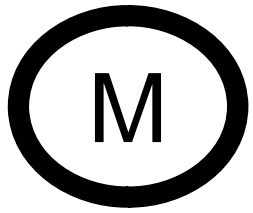


Markov Microsimulation: Model Debugging and Reporting Techniques

TreeAge Software, Inc.

- **Revision 1:**

- * Added text on Slide 15



Simulation Modeling

- Webinar presumes:
 - Experience building Markov models in TreeAge Pro
 - Some exposure to simulation (e.g., microsimulation and tracker variables)
 - Interest in issues related to building complex simulation models
 - Debugging, customized reporting, etc...

Simulation Modeling

- To review:
 - Simulating 1st-order “uncertainty” (variability)
 - Randomly walk individuals through **chance events**, accumulating costs, utilities, et al
 - Run many individual trials and summarize the list of results (mean and statistics)
 - Enables tracker variables for patient history...
 - ... but, simulation includes more options than just trackers, as we’ll see

Simulation Modeling

- **If you *are not* very familiar with simulation and tracker variables, that's okay... still should be useful.**

Agenda

- Simulation-related tools and techniques:
 - Modeling
 - Tracker variables (of course)
 - Not the very basics, but interaction with...
 - Distributions for individual-level variability
 - As opposed to for PSA parameters
 - Reporting/Debugging
 - GlobalN() and global matrices –
 - **More flexible than even trackers**
 - Debugging (pane, preferences, and options)

Cancer Model

- The next few slides describe the model we'll use to illustrate some features
 - Loosely based on:
 - BCMA > BC Medical Journal > Issues > BCMJ July/Aug 2003 Edition
 - Computer simulation of the effect of different colorectal cancer ... A computer simulation model of colorectal cancer in British Columbia is used to compare ...
 - www.bcma.org/public/bc_medical_journal/BCMj/2003/july_august_2003/computer_simulation.asp - 57k -

Cancer Model

- Colorectal cancer model assumptions:
 - Ignoring incidence
 - Modeling prevalence
 - Everyone starts with polyps
 - Polyp type and location are defined for each trial at start
 - Progression to cancer
 - Depends on type and location of polyps
 - Progression of cancer through stages
 - Stages
 - 1 = Local; 2 = Regional; 3 = Distant
 - Transitions to diagnosis, cure and death depend on cancer stage

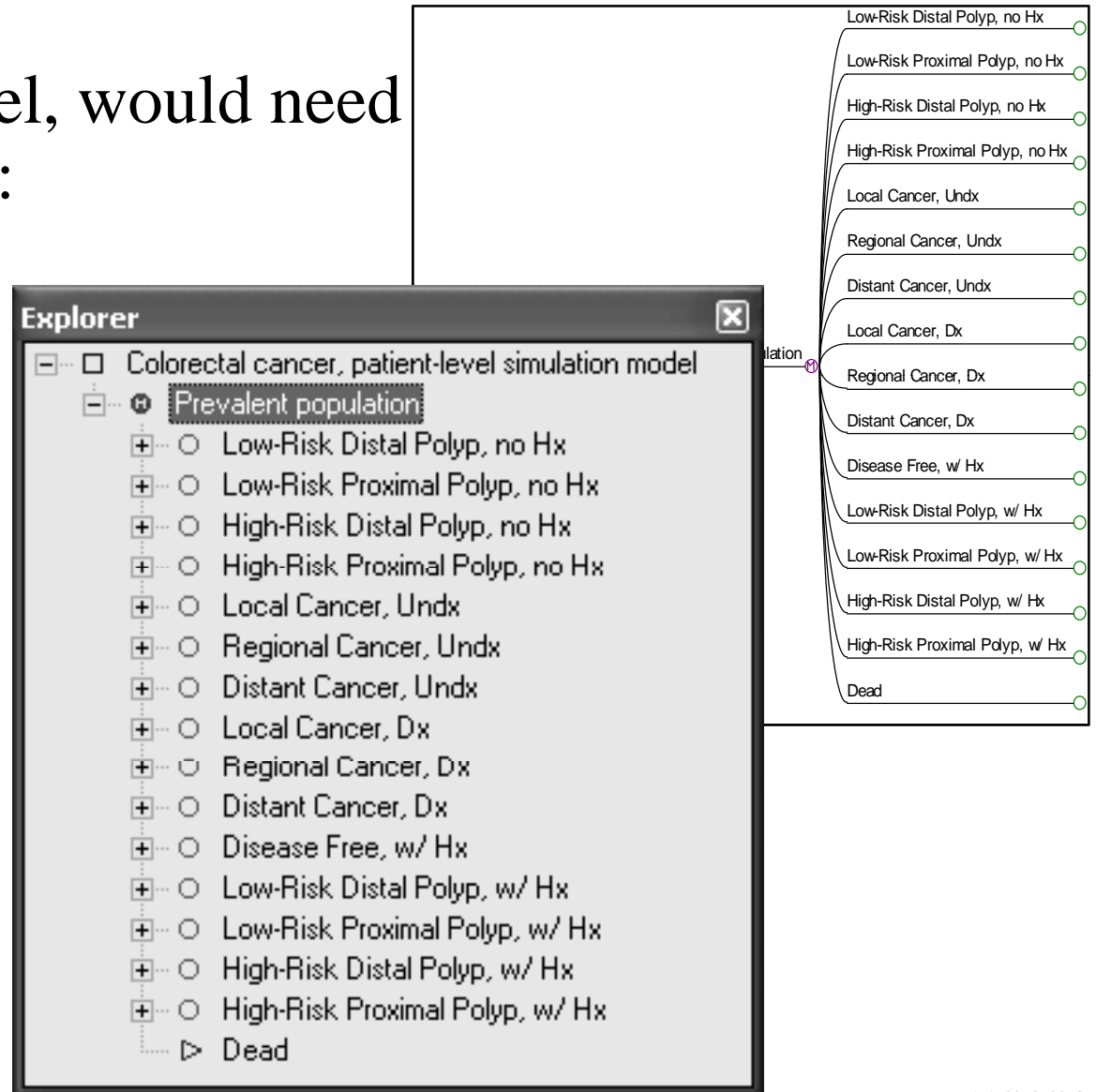
Cancer Model

- Colorectal cancer model:
 - 4 states: Polyps, Cancer, Disease-Free, Dead
 - These states would be insufficient for cohort analysis
 - Would need more states to handle transitions that depend on polyp attributes and cancer stage
 - Our model will use trackers to handle “sub-states”

Cancer Cohort Model

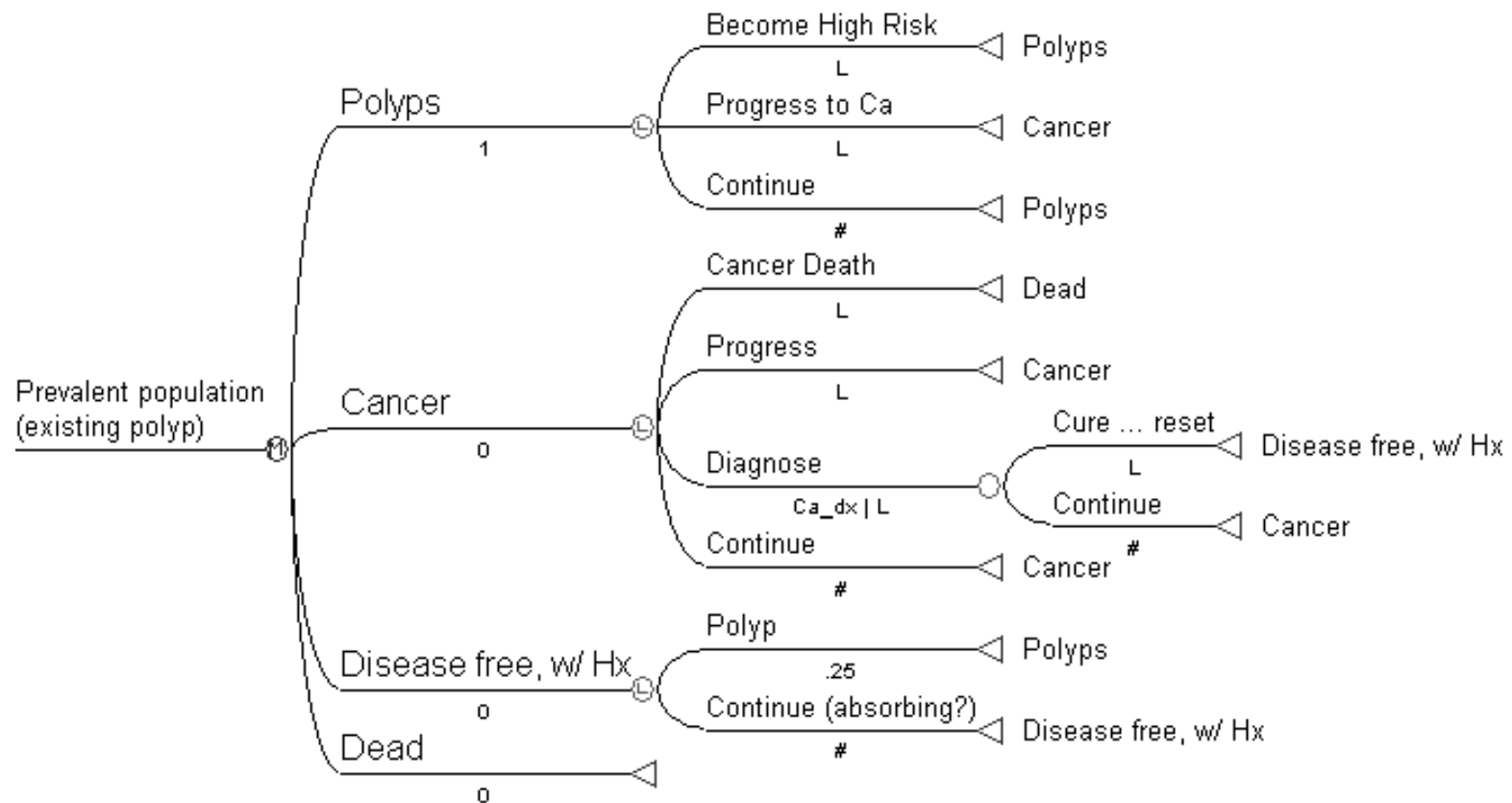
- As a cohort model, would need lots of states, e.g:

- Need explicit branches for all “sub-states” (permutations of factors/traits within a state)



Simulation Model

- Simulation handles complexity in a structurally simpler (fewer states) tree:



Simulation Model

- We will measure:
 - Survival-related measures:
 - Life years, counts (deaths from cancer), etc...
 - Both cohort and simulation models could handle these
 - Cohort model can report 9 attributes/reward sets
 - Simulation model – unlimited report attributes
 - Global matrices take reporting to another level
 - Not measuring costs

Simulation Model

- Parameter definitions:

p_Distal	Proportion of distal polyps	0.6
p_DistantCaCure	Probability of cure of distant cancer	0.04
p_DistantCaDx	Annual probability of diagnosis from s	1
p_DistantCaMort	Annual colorectal mortality rate distar	0.55
p_HiRiskPolyp	Prevalence of high-risk polyps at age	0.03
p_HiRiskPolyp_To_LocalCa	High-risk polyp to asymptomatic loca	0.017
p_LocalCa_to_Regional	Asymptomatic local cancer to asymp	0.22
p_LocalCaCure	Probability of cure of local cancer	0.65
p_LocalCaDx	Asymptomatic local cancer to symptc	0.17
p_LocalCaMort	Annual colorectal mortality rate local	0.23
p_LoRiskPolyp_To_Hi	Low- to high-risk polyps annual trans	0.024
p_RegionalCa_To_Distant	Asymptomatic regional cancer to dist	0.5
p_RegionalCaCure	Probability of cure of regional cancer	0.45
p_RegionalCaDx	Asymptomatic regional cancer to syn	0.45
p_RegionalCaMort	Annual colorectal mortality rate regio	0.28
rand	rand number generator	DistForce(1)

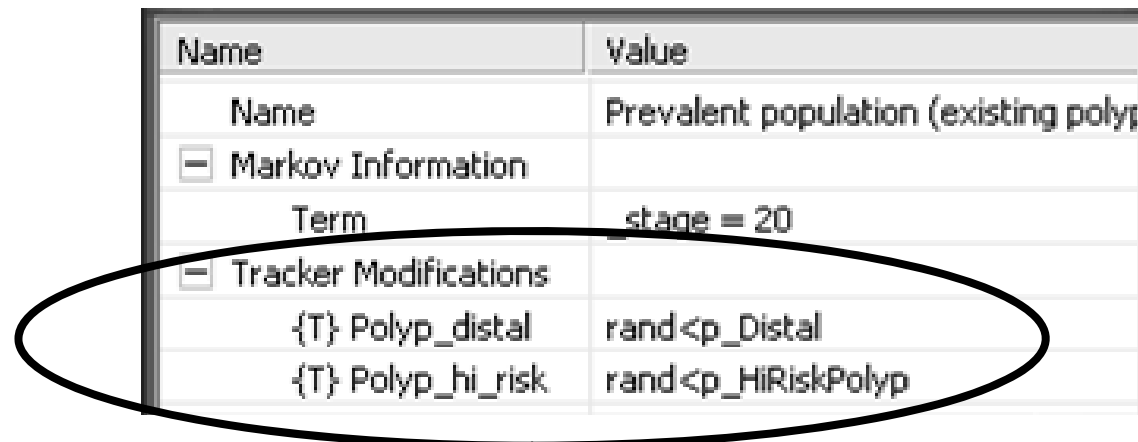
Name	Value
+ Comments	
[-] Variables	
p_Distal	0.6
p_DistantCaCure	0.04
p_DistantCaDx	1
p_DistantCaMort	0.55
p_HiRiskPolyp	0.03
p_HiRiskPolyp_To_LocalCa	0.017
p_LocalCaCure	0.65
p_LocalCaDx	0.17
p_LocalCaMort	0.23
p_LocalCa_to_Regional	0.22
p_LoRiskPolyp_To_Hi	0.024
p_RegionalCaCure	0.45
p_RegionalCaDx	0.45
p_RegionalCaMort	0.28
p_RegionalCa_To_Distant	0.5
rand	DistForce(1)
[-] Global Values	
uniform_rng	Dist(1)
[-] Tracker Defaults	
{T} Ca_cured	0
{T} Ca_death	0
{T} Ca_dx	0
{T} Ca_stage	0
{T} Polyp_distal	0
{T} Polyp_hi_risk	0

Simulation Model

- Markov node:

- Initialize each individual – starts with polyp, but location/type varies

- **Tracker modifications at Markov node happen once**



Prevalent population (existing polyp) [0] [+]

Name	Value
Name	Prevalent population (existing polyp)
[-] Markov Information	
Term	_stage = 20
[-] Tracker Modifications	
{T} Polyp_distal	rand<p_Distal
{T} Polyp_hi_risk	rand<p_HiRiskPolyp

- Could instead have used a temporary state (“Initialize”), with chance nodes (but easy to avoid in this case)

Simulation Model

- Why {T} tracker = “rand < probX”?
 - Mimics a chance node/event
 - Different techniques for this, but basically need to get hold of a random number
 - “rand” variable generates a random number, which is compared to an event probability
 - Versus letting a chance node do this automatically
 - Equivalent to: $\text{If}(\text{rand} < \text{probX}; 1; 0)$

Prevalent population
(existing polyp)

```
--- Tracker Modifications
{T} Polyp_distal=rand<p_Distal
{T} Polyp_hi_risk=rand<p_HiRiskPolyp
```

Distribution Properties

Index: 1

Name: uniform_rng

Description: Multi-purpose [0 to 1]

Distribution Type: Uniform, Real-numbered parameters,

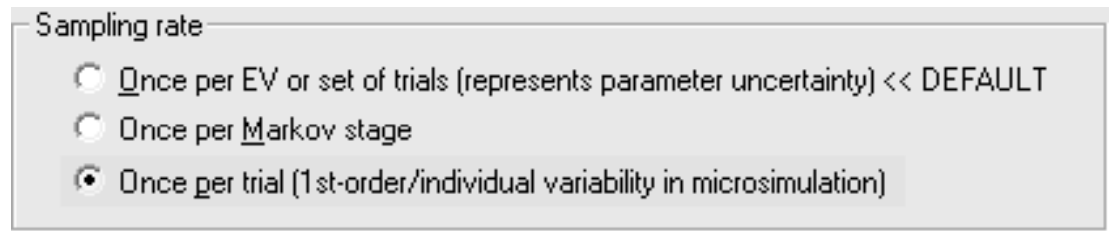
Change Distribution Type...

1st-Order Distribution Notes

- Details: “rand” and uniform distribution
 - rand = DistForce(1)
 - Any reference to “rand” forces a new random number (0 to 1) from uniform distribution
 - So, in this case, **sampling rate** property not relevant
 - Because **not** referring to it as “Dist(1)” or by its name
- One caveat with DistForce(n):
 - Ignore MCS text report column for Distribution #n because it won't reflect all forced sample values

1st-Order Distribution Notes

- Typical patient-level distributions
 - Using Dist(n) syntax; **sampling rate** does matter...
 - “Once per trial”



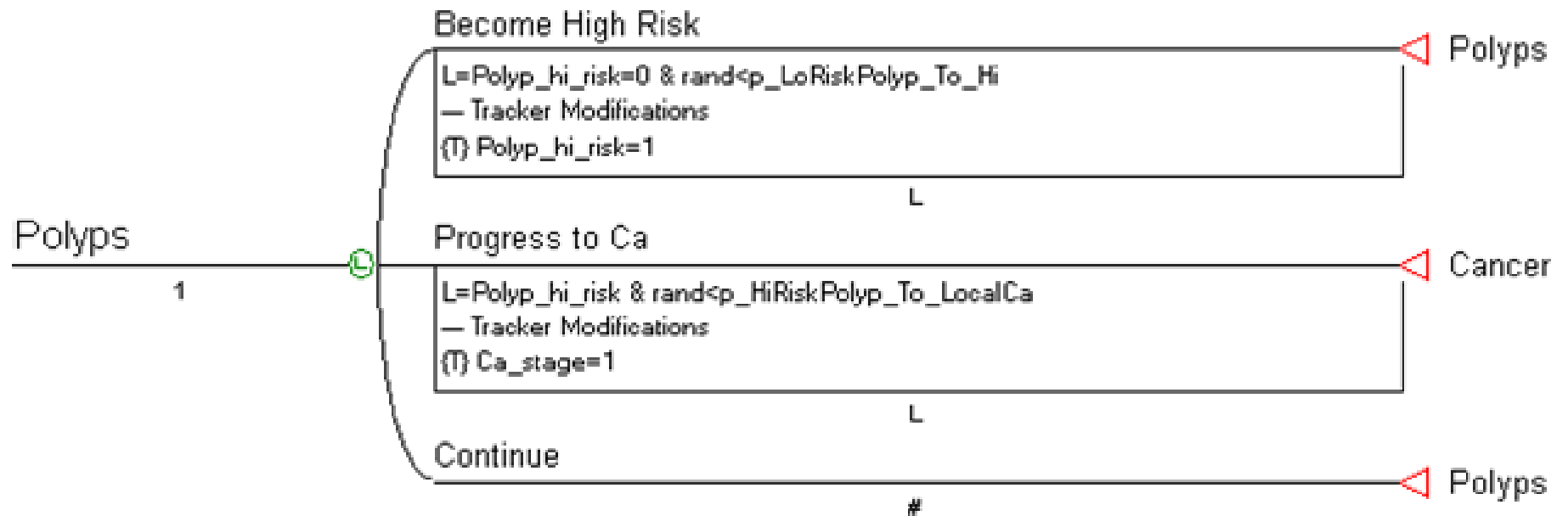
- Either use built-in distribution (Normal, gamma, etc.)
 - Or use Table-type distribution
 - “pmf” – probability mass function (probabilities sum to 1.0)
 - Or (more flexible): use Uniform distribution as table lookup
 - “Index” column describes cumulative distribution (0 to 1)
 - E.g., sample “failure time” (reverse of survival curve)
 - Uniform number (0 to 1) picks row; get value

1st-Order Distribution Notes

- Patient-level distributions (“Per-trial”)
 - In a population model, use to randomly assign patient characteristics, e.g.:
 - distBodyMassIdx: sample from ~ Normal distribution
 - distGender: sample 0=male, 1=female from ~ Binomial[n=1, p=% female]
 - Or bootstrap from Table-type distribution
 - Or use Uniform distribution as table index lookup

Simulation Model

- Back to our model:
 - Assume 100% start in Polyps state



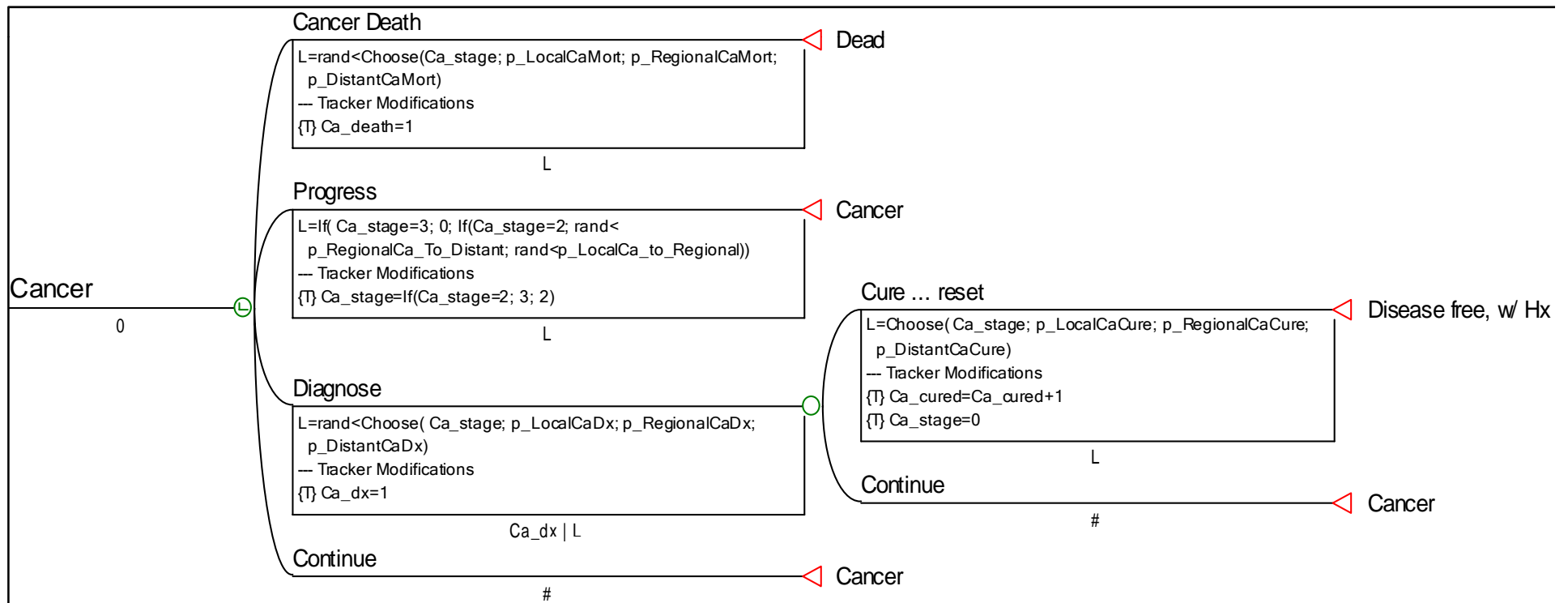
- “Virtual chance nodes” in each logic test “L”
 - **&** = “AND” – means both expressions must be true
 - Polyp change to high risk (chance only if **low** risk)
 - Progress from polyp to cancer (chance only if **high** risk)

Simulation Model

- Logic node:
 - Evaluate top branch first
 - If true, move to that branch
 - Otherwise move to next lower branch
- Polyps state events:
 - More references to “rand”
 - Like Markov node initialization of polyp location/type (previous slide)
 - In Polyp change/progression events, references to “rand” each force a new random number, **to** compare to corresponding event probabilities
 - Logic branch taken if $rand < Prob$ is true

Simulation Model

- Cancer state
 - Cohort model would have multiple Cancer states (local, etc.) each with different chance nodes
- Simulation model uses: {T} Ca_stage = 1, 2, or 3



Simulation Model

- Cancer state events
 - Again, “rand” used numerous times
 - Each logic test “L” embeds a **successive** chance event (don’t need coherence between L’s)
 - Logic conditional on {T} Ca_stage, here using Choose() or If() functions (next slide)
 - Chance nodes replaced by complex expressions
 - Cancer death; cancer stage progression, cancer cure ... up to 3 more random numbers drawn (only 1 if death occurs)
 - Note: *conservative* ordering of logic node branches
 - Death checked first, then Progress, then Cure

Simulation Model

- Cancer state events
 - Choose(*selection*; option1; option2, option3...)
 - Use *selection* (≥ 1) to pick *optionN*
 - If(condition; doIfTrue; doIfFalse)
 - If the condition is true, do the first option
 - Otherwise do the second option
 - Can nest if functions
 - If(cond1; doIfTrue1; if(cond2; doIfTrue2; doIfFalse2))
 - “|” = “OR”:
 - Vertical bar means either “Ca_dx” or “L” must be true

Simulation Model

- Notes:
 - Progression **could** depend on **time in state** (since individual's cancer progressed)
 - Would need {T} Ca_local_yrs ... etc.
(more efficient than a tunnel in simulation)
 - Could easily track/count more events:
 - Time spent in Polyps and/or Cancer
 - Will do this with Global matrix instead, later



Debugging/Reports

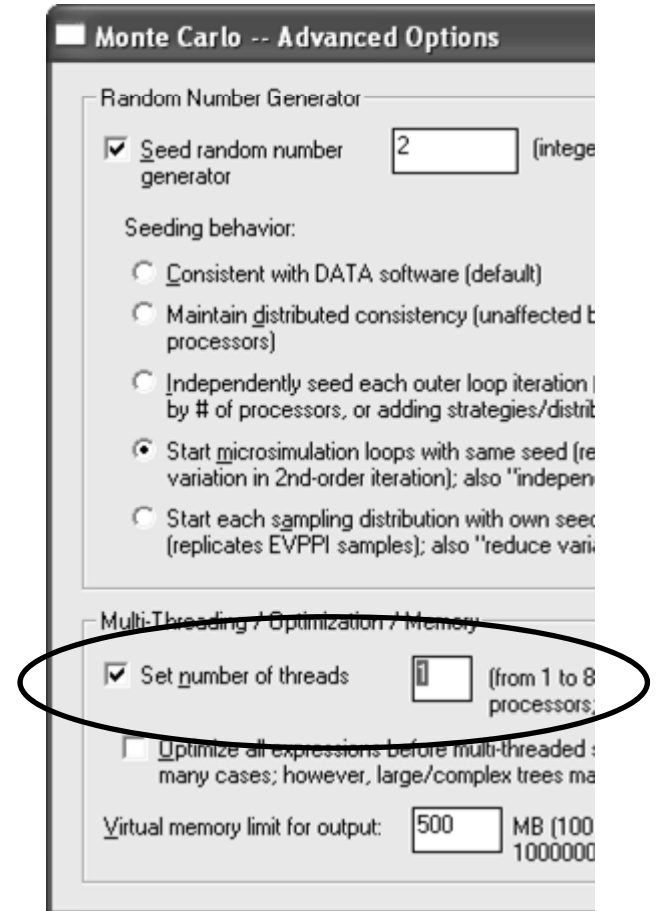
- What to do when your model actually starts running? (Either a complete simulation, or just a few trials before error.)
 - Hopefully you planned for this moment
 - Debugging
 - Reports
 - We'll look at custom reports via **Global matrices**
 - Calibration
 - Sensitivity analysis (simple and PSA)

Debugging

- Error handling and/or Debugging preferences
 - Use the Debug pane to display calculations at time of error or all calculations
 - Some preliminary/temporary setting changes for debugging:
 - Seeding
 - Threading

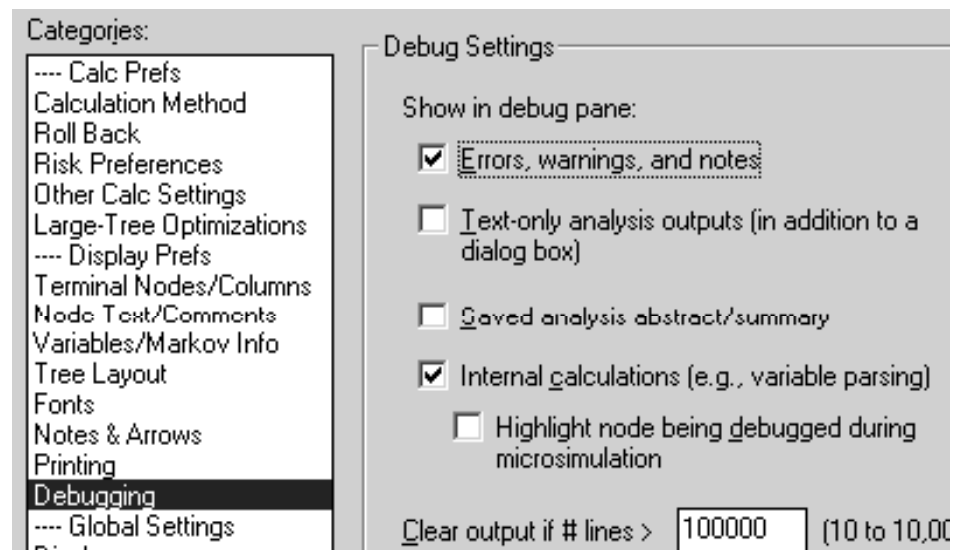
Debugging

- Seeding. Why?
 - Helpful to get same trials each time while looking for issue
 - Helps if timing of error is predictable
 - Or you may be repeatedly running a short microsimulation looking for something to change
- Single thread. Why?
 - If using verbose debugging (later slide), threads interrupt each other, output gets uglier



Debugging

- Debugging preferences/options:
 - Internal calculations...
 - If errors stop calculations, use this to see calculations up to error
 - Errors, warnings...
 - Easier to track messages (i.e., as text instead of in dialogs)

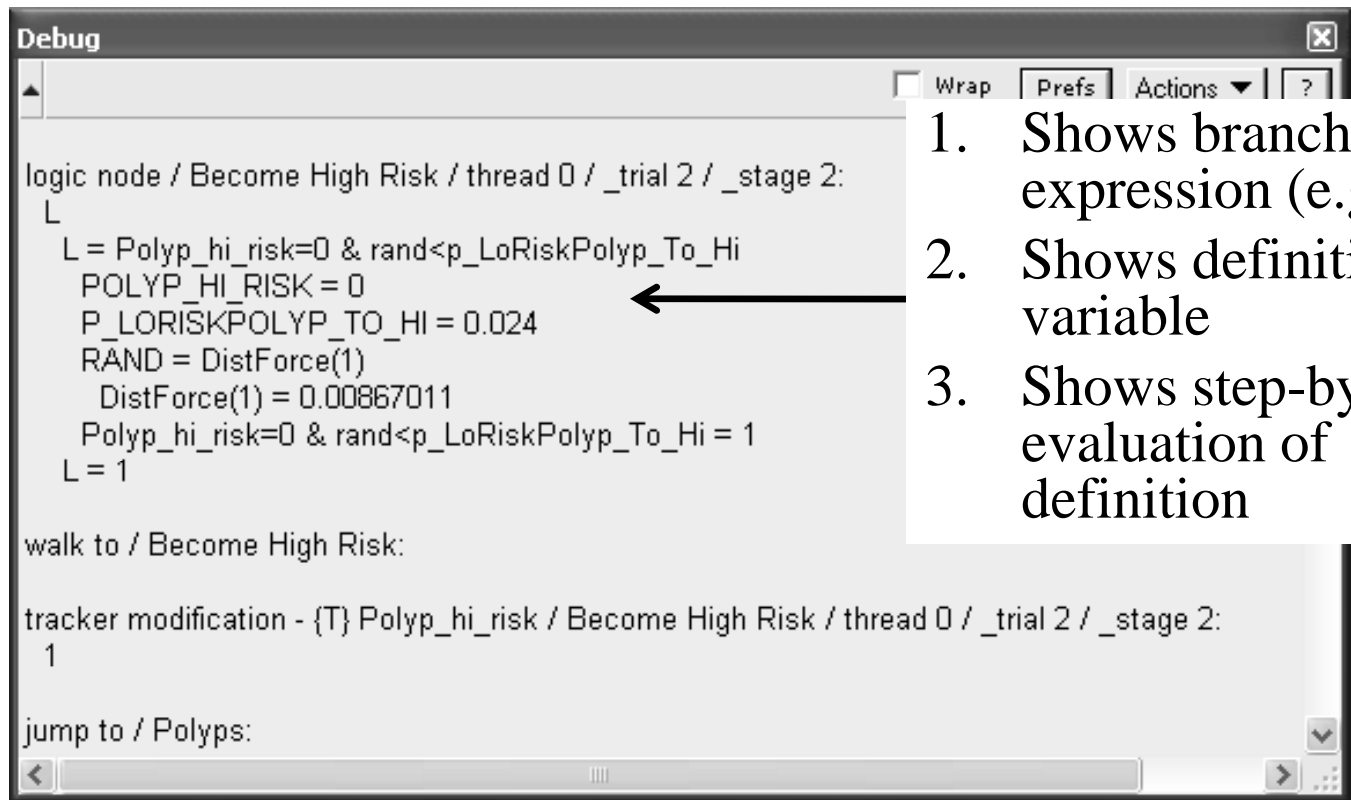
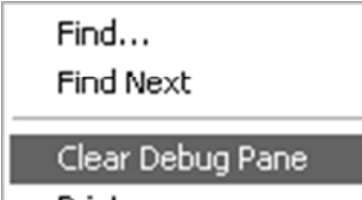


Debugging

- Running debugging simulations
 - Use minimum number of trials (e.g., 1 or a few) initially. Output is VERY verbose.
 - You can also turn debugging output on/off **during** simulation (e.g., if you know error doesn't occur until iteration 5000)
 - Manually: press F11 after simulation starts; can also use “Pause” to hold simulation while changing preference
 - Dynamically: Expressions evaluating Debug(“1”) or (“0”)

Debugging

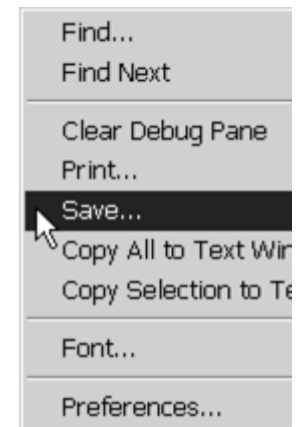
- Example:
 - Can “Clear” first to avoid confusing new output with old



1. Shows branch expression (e.g., “L”)
2. Shows definition, if variable
3. Shows step-by-step evaluation of definition

Debugging

- Debug pane options
 - Move/resize/dock pane
 - Basic word search with Actions > Find
 - Searched for “L = 1” (with spaces) on previous slide
 - For real searching, use Actions > Save...
 - Open saved text file in Word, or other full-fledged text editor



Debugging

- After you are done, turn off debugging
 - Calculation debugging slows calculations down at least 5x or more
 - Seeding can hide variability/uncertainty

Reports

- The obvious: examination of regular outputs
 - Text report
 - Dump into Excel or other data processing environment
 - Statistics report
 - Look for obvious problems (maximum tracker value too high, minimum value too low)
 - Note last row – “Sum” tracker stat interpret as *count* if a binary tracker (0 or 1)
 - Distribution graphs
 - Useful for checking/interpreting count-type trackers

Reports

- Tracker variables – What is reported?
 - If **1-dimensional** simulation (trials only):
 - Reports **final** value of trackers for each individual.
 - Does not show **intermediate** values of trackers
 - Example: If cancer cured, {T} Ca_stage reset to 0, and trial “forgets” if cancer was local, regional.
 - Could add trackers to keep such details if needed
 - E.g.: {T} Cured_1st = 1, 2, or 3 (for local, etc.)
 - Or:
 - A **Global matrix** (or N matrices) can be used to keep **much** more detail without having to manage more trackers...

Reports

- Simulation – What is reported?
 - Unlike cohort analysis, simulation does not report changing distribution among states over time (State Probs graph in Cohort output)
 - Couldn't reasonably attempt with trackers
 - However...
 - A **Global matrix** (or N matrices) can be used to keep **much** more detail without having to manage more trackers...
 - Manipulation of Global matrix (e.g., using Count) in Excel to get charts

Reports

- Tracker variables – What is reported?
 - What if **2-dimensional** simulation (PSA around a microsimulation)?
 - Not reporting individual trial results (e.g., tracker values) if running > 1 trial per sample (as you have to for PSA)
 - Each outer, sampling loop iteration reports summary for a microsimulation (average of n trials' tracker values)
 - However, if need more than just mean of trackers, for example standard deviation or list of individual values...
 - A **Global matrix** (or N matrices) can be used to record per-trial results (e.g., into separate rows, columns, or N matrices for any/all sample iterations)



Using a Global Matrix

- Global Matrices – more flexibility than trackers
 - **Efficiently** store and retrieve any data at any time during analysis
 - Output to text file or via Excel during or after analysis
 - Unlike trackers (which are private to current trial), global matrices are **public**:
 - Store population-level information
 - Share information between individuals (e.g., model limited system resources)
 - Matrices are more dynamic than tables, and without special index column or missing-row behavior
 - Density of data – **GlobalN(index; row; column)**
 - Use like array of trackers (or array of arrays of arrays):
 - Matrix **index** is one dimension
 - **Row** and **column** of matrix cells two more dimensions

Using a Global Matrix

- GlobalN() and Global() overview – inputs
 - To store value in matrix “cell”
 - GlobalN(Index; Row; Column; NewValue)
 - Use where NewValue used in a calculation (tracker, reward, probability, distribution parameter, etc.)
 - Returns NewValue (i.e., wrap an expression with function)
 - Automatically resizes (skipped cells filled with zeros)
 - See Ch. 21 for special options/syntax

Using a Global Matrix

- GlobalN() and Global() overview – outputs
 - To retrieve matrix cell value
 - GlobalN(Index; Row; Col)
 - Expression returns the existing value in matrix
 - **Error** if matrix *n* undefined, or row/column out of range
 - See Ch. 21 for special options/syntax
 - To output entire matrix
 - Text file: GlobalN(*n*; *suffix*)
 - Saved to same location as tree: “c:\...\myTree_Global_*suffix*.txt
 - EXCEL Module: Command(“Excel”;”ExportGlobalMatrixN”; n)

Using a Global Matrix

- v2008:
 - GlobalNIncr(n; r; c; incr)
 - Simultaneously increment cell and retrieve value
 - Equivalent to: GlobalN(n; r; c; incr+GlobalN(n; r; c))
 - If “incr” omitted, “1” assumed
 - Python user-defined functions access matrices as objects:

```
def myFunction( x, y):  
    m1 = treeage.getGlobalMatrixN(1)  
    m2 = treeage.getGlobalMatrixN(2)  
    a = m1.getElement(x,y)  
    b = m2.getElement(x,y)  
    return a+b
```

Using a Global Matrix

- Trackers and GlobalN calls
 - In our example, we use an extra tracker to force calls to update Global matrix:
 - {T} doGlobal = GlobalN(1; 1; 1; 1)
 - Reporting of doGlobal not necessarily relevant
 - Can place multiple **GlobalN()** calls in tracker modification (or in termination condition, rewards, etc.)
 - .. = GlobalN(1;1;1;1)+Global(2;1;1;1)

Using a Global Matrix

- In our model, set up a matrix to store state by stage
 - Polyp = 1, Cancer = 2, Dis. Free w/ Hx = 3, Death=4
 - Put trial number (or other index) in first column, using `_trial` counter (or `_sample`)
 - `{T} doGlobal = GlobalN(1; _trial; 1; _trial) + ...`
 - Not required; but may make reading a matrix easier
 - Put state at each stage via...
 - `{T} doGlobal = ... + GlobalN(1; _trial; _stage+2; 1)`
 - “`_stage+2`”
 - Skip column 1 – used for `_trial` number
 - Account for `_stage 0` – cycle 1
 - Placed death state accounting in transition
 - `{T} doGlobal = GlobalN(1; _trial; _stage+3; 4)`
 - Tracker modifications not executed in absorbing state

Using a Global Matrix

- In our model, set up a matrix to store state by stage
 - Polyp = 1
 - Cancer = 2
 - Dis. Free w/ Hx = 3
 - Death = 4
- Put trial number (or other index) in first column, using `_trial` counter (or `_sample`)
 - `{T} doGlobal = GlobalN(1; _trial; 1; _trial) + ...`
 - Not required; but may make reading a matrix easier

Using a Global Matrix

- Store state at each stage
 - {T} doGlobal = ... at each state branch
 - At State 1: GlobalN(1; _trial; _stage+2; 1)
 - At State 2: GlobalN(1; _trial; _stage+2; 2)
 - At State 3: GlobalN(1; _trial; _stage+2; 3)
 - At State 4: Absorbing state
 - Statement would not be executed
 - Column expression: “_stage+2”
 - Skip column #1 used for _trial counter
 - Skip column #2 because _stage counter starts at 0
 - Placed death state accounting in transition
 - {T} doGlobal = GlobalN(1; _trial; **_stage+3**; 4)

Using a Global Matrix

- Export Global matrix for review
 - See the following definition at root node...

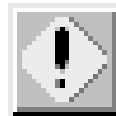
_node_action =

```
Command("Excel"; "ExportGlobalMatrixN"; 1; "";  
        "StateTransitions")           [to Excel]  
GlobalN(1;1)                          [to Text]
```



- Clicking toolbar's yellow diamond button to evaluate `_node_action`, and executes its commands

Using a Global Matrix

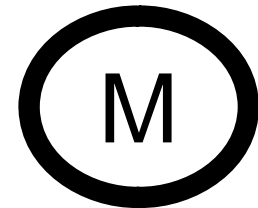
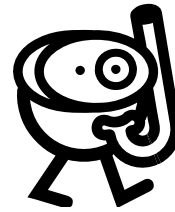
- Export the Global matrices to review movement among the states
 - Click Node Action button on toolbar 
- Matrix data
 - Each trial has a row
 - State membership at each stage is reflected in successive columns to the right

Using a Global Matrix

- Lots of other uses
- Even outside of microsimulation

Summary

- Questions?
 - Trackers
 - Distributions
 - Debugging
 - Global matrices



- Submit questions via GotoMeeting Q&A or send e-mail to [<support@treeage.com>](mailto:support@treeage.com)

Webinar Feedback & Materials

- Please provide feedback
 - Via GoToMeeting survey
- All webinar materials available at
 - <http://server.treeage.com/treeagepro/training/webinars.asp>