in models\CaseStudy\v7_roads)

Landscape Event Language
(.lse files)

General

- Declarative:
  - state behaviour without step-by-step details
- Non-linear:
  - behaviour specified by properties
  - expressions associated with properties
- Properties:
  - can appear in any order
  - can only appear once (or not at all)
- Case sensitive

Landscape Event (.lse) Files

guidelines for reading

- Focus first on the main expression of each *property* to understand the behaviour before focusing on state changes
  - ➢ Review the properties in the next two slides (and Module 2)

- The order of properties is irrelevant, but the assignments within each property are evaluated sequentially

- Focus on the current context (time and location)
  - ➢ Keep in mind that the main expression and preceding (preliminary) expressions are evaluated in different contexts from the consequent expressions (which is non-linear in space and time)

➢ **Use LSEditor**

Landscape Event Properties Review

startup and initiation of active cells

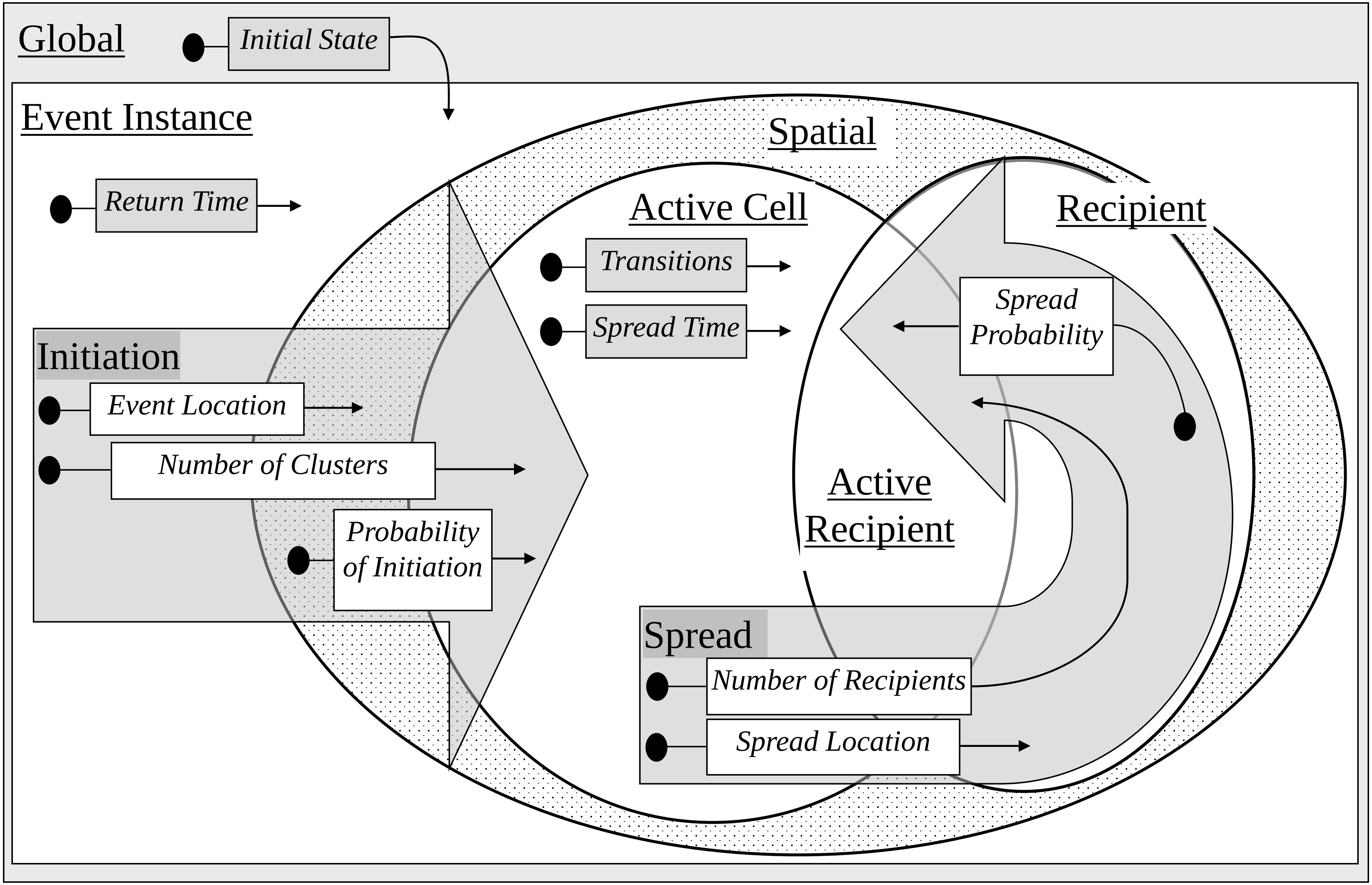
Initial State	Number of initial instances of the event
Return Time	Interval of time between successive instances of the event
Event Location	The set of cells in which the event can potentially initiate
Number of Clusters	The number of cells in which the event will initiate
Probability of Initiation	The relative or absolute probability that the event will initiate in a particular cell
Transitions	Whether the event occurs in a cell or not

Landscape Event Properties Review

spread from an active cell

Spread Time	Interval of time required for an event to spread from the current cell to its neighbours.
Spread Location	The set of cells to which an event can potentially spread from a cell
Number of Spread Recipients	The number of cells to which the event will spread from an affected cell
Probability of Spread	The absolute or relative probability that the event will spread to a particular cell

Spatio-Temporal Contexts Review



Landscape Event (.lse) Files

syntax

Three general sections:

- Event name
- Declarations
- Properties
  - a .lse file only includes the properties it requires (and relies on default behavours for properties not included)

Example Event Name:

```
LSEVENT: Fire
```

Landscape Event Declarations

variable types

variable type: variable name

<u>External State</u>

- Spatial Layers
- Global Variables

<u>Internal State</u>

- Local Variables
- Event Variables
- Cluster Variables
- Cell Variables
- Output Variables

Landscape Event Declarations
full format

Examples:

```
DEFINITIONS
  GLOBAL CONSTANT: HaPerCell, MHA[]
  GLOBAL VARIABLE: meanFireSize, SppProfile[]

  LAYER: StandAge, Elevation

  LOCAL: MeanFiresPerYr
  EVENT VARIABLE: NumFiresToIgnite
  CLUSTER VARIABLE: AreaToBurn
  CELL VARIABLE: Intensity

  OUTPUT VARIABLE: fireStats = fireStats.txt
ENDDEF
```

Landscape Event Properties

template

Preliminary assignments
Main Expression
Consequent assignments

- Main Expression: value drives property behaviour
- Assignments specify state changes:

  variable = expression

- The context defines the spatial domain and available dynamic variables

Landscape Event Properties

property names and end labels

INITIALSTATE	ENDIS
RETURNTIME	ENDRT
EVENTLOCATION	ENDEL
NUMCLUSTERS	ENDNC
PROBINIT	ENDPI
TRANSITIONS	ENDTR
SPREADTIME	ENDST
SPREADLOCATION	ENDSL
NUMSPREADRECIPIENTS	ENDNR
SPREADPROB	ENDSP
ENDCLUSTER	ENDEC
ENDEVENT	ENDEE

Landscape Event Properties

general form of syntax

Two forms:

(i) Standard full form:

<Property Name>
*expression**
<Property Name> = expression
*expression**
<End Label>

Example:
TRANSITIONS
go = Age > 0
TRANSITIONS = go
Age = 0
ENDTR

where *expression** denotes 0 or more expressions

(i) Short-hand form when there is only a main expression:

<Property Name> = expression

Example: TRANSITIONS = Age > 0

Expression Types

Covered in this module:

- Built-in variables
- Continuous
- Bounding
- Classified (Discrete)
- Combinatorial
- Control (loops and iteration)
- Probability Distributions and Density Functions
- Region and Spatial
- Output

Covered in module 3:

- Arithmetic
- Relation and Boolean
- Basic Control (“if”)
- Display

Covered in module 7:

- **Bit-Vector**
- **Matrix**
- **Set, list, tree, graph**

➢ see the User Documentation Appendix 1 for a full list of expressions and options

Built-in Constants and Variables

NUMCOLS

NUMROWS

NUMCELLS

Location	- current cell location
Index	- used in OVER INDEX SEQUENCE
Time	- current time
EndTime	- simulation duration
Run	- replicate number

Continuous Functions

LOG(Expression) Natural logarithm

MAX(Expression, Expression)

ROUND(Expression)

FLOOR(Expression)

CEIL(Expression)

| *Expression* |

Examples:

StandAge = MIN(StandAge+1, MaxStandAge)

Diff = | OldValue - NewValue |

Bounding Functions

MIN(Expression, Expression)
MAX(Expression, Expression)
CLAMP(Expression, MinExpression, MaxExpression)

ROUND(Expression)
FLOOR(Expression)
CEIL(Expression)
| *Expression* |

Examples:

StandAge = MIN(StandAge+1, MaxStandAge)
Diff = | OldValue - NewValue |

Classify Functions

CLASSIFY(Variable)

$value_i$: $Expression_i$

...

END

Example:

```
fireSusc = CLASSIFY(Species)
                Pine: 1
                Spruce: 0.8
                Alder: 0.2
           END
```

Combinational Expressions

KEYWORD
  Expression
  ...
END

AND, OR, SUM, PRODUCT, MIN, MAX, MEAN

Example:

```
oldPine = AND
    StudyArea > 0
    Spp1 EQ Pine
    Age > 100
   END
```

Control Expressions

while loops

WHILE (Expression)
…
END

run sub-expressions as long as the condition is TRUE

OVER INDEX(Start, End)
  x = Index
  …
END

iterate the build-in *Index* variable from *Start* to *End* in increments of 1

- Over Index is the same as a “for loop” but using the built-in *Index* variable

Probability Distributions

Draw a number from a distribution

NORMAL(Mean, StandardDeviation)

LOG NORMAL (Mean, StandardDeviation)

NEXEXP(Mean)

UNIFORM(Min, Max)

POISSON(Lambda)

Examples:

x = NORMAL(10,5)

makeChoice = UNIFORM(0,1) < p

Probability Distributions

discrete distributions

Draw a number from a distribution

CLASSIFIED_DIST
  value$_i$: *Expression*$_i$
  ...
ENDFN

Example:

```
x = CLASSIFIED_DIST
        1: 0.1
        2: 0.2
        3: 0.2
    END
```

Probability Distributions

discrete distributions

Draw a number from a distribution

CLASSIFIED _DIST[VectorVariable]

- *allows drawing from an arbitrary input distribution*

Example:

x = CLASSIFIED _DIST[inputDist]

Where inputDist is a one-dimensional array with values 0.1, 0.2, 0.2, 0.2, 0.3
Note: the first value in a 1-dimensionaly array (vector) is index 0.

Spatial Expressions

DIRECTION(StartLocation, EndLocation)

DISTANCE(StartLocation, EndLocation)

Note: direction is in degrees and distance is in cell units

Example:

Alpha = DIRECTION(Location, ClusterStartLoc)

X = COS(Alpha)

Region Expressions

what regions represent

Region expressions "*return*" a set of cell locations (0 or more)

Can only be used in these situations

- as the main expression of the EventLocation or SpreadLocaton properties
- in an "Over region" expression" (described subsequently)

Region expressions can include an optional a decision function that filters the general set of cells specified by the type of expression

Region Expressions
all cells that meet the condition

REGION WHOLE MAP

DECISION Expression

- Returns all grid cells for which the decision evaluates to TRUE

Example:

```
EVENTLOCATION
  REGION WHOLE MAP
    DECISION StudyArea > 0
END
```

Region Expressions

neighbouring cells that meet the condition

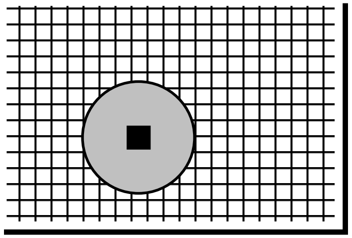
REGION CENTRED(Min, Max, options)

DECISION Expression

- Returns grid cells within the minimum and maximum distance in cell units from the current cell and for which the decision evaluates to TRUE (the distance to the current cell itself is 1)

Example:

```
SPREADLOCATION
  REGION CENTRED(1,1.5)
END
```

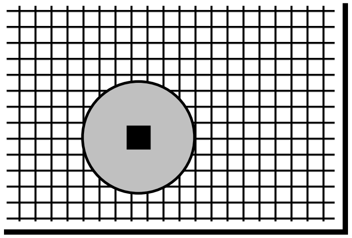


Region Expressions

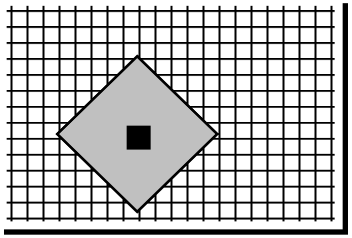
neighbourhood distance options

REGION CENTRED geometric options

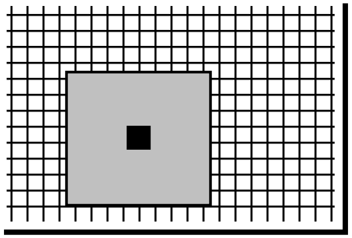
EUCLIDEAN default



CARDINAL



RECT



Region Expressions

cells in a rectangle that meet the condition

REGION RECT(Bottom, Left, Top, Right)

DECISION Expression

- Returns all grid cells in the specific bounding box for which the decision evaluates to TRUE

Example:

```
EVENTLOCATION
  REGION RECT(boxBottom, boxLeft, boxTop, boxRight)
    DECISION StudyArea > 0
END
```

Region Expressions

cells along a straight line that meet the condition

REGION VECTOR(StartLocation, EndLocation)

DECISION Expression

- follows cells along closest straight line
- a cell with be included in the region if the line is within ½ cell distance from the cell centre

Region Expressions

cells in a list of location that meet the condition

REGION LOCATION LIST[X, n]

DECISION Expression

- assesses a set of pre-computed locations
- X is a vector variable (1-dimensional array), and n is the number of items (which must be at most the size of X)

Region Expressions

over region expressions

OVER REGION …

  Expression

  …

END

Visit (go to the spatial context of) each cell in a region

- If assessed in a spatial context, can use the SOURCE keyword to refer to context variables of the originating cell

Example:

```
OVER REGION WHOLE MAP
  StandAge = 0
  Spp1 = DouglasFir
END
```

Output Expressions

output records to a file

OUTPUT(OutputVariable)

label: Expression

varName

…

END

Output a record (single row) to a file (referenced by the output variable

- Output variables are declared in the definitions section of landscape events which specifies the name of the file
- By default, values are numbers (the result of the expression or value of a named variable)

Output Expressions

output legend labels

OUTPUT(OutputVariable)
label: $Variable
label: $Variable {LAYER}
label: $Variable {LegendVector}
END

Output a legend label instead of the associated value

- The legend is either from a layer variable or a legend global constant

Example:

```
OUTPUT(f)
  Soil: $Soil
  newSoil: $x {Soil}
END
```

Exercises
reading models

1) Review the .sel and .lse files to understand the behaviour of the following models
   a) Game of Life
   b) Simple fire model: top down and bottom up versions (FireModelTopDown.lse and FireModelBottomUp.lse)

Exercises

basic model mechanics

Some options to try your skills as making simple changes to an existing model:

- Add outputs (to improve understanding, indicators)
- Generalize (to improve adaptability)
- Simple behaviour changes