Module 6

SELES – the Meta-model

Explore Landscape Event Properties

Andrew Fall

Landscape Systems Ecologist

Gowlland Technologies Ltd.

April 2024

Module 6 Objectives

What you can expect to learn from this module:

- Improve insights in the landscape event meta-model and spatio-temporal contexts:
  - When scheduling event returns
  - In the event initiation process
  - During event spread
  - Context sequencing in time

➢ See SELES User Documentation: Part 1 - section 4

Module Overview

This module will use simple models to explore landscape event properties and related aspects of the landscape event meta-model

- These models are included in the tutorial model files download in the ExploreLSEventProperties folder

1) ReturnTime model: Explore scheduling of two event over time
2) Initiation model: Explore the event initiation process
3) Spreading model: Explore event spread over time
4) Contexts model: Explore sequencing of property contexts over space and time

Return Time Exploration Model

(ReturnTime)

Hands-on

description of the "return time exploration model"

The model state-space includes:

- One dynamic layer named MainLayer, with range [0,1] and initialized to 0
- Three global constants created as a legend:
  - Two legend entries: Preliminary (= 1) and Conseqent (= 2)
  - One legend array: ContextLegend (consists of the 2 preceding constants)

➢ Note: legend arrays are 1-dimensional arrays whose values are the identity function (i.e. index *i* has value *i*), but linked with legend labels (used for i/o)

There is one modelled process:

(ii) ReturnTime: create two event instances (copies of the event). For each:

- o Schedule each return of the event instance randomly between 1 and 10 time units
- o Initiation: start 0 clusters (no initiation – i.e. do nothing)
- o Display the event id and simulation time when each event instance is scheduled, processed (pulled from the event queue) and terminates (after initiation)

➢ That is, return to the landscape at random return steps and do nothing (other than display)

Hands-on

read the "return time exploration model"

Use LSEditor to read the model files, starting with the scenario script

- Open ReturnTime.scn - the script commands are:
  a) Set dimensions to 1 row and 1 column (i.e. a 1 cell landscape)
  b) Load the model configuration ReturnTime.sel file

  ➢ This is about as basic as a scenario can be

- Open ReturnTime.sel - the state-space is configured as:
  a) Time Units: Step, with DecaStep defined as 10 Steps, and a default duration of 100 Steps
  b) Load one landscape event ReturnTime.lse
  c) Create one legend (ContextLegend) with two entries (Preliminary and Consequent)
  d) Create one spatial variable (MainLayer) with range 0 to 1 and initial value 0

  ➢ The state space is about as basic as possible for a spatial model (one spatial variable), and no parameters or other global variables

Hands-on

read the "return time exploration model"

- Open ReturnTime.lse - the event is declared as:
  a) Definitions
     - load the relevant portion of the state space
     - declare one *event variable* (currEvent) – will associate a unique value with each event created by the *InitialState* property

  b) Focus on the main expressions of properties first
     a) INITIALSTATE = 2: create 2 instances of the event on simulation startup
     b) RETURNTIME = timeInc: schedule return of event using the timeInc variable, where timeInc = ROUND(UNIFORM(1,10))
     c) NUMCLUSTERS = 0: start 0 clusters (no initiation)
     d) ENDEVENT = TRUE: allow the event to end (the only current option)

  c) The other expressions (in the preliminary and consequent contexts of the *InitialState*, *ReturnTime* and *EndEvent* properties) are to set up and display the context, time, the *EventId* built-in variable (incremented each return) and the currEvent event variable

➢ That is: schedule each event instance to return between 1 and 10 time units in the future, selected from a uniform distribution (and rounded to the nearest step), do nothing then end

Hands-on

run the "return time exploration model"

- Start SELES and open ReturnTime.scn
- Open the Simulation control (DynamicModels menu: Simulate or down blue arrow) and start the model (press Simulate)

- For each display box that pops up (from a DISPLAY expression), look in the landscape event file to find where the model is at. Press OK to close it (pressing Cancel will stop that DISPLAY expression until the landscape event is reloaded – press when you get bored and want to end the simulation)

- The first display is time 0 in the *InitialState* property, *Preliminary* expressions
- The next is also at time 0 in the *InitialState* property, but now in the *Consequent* expressions – this is for the first of the two instances created by the *InitialState* property
  - the consequence of setting *InitialState* to 2 is to create two instances, each with their own currEvent event variable
  - at this point the currEvent event variable has the same value as the EventId built-in variable

Hands-on

run the "return time exploration model"

- The next is also at time 0 in the *ReturnTime* property, *Preliminary* expressions – this is to schedule the first instance (currEvent =1)
  - The timeInc field shows the time at which this instance will return for initiation (note this – it helps to track the time sequencing)
  - the consequence of setting *ReturnTime* to an integer between 1 and 10 is to schedule the event instance to be processed (i.e. initiation) at that time increment
- The next is still at time 0 in the *InitialState* property, *Consequent* expressions – this is for the second of the two instances created by the *InitialState* property
  - Note the currEvent event variable has value 2 instead of 1
- This is followed by a similar display from the *ReturnTime* property, *Preliminary* expressions
  - Note the time increment – the events will be pulled off the queue in order of time

Hands-on

run the "return time exploration model"

- The next display is now at some time *t* > 0 (the minimum increment selected for the two instances (note the value of `currEvent`) in the *ReturnTime* property, but now in the *Conseqent* expressions
    - ➢ The consequent expressions of *ReturnTime* are non-spatial (but for an event instance) and often used to set up a an event prior to initiation

- There is no initiation (since *NumClusters* is 0), so no clusters are initiated

- The next display is also at time *t* in the *EndCluster* property, *Consequent* expressions – the event instance is terminating
    - ➢ Often used for reporting or cleanup when an instance is finished
    - ➢ In spreading events *EndCluster* may be happen at a later time period than initiation, based on the cumulative time spent spreading

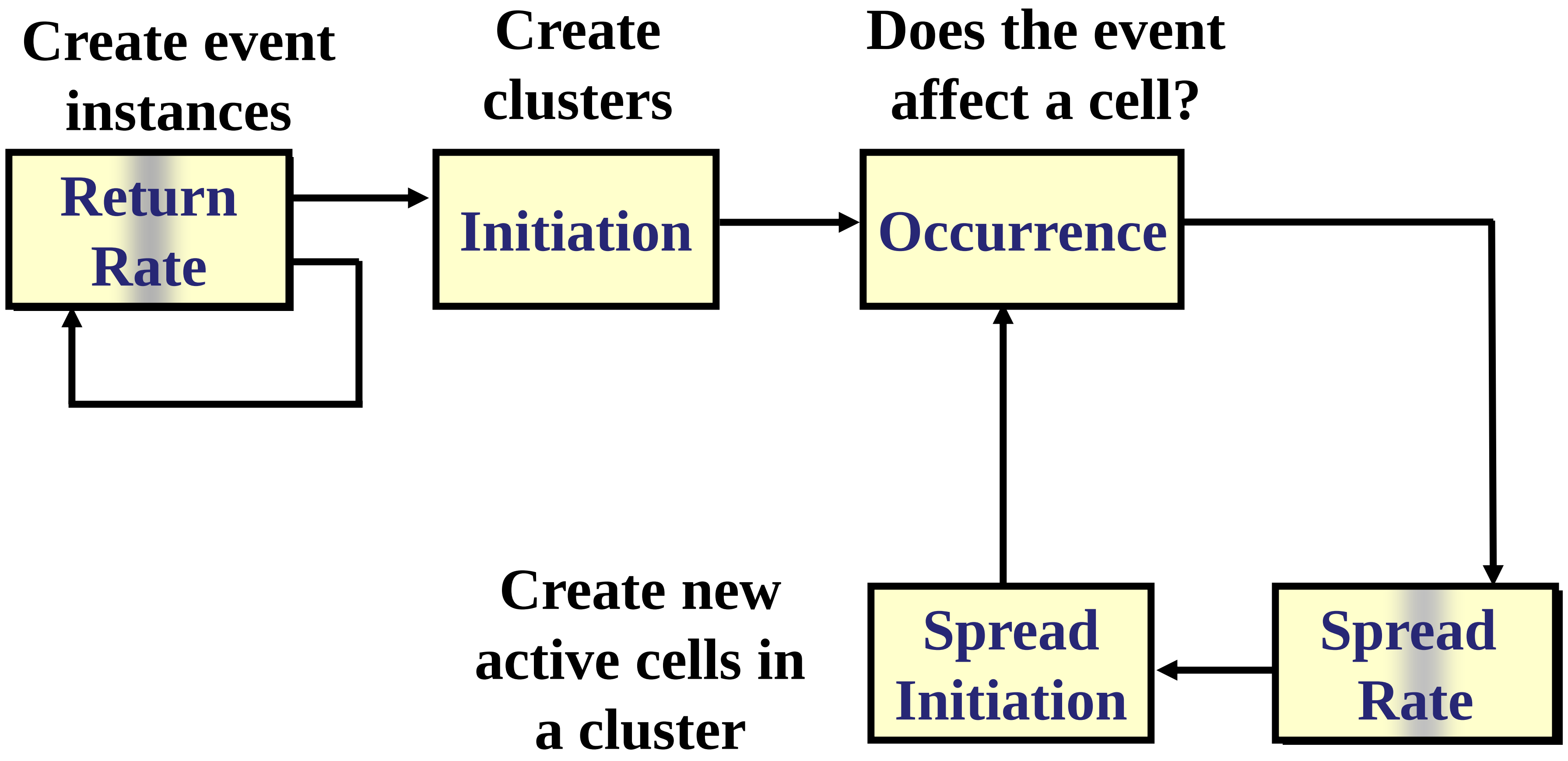
Hands-on

run the "return time exploration model"

- This next display, still at time *t*, is back to the *ReturnTime* property, *Preliminary* expressions – this is to schedule the next round of the event instance first pulled off the queue
  - ➢ Review the general meta-model diagram on the next slide – essentially, the very last thing an event instance does is to create a new instance to be scheduled by *ReturnTime*
  - ➢ Even though this is a new event instance (EventId is 3), SELES transfers the values of any event variables (so you can alternatively think of it as a continuation of the same event instance – with the same value for currEvent)
  - ➢ The timeInc field shows the time *increment* at which this instance will return for initiation (the simulation time will be *t* + timeInc as shown in the *TimeToProcess* field)

- As this sequence continues, at each (randomized) time step the next event instance will be pulled off the event queue then processed (which may be one or the other currEvent)
  - ➢ Sometimes one event instance may be processed multiple times before the other one

Navigating Contexts via Landscape Events



Hands-on

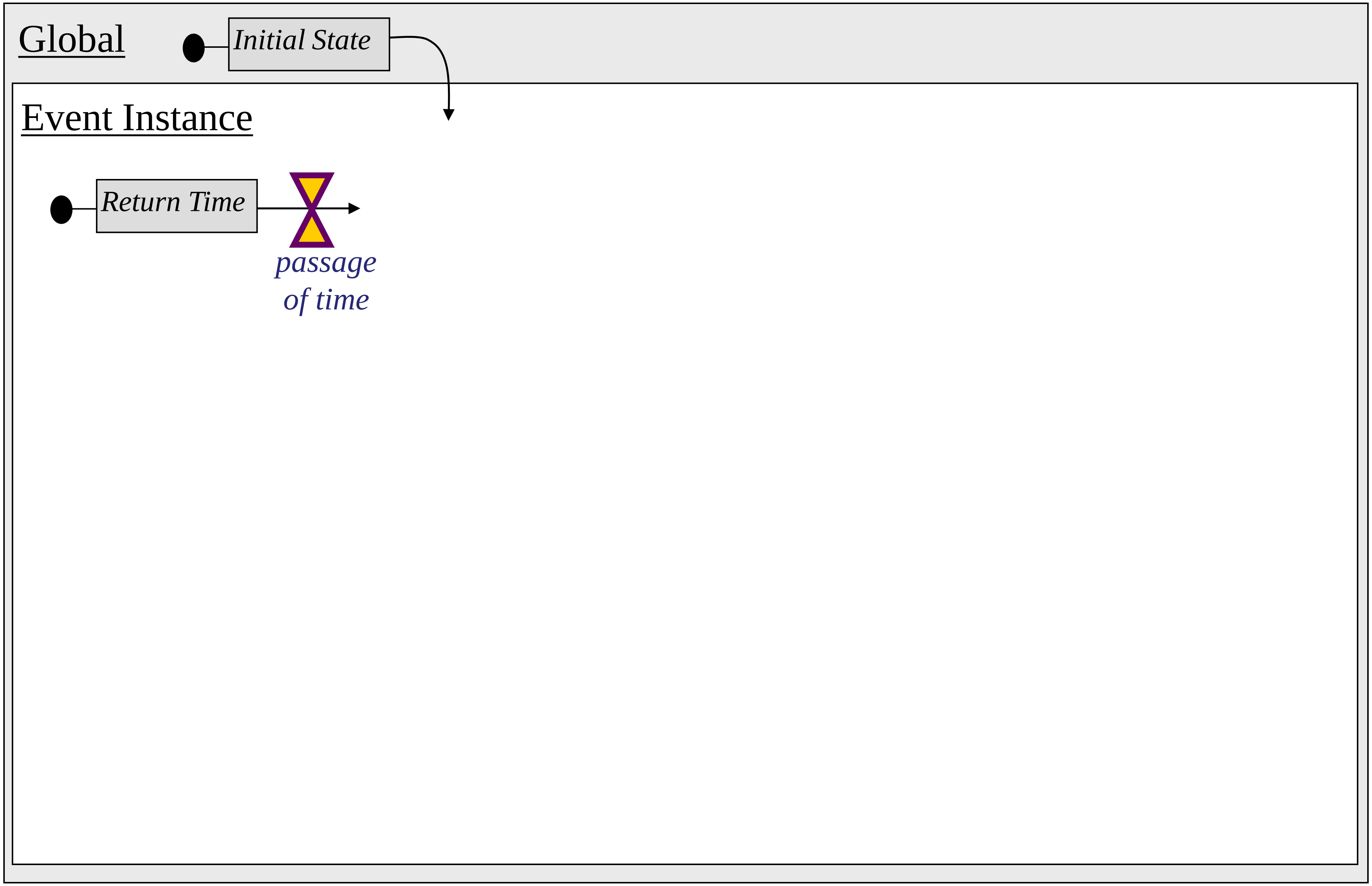
lessons from the "return time exploration model"

- The *InitialState* property can be used to create multiple copies of an event
  - ➢ This can be useful to create a separate instance for sub-regions (e.g. different wildfire regime types or different management units)

- The *ReturnTime* property is one key way that models navigate through time
  - ➢ Event instances will be placed on the queue for processing at a future time *t*
  - ➢ Many other scheduled processes may occur in the intervening time, but eventually time advances to *t* and the event instance is pulled off the event queue and processed
  - ➢ Sequences of event instances may be simple (e.g. annual increments) or complex

- The roles of these two properties in the overall meta-model of a landscape event are illustrated on the following slide
  - *InitialState* takes a model from a global context (no event instances) to one or more event instance contexts (non-spatial but with an active event, including event variables)
  - *ReturnTime* takes a model from an event instance context at time *t* to an event instance context for the same event instance but at time *t* + *ReturnTime*

Spatio-Temporal Contexts



Event Initiation Exploration Model (Initiation)

Hands-on

description of the "initiation exploration model"

The model state-space includes:

- Three spatial constants: StudyArea, Elevation and Spp1
- Two global constants: MaxInstances and MaxProbInit
- Three dynamic layers: ProbInitSurface with range from 0 to MaxProbInit, and EventLoc and Initiations with ranges from 0 to MaxInstances, all initialized to 0
- One global variable: currClusterId, initialized to 0

There is one modelled process:

(i) Create one event instance

(ii) On each time step:

- Initiate in 1000 cells in the study area with a relative probability that increases with elevation based on the function: $((Elevation - 600)/1000)^2$

Hands-on

read the "initiation exploration model"

Use LSEditor to read the model files, starting with the scenario script

- Open Initiation.scn - the script commands are:
    a) Define a script variable $gisData$ with a path to the case study grids
    b) Load three input layers from GeoTiff files: StudyArea, Elevation and Spp1
    c) Set dimensions based on the StudyArea layer
    d) Load the model configuration Initiation.sel file

- Open Initiation.sel - the state-space is configured as:
    a) Time Units: Step, with DecaStep defined as 10 Steps, and a default duration of 10 Steps
    b) Load one landscape event Initiation.lse
    c) Create spatial constants, global constants, spatial variables and global variables as in the description on the preceding slide

Hands-on

read the "initiation exploration model"

- Open Initiation.lse - the event is declared as:
  - a) Definitions
    - o load the relevant portion of the state space
  - b) Focus on the main expressions of properties first
    - *a) InitialState:* not specified – default is to create 1 instance on simulation startup
    - b) RETURNTIME = 1: schedule return of the event each time unit
    - c) EVENTLOCATION: all cells (REGION WHOLE MAP) in the study area (StudyArea > 0)
    - d) NUMCLUSTERS = 1000: start 1000 clusters
    - e) PROBINIT = p: select cells stochastically with relative probability *p* where
      p = 100*((Elevation-600)/1000)^2 (increasing function of elevation)
    - f) No *Transitions* (so nothing occurs) and no spread properties (so no spreading)
  - c) The other expressions (in the consequent contexts of the *EventLocation* and *NumClusters* properties and both contexts of the *ProbInit* property) are to assign spatial and global variables to illustrate when and what happens in these contexts

➢ That is: on each time step, initiate in 1000 cells selected with likelihood increasing with elevation

Hands-on

run the "initiation exploration model"

- Start SELES and open Initiation.scn
- Open the Simulation control and start the model

- The default duration is 10 steps (so 10,000 clusters initiated)

- Open a legend (View menu: Show Legend) and set the Slowdown on the Simulation control to 100. Run again to observe changes.

- The EventLoc layer increments by 1 each step, from 1 to 10, in the study area
  - ➢ The consequence of declaring the *EventLocation* to be all cells with StudyArea > 0 is to create a spatial context for each of these cells *in which the event may potentially initiate*
  - ➢ The *EventLocation Consequent* expressions are evaluated in each of these spatial contexts – the only expression is to set the value in the current cell of the EventLoc layer to EventId (but not more than MaxInstances which is 10)

Hands-on

run the "initiation exploration model"

- The *NumClusters* property is evaluated once for each event instance, when it is pulled off the event queue
  - ➢ The consequence of declaring *NumClusters* to be 1000 is to pick 1000 cells and to initiate a cluster in each (with the selected cells *activated*)
    - ➢ An *active cell* is a cell location that has been dynamically selected via initiation or spread
    - ➢ Note: there may be more than one active cells in given grid cell at any given time
  - ➢ The *NumClusters Consequent* expressions are evaluated in each of the resulting spatial active cell contexts – the only expression is to increment the currClusterId global variable – this can be seen on the Simulation control to increase to 10,000 (the expected number of clusters over 10 steps)

Hands-on

run the "initiation exploration model"

- The *ProbInit* property is evaluated once per cell in the *EventLocation*
  - The *ProbInit Preliminary* expressions set p= 100*((Elevation-600)/1000)^2 and ProbInitSurface to p (p is not part of the state space so is temporary)
  - Since Elevation doesn't change over time, the value of p and ProbInitSurface remain the same in each cell (even though they are recalculated each step)

  - The main expression of *ProbInit* is set to p (so the internal real-valued probability surface is the same as the ProbInitSurface layer)

  - The consequence of specifying *ProbInit* to increase with Elevation is to pick cells in which to initiate clusters with higher likelihood at higher elevations
  - The *ProbInit Consequent* expressions are evaluated in each of the resulting spatial active cell contexts – the only expression is to set the value in the current cell of the Initiations layer to EventId (but not more than MaxInstances which is 10)
  - Note how the Initiations layer only has relatively few cells set (compared to the EventLoc and ProbInitSurface layers)

Excercises

check outputs

1) What is the average elevation of initiations compared to overall (in the study area)?
   - Hint: use a Value Model

2) How many unique cells are selected out of the 10,000 initiations?

3) Compare the frequency distributions of elevation and initiations (in the study area)

Review Module 3 (user interface) if needed

Hands-on

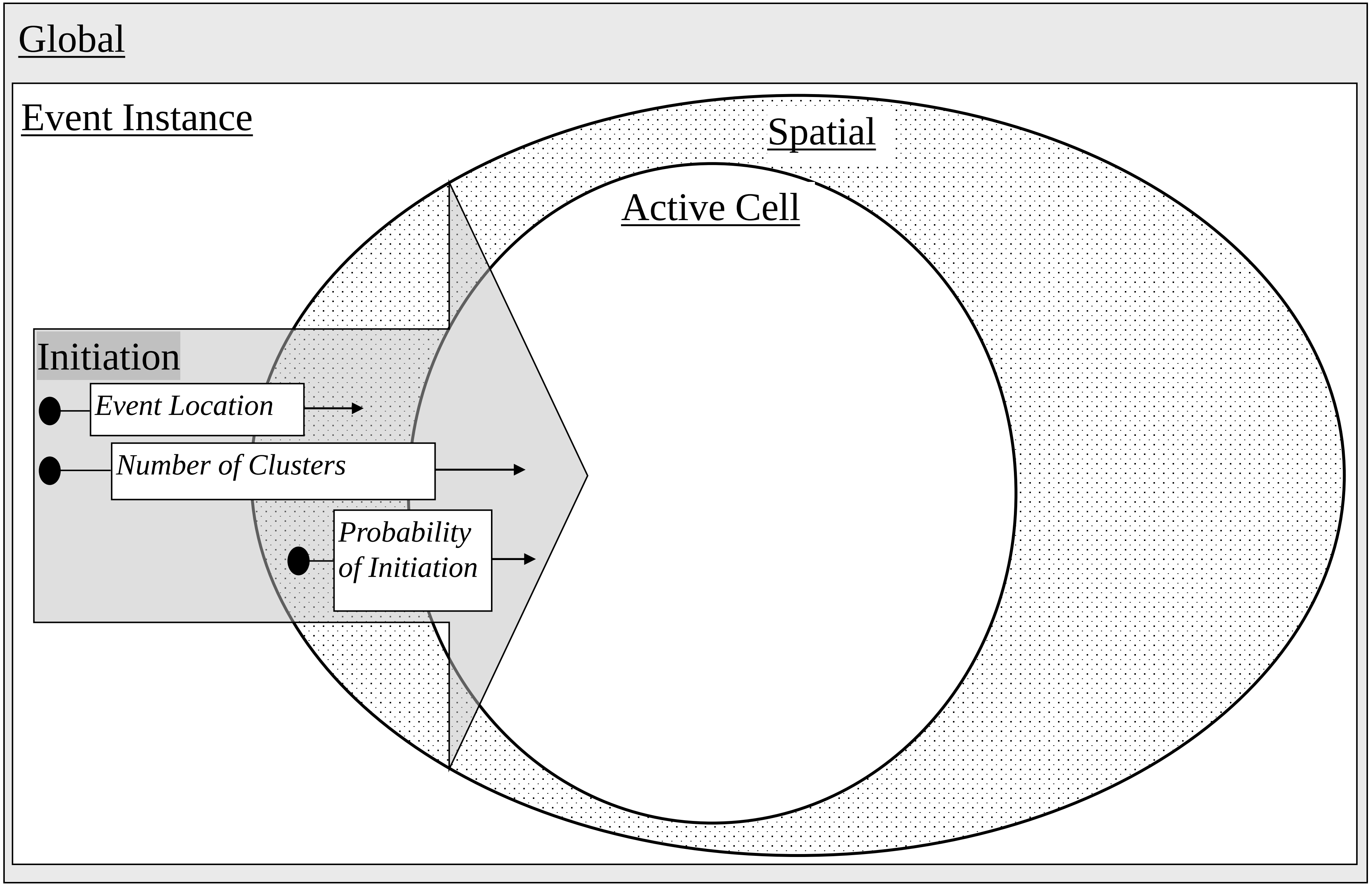
lessons from the "initiation exploration model"

- The *EventLocation*, *NumClusters* and *ProbInit* properties operate together to take an event from a non-spatial context (after being pulled off the queue) to zero or more spatial active cells, each in their own clusters

- The roles of these two properties in the overall meta-model of a landscape event are illustrated on the following slide
  - *EventLocation* takes a model from a non-spatial event instance context to a set of spatial contexts (that aren't active, just *potential activation locations*)
  - *NumClusters* takes a model from a non-spatial event instance context to a set of active cell contexts (spatial activated cells, one for each cluster)
  - *ProbInit* takes a model from a set of spatial (but not active) contexts defined by the *EventLocation* to a set of active cell contexts (spatial activated cells, one for each cluster)

➢ Different combinations and options for these three properties can be used for a wide range of initiation behaviours (including emergent number of initiations and initiating cells in a certain order)

Spatio-Temporal Contexts



Event Spreading Exploration Model (Spreading)

Hands-on

description of the "spreading exploration model"

The model state-space includes:

- Two dynamic layers: DistanceFromSrc and TimeFromSrc with ranges from 0 to at least the max distance/time steps from the center
- One global variable: SpreadType, initialized to 1

There is one modelled process:

(i) Create one event instance, run once at time 0 and initiate one cluster in the centre

(ii) Spread to adjacent neighbours (not yet visited) using one of 5 spreading rules:

1) To the *4* cardinal neighbours (up, down, left, right) at an *equal rate*
2) To the *8* neighbours (including diagonals) at an *equal rate*
3) To the *8* neighbours with a *differential rate* base on local distance
4) On even time steps: to the *4* cardinal neighbours; on odd time steps to the *8* neighbours, all at an *equal rate*
5) To the *8* neighours at a *differential rate* based on the incremental distance from the start location (centre cell)

Hands-on

read the "spreading exploration model"

Use LSEditor to read the model files, starting with the scenario script

- Open Spreading.scn - the script commands are:
  a) Set dimensions to 100 rows x 100 columns (i.e. a 10,000 cell landscape)
  b) Load the model configuration Spreading.sel file

- Open Spreading.sel - the state-space is configured as:
  a) Time Units: Step, with KiloStep defined as 1000 Steps, and a default duration of 100 Steps
  b) Load one landscape event Spreading.lse
  c) Create spatial constants, global constants, spatial variables and global variables as in the description on the preceding slide
    - ➢ The global constant MaxDist is used to set the upper bound of the TimeFromSrc and DistanceFromSrc layers (MaxDist is set to the number of rows + number of columns divided by 2 - the maximum distance/time for any of the spread rules)
    - ➢ Both layers are initialized to MaxDist
    - ➢ Include a second global variable nActiveCells to track the number of active cells

Hands-on

read the "spreading exploration model"

- Open Spreading.lse - the event is declared as:
  a) Definitions
     - load the relevant portion of the state space
     - Declare a cluster variable StartLocation and two active cells variables distInc and d
  b) Focus on the main expressions of properties first
     a) RETURNTIME = 0: schedule the first instance at time 0 and no scheduling thereafter (*ReturnTime* of 0 only schedules an event instance during simulation startup)
     b) EVENTLOCATION: centre cell (REGION LOCATION using the middle row and column)
     c) PROBINIT = 1: just initiate in that single centre cell
     d) TRANSITIONS = TRUE: always occur (continue)
     e) SPREADTIME: spread at a rate set by cell variable distInc (depends on the spread rule and is either fixed at 1 step or based on a distance function)
     f) SPREADLOCATION: adjacent neighbours, including diagonals for some spread rules
     g) SPREADPROB = 1: spread to all neighbours in the *SpreadLocation*

➢ That is: initiate 1 cluster in the centre and spread to adjacent neighbours until the edge of the grid is reached, with the rate of spread and the neighbourhood controlled by the spread rule

Hands-on

read the "spreading exploration model"

c) Other expressions are for output

- Track the number of active cells along the spreading front
  - Increment when a new active cell is created after initiation or spread (in the *Consequent* expressions of the *Transitions* property)
  - Decrement after an active cell has finished spreading and is about to terminate (in the *Consequent* expressions of the *SpreadTime* property – often later in time)

- Set the distance and time from the start cell
  - Set the cluster variable StartLocation to the built-in variable Location in the *Consequent* expressions of *ProbInit* (the current location, which is the centre cell)
  - Set the DistanceFromSrc layer to the distance between the centre cell and the current location in the *Consequent* expressions of the *Transitions* property (i.e. in an active cell when it is first reached)
  - Set the TimeFromSrc layer to the current time step in the *Consequent* expressions of the *SpreadTime* property (i.e. in an active cell after it has finished spreading and is about to terminate) – this must be done here to account for the *SpreadTime* increment

Hands-on

run the "spreading exploration model"

- Before diving into the spread rules, start SELES and open Spreading.scn
- Open the Simulation control and start the model
- Change the SpreadType on the Simulate control from 1 to 5 and run each setting (click on the variable, change the value at the top of the list and press Set Initial State)
  - To see in slow motion, set the Slowdown value to 100
- For each run, compare the DistanceFromSrc layer (which records the straight-line distance from the centre cell) with the TimeFromSrc layer (which records the time step at which a cell is reached)
  - Watch the nActiveCells global variable on the Simulate control increase then decrease
  - Use a Value Model to query differences

Hands-on

read and run the "spreading exploration model"

Spread rules: for each, set the SpreadType to the correspond rule number and run the model while also reading at the Spreading.lse file with that rule in mind

1) To the 4 adjacent cardinal neighbours (up, down, left, right) at an equal rate

- Maximum spread distance: 1 cell (just the closest cardinal neighbours)
- Spread rate: 1 cell/step
- ➢ Spread at an equal rate to unvisited neighbours up, down, left and right
- ➢ Effect: bias in diagonal directions - cells along those directions are reached later in time than by a straight line (i.e. *too slow*), with the discrepancy decreasing toward cardinal directions where distance matches time

Exercise: what is the average and maximum bias error?

- ➢ Hint: use a Value Model

Hands-on

read and run the "spreading exploration model"

2) To the 8 adjacent neighbours (including diagonals) at an equal rate

- Maximum spread distance: 1.5 cells (diagonal neighbours are at a distance of the square root of 2 cells (a bit over 1.4) so 1.5 is useful to include diagonals but not further cells)
- Spread rate: 1 cell/step
- ➢ Spread at an equal rate to unvisited adjacent neighbours including diagonals
- ➢ Effect: bias in diagonal directions - cells along diagonal directions are reached sooner in time than by a straight line (i.e. *too fast*), with the discrepancy decreasing toward cardinal directions where distance matches time

Exercise: what is the average and maximum bias error?

Hands-on

read and run the "spreading exploration model"

3) To the 8 adjacent neighbours with rate equal to the distance

- Maximum spread distance: 1.5 cells (include diagonal neighbours)
- Spread rate: distance to neighbour in cell units/step (i.e. 1 for cardinal, 2^0.5 for diagonal)
- ➢ Spread to unvisited adjacent neighbours including diagonals at a rate equal to their local distance from the spreading cell
- ➢ Effect: bias between cardinal and diagonal directions - cells along those directions are reached later in time than by a straight line (i.e. *too slow*), with the discrepancy decreasing toward cardinal and diagonal directions where distance matches time

Exercise: what is the average and maximum bias error?

Hands-on

read and run the "spreading exploration model"

4) On even time steps: to the 4 adjacent cardinal neighbours; on odd time steps to the 8 adjacent neighbours, all at an equal rate

- Maximum spread distance: 1 cell on event time steps (cardinal neighbours) and 1.5 cells on odd time steps (include diagonal neighbours)
- Spread rate: 1 cell/step
- ➢ Spread at an equal rate to unvisited adjacent neighbours, with diagonals included every other step
- ➢ Effect: bias between cardinal directions - cells off cardinal directions are reached later on cardinal directions, with discrepancy highest in the mid-point between cardinal and diagonal directions)

Exercise: what is the average and maximum bias error?

Hands-on

read and run the "spreading exploration model"

5) To the 8 adjacent neighours at a time step equal to the incremental distance from the start location

- Maximum spread distance: 1.5 cells (include diagonal neighbours)
- Spread rate: incremental distance from start location in cell units/step: the difference between the distance (a) from the start point to this cell and (b) from the start point to the spreading cell (may be any real value)
- ➢ Spread to unvisited adjacent neighbours including diagonals at a rate equal to their incremental distance from the spreading cell
- ➢ Effect: no bias (time = distance)

Exercise: verify if there is any bias error?

Hands-on

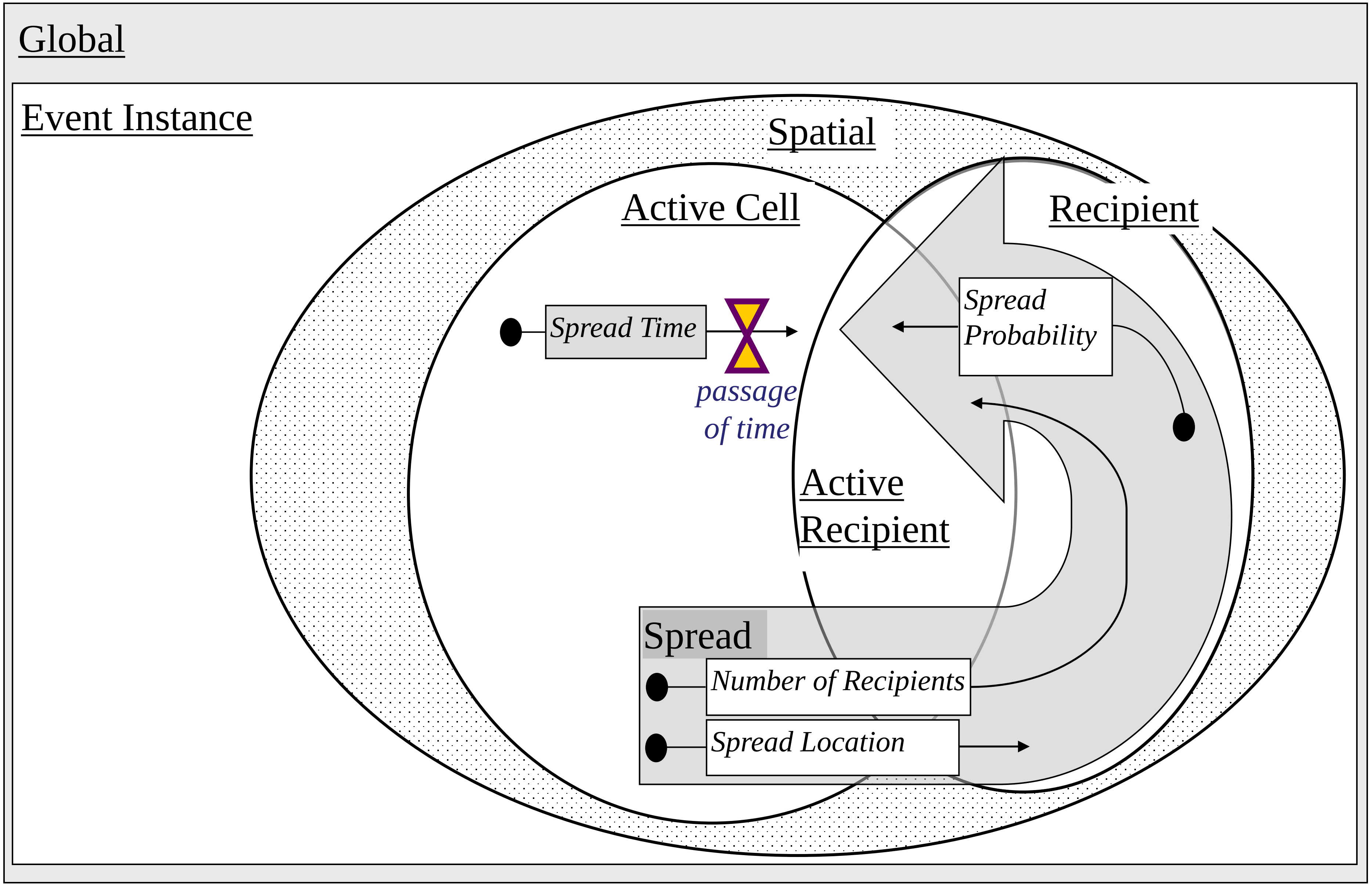
lessons from the "spreading exploration model"

- The *SpreadTime*, *SpreadLocation*, *NumRecipients* and *SpreadProb* properties operate together support modelling of very different spread behaviours in cell units/step to navigate through space and time

- The roles of these two properties in the overall meta-model of a landscape event are illustrated on the following slide
  - *SpreadTime* takes an active cell created at a given time t to its spreading context at time *t* + *SpreadTime* (i.e. schedules it on the event queue)
  - *SpreadLocation* takes an active cell context (after being pulled off the event queue) to a set of zero or more spatial recipient contexts (that aren't active, just *potential cells that may be activated via spread*)
  - *SpreadProb* takes a set of spatial (but not active) recipient contexts defined by the *SpreadLocation* to a set of active cell contexts (new activated cells in the same cluster)
  - Cells activated via spread that pass the *Transitions* property test loop back to *SpreadTime*

➢ Modellers must be aware of potential grid biases (which become less prevalent with increasing stochasticity and variability)

Spatio-Temporal Contexts



Exercise

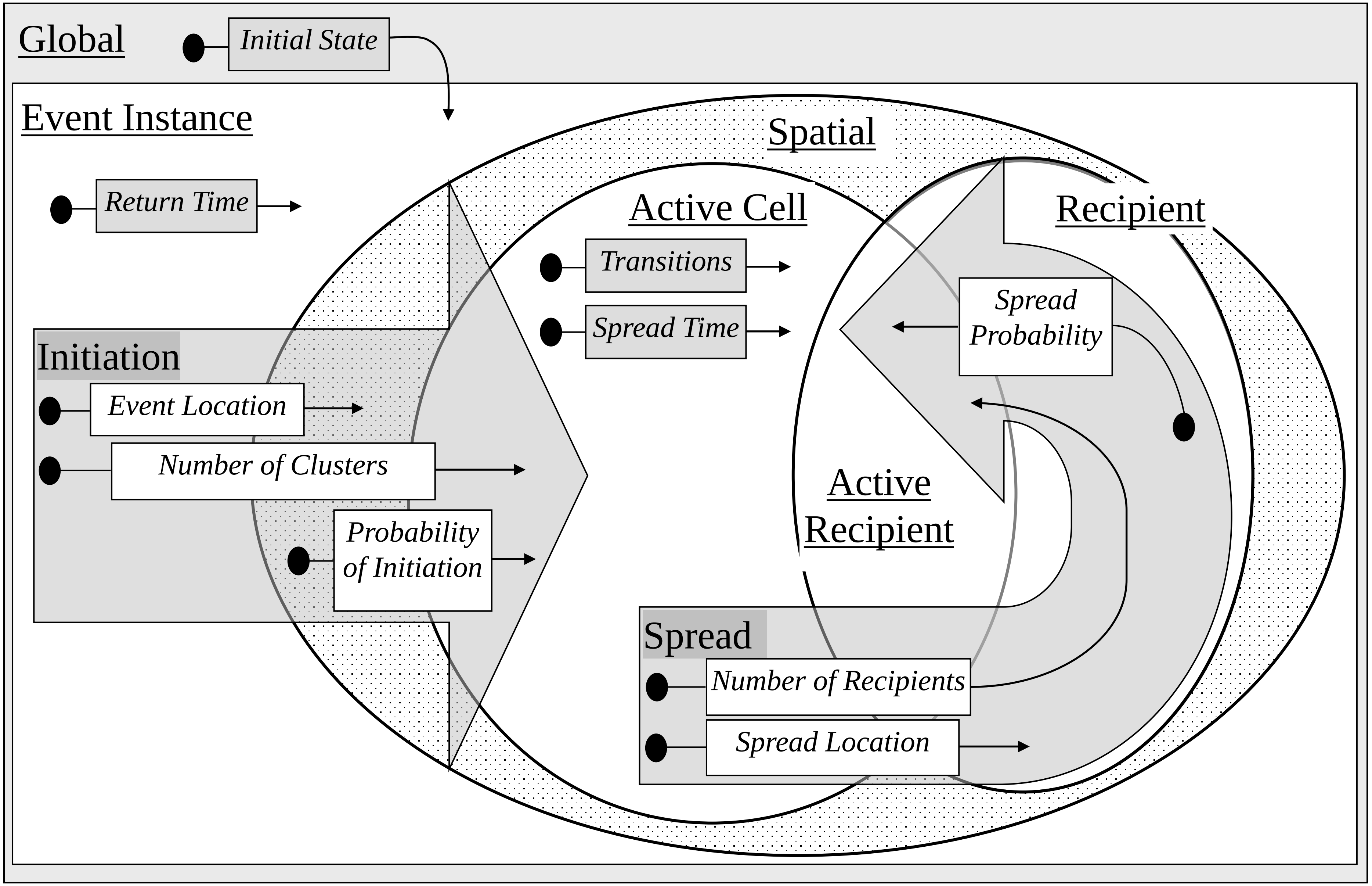
read and run the "context exploration model"

- Read the ContextsModel_readme.txt file for an overview
- The model files are Contexts.scn, Contexts.sel and Contexts.lse

- This model is similar to the "*return time exploration model*", but displays information for a more complete set of properties and contexts
  - Run and modify the model to see how contexts change
  - May be useful to gain more insight into how properties navigate space and time

- This model is designed to help understand the roles of all properties in the overall meta-model of a landscape event as illustrated on the following slide

Spatio-Temporal Contexts


Global
Initial State
Event Instance
Return Time
Spatial
Active Cell
Transitions
Spread Time
Recipient
Spread Probability
Initiation
Event Location
Number of Clusters
Probability of Initiation
Active Recipient
Spread
Number of Recipients
Spread Location