



Tech Note #13

Discrete Event Simulation

**Microsimulation, Parallel Trials,
and “Queuing” Models in
TreeAge Pro 2008**

TreeAge Software, Inc.



Outline

- Simulation and Discrete Simulation
- Microsimulation in TreeAge Pro
- Illustrate Parallel Trials Applications:
 - Resource Competition, other DES aspects
 - Modeling Open/Dynamic “Populations”
- Compare dynamic simulation models to dynamic cohort models



Discrete Event Simulation

- We'll look at options for discrete simulation in TreeAge:
 - Sequential (independent) microsimulation trials is traditional ...
 - ... But parallel (interacting/competing) trials an option since v2004
 - “Closed population” of indeterminate size is traditional ...
 - ... But parallel trials allow for open population and dynamic sizing (in v2008)
 - Note: Dynamic/open option with Markov **cohort** (i.e., non-simulation) models available since '04
- But first, the background...



Discrete Event Simulation

- Simulation is, for our purposes:
 - Using a mathematical computer model,
 - to imitate a system (e.g., a patient and/or population),
 - so that we can perform experiments.
- In TreeAge Pro, called “microsimulation”
 - Traditionally used to simulate variability in individual patients (or trials), get mean and other statistics as part of decision analysis
 - Slides don’t cover very basics of microsimulation
 - Presentation looks at microsimulation modeling/analysis in the context of “Discrete Event Simulation”



Discrete Event Simulation

- More Resources (beyond these slides):
 - Many useful simulation model publications, e.g.:
 - Start with:

Le Lay, et al. “Can discrete event simulation be of use in modelling major depression?” Cost Eff Resour Alloc. 2006; 4: 19.

<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1762026>

Thorough discussion of Discrete Event Simulation and microsimulation in TreeAge Pro

- BRAM model, NHS monograph 811

Shown later in the slides... monograph includes thorough model discussion



Discrete Event Simulation

- Discrete event = discrete change model
 - “**State** changes only when **events** occur, [in turn causing] **time** advances.”
Discrete Event Simulation: A Practical Approach (1992)
 - **Time** may be modeled either as stochastic jumps or fixed increments (i.e., cycle length)
- *Continuous* background change models are as a rule not a feature in computer simulations



Discrete Event Simulation

- Discrete simulation defined...
 - Discrete change (vs. continuous vs. hybrid)
 - Stochastic (vs. deterministic)
 - Dynamic/time-dependent (vs. static)
 - *This defines microsimulation, too*
- DES software tools... is TreeAge one?
 - Discrete events ← microsimulation
 - Parallel processes ← can run “parallel trials”
 - Dynamic creation of “objects” ← v2008 enabled
“on-the-fly” creation of trials



Discrete Event Simulation

- Simulation software...
 - Simulation languages evolved over 40 years
 - GPSS, Simula67, ...
 - Graphical environments specific to DES, last 20 years
 - Arena, ...
- TreeAge Pro
 - Simulation evolved in context of decision analysis (CEA, health economics, PSA, etc.)



Discrete Event Simulation

- Terminology of DES-only (i.e., Arena-like) tools or languages:
 - System model; Entities (permanent or transitory); Entity attributes; Activity (cause change); Transactions;
 - Scheduling: Activity scanning; Event orientation; Process orientation. Time management: Periodic scan (fixed increment); Event scan (variable increment).
- TreeAge microsimulation
 - Couched in Markov terms, instead of traditional discrete simulation terminology
 - But not limited to Markov/semi-Markov assumptions



Discrete Event Simulation

- Microsimulation is term from economics originally
 - Model “agents” individually
 - Simulate changes in **state**, attributes, behavior
 - Examine aggregate & average outcomes
 - Also examine distribution
 - Agents **sequentially** processed ...
 - So, in theory/strictly defined, doesn't deal with system/population interactions
 - To model agent (i.e., entity) interactions – for example, resource competition, force of infection – need to run microsimulation trials in parallel



Discrete Event Simulation

- What are options for discrete simulation in TreeAge?
 - Sequential (independent) trials are traditional
 - Many such published patient-level simulation studies
 - Parallel (interacting) trials newer option ('04)
 - Closed population/static size is traditional
 - Open population, dynamic sizing in microsimulation very new ('08)



TreeAge Simulation Models

- TreeAge simulation models, standard elements:
 - Markov node (to create event loop)
 - Monte Carlo microsimulation (1st-order trials)
 - Trackers (to keep track of private state)



TreeAge Simulation Models

- Options (beyond standard elements):
 - **Parallel trials** to allow interaction...
 - 1st-order **distributions** to sample time-to-event, patient state, etc.
 - **GlobalN()** matrixes for global/public state
 - **Non-coherent probabilities** tree preference for open/dynamic population (v2008)
 - **{T} _CLOCK** for synchronizing trials if stochastic time advance (v2008)
 - **Python**, user-defined functions for complex algorithms (e.g., search/sorting)
 - <http://www.treeage.com/support/python.html>



Discrete Event Simulation

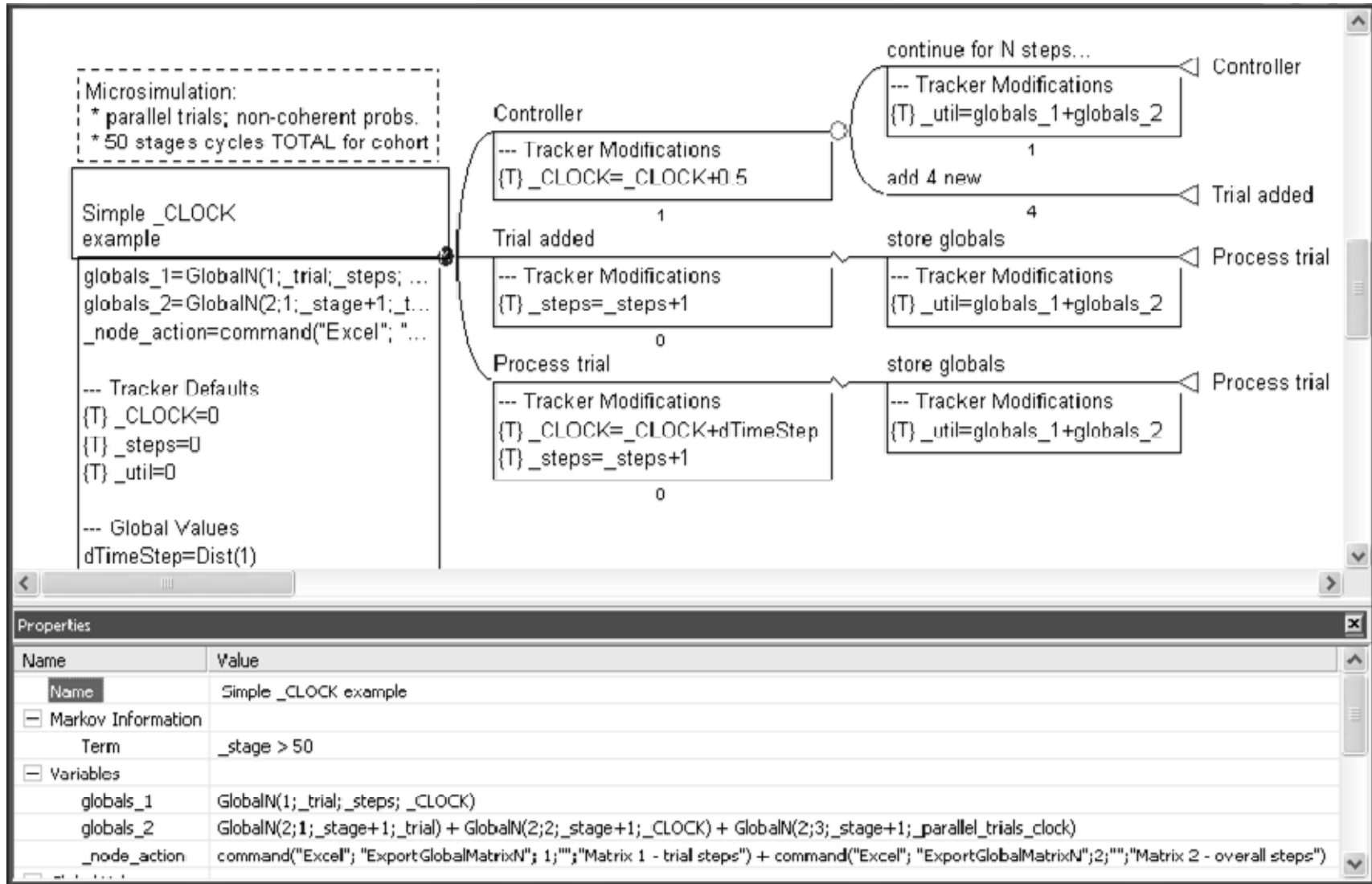
- Examples (going beyond just using trackers):
 - “_CLOCK example.tre”
 - Simplistic, abstract model (doesn't solve a problem)
 - Use to describe core DES-related features **in detail**
 - “Organ allocation.pkg”
 - **Quick** illustration of a more complex model than simple “_CLOCK example”, plus a realistic issue
 - Download to dig into the details and try simulations
 - “BRAM simulation.pkg”
 - Replicates published, well-documented model from NHS study; regular microsimulation
 - See the NHS monograph for extensive model description
 - Trackers/distributions for time-to-event simulation (i.e., variable cycle length, no use of _stage)



Discrete Event Simulation

- Example #1:
 - “_CLOCK example.tre”
 - Abstract model, doesn't solve a problem.
Only function is to illustrate...
- Using subset of mentioned sim tools, e.g.:
 - Microsimulation, individual-level simulation
 - Trackers for private state info (like time/age)
 - Distributions, for sampling time-to-event ...
 - Effectively stochastic “cycle length”
 - Parallel trials for competition/interaction
 - Using {T} _CLOCK tracker to synchronize trials
 - GlobalN() for public/shared state info (versus *private* trackers)
 - Also for more detailed reporting
 - Non-coherent probabilities, for open/dynamic population

Discrete Event Simulation



_CLOCK example



Discrete Event Simulation

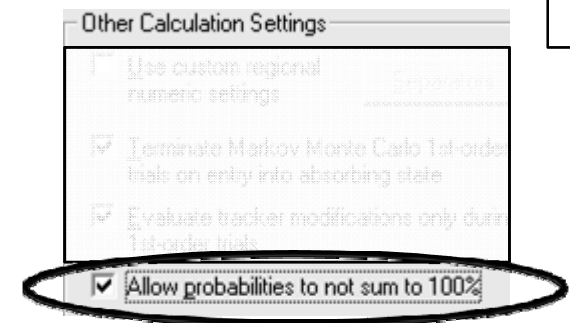
- Key model element #1a:
 - Cycle length: `_stage` **not** used to keep patient time
 - Fixed cycle length not an assumption in our model
 - “Time” instead progresses independently for each trial in stochastic jumps (e.g. based on sampled time-to-event)
 - Patient time stored in a tracker
- Key model element #1b:
 - Patient time stored in tracker specifically called `{T} _CLOCK`, so that trials are synchronized ...
 - ... Only works if trials run in parallel (versus default/sequential)
 - So, if parallel trials and `{T} _CLOCK` exists:
 - Trial with lowest `_CLOCK` time runs next cycle/event
 - Use a different `{T}` tracker name for patient time to avoid synchronization (e.g., for speed)
 - NOTE:
 - Patient-time tracker **not** required with parallel trials
 - Can still be `_stage`-based

`_CLOCK` example

Discrete Event Simulation

- Key model element #2:
 - “States” not really health states, rather events (for system or individual) or event generators
 - E.g., trial #1 in “Controller” state is not an individual

- Key model element #3:
 - Dynamic creation of (parallel) trials using **non-coherent** transition probabilities
 - This is purpose of “Controller” state
 - Although initial probabilities are coherent in this example, they don’t need to be
 - In parallel dynamic trials, sum of initial probabilities determines starting # of “trials” (integer $>- 1$)

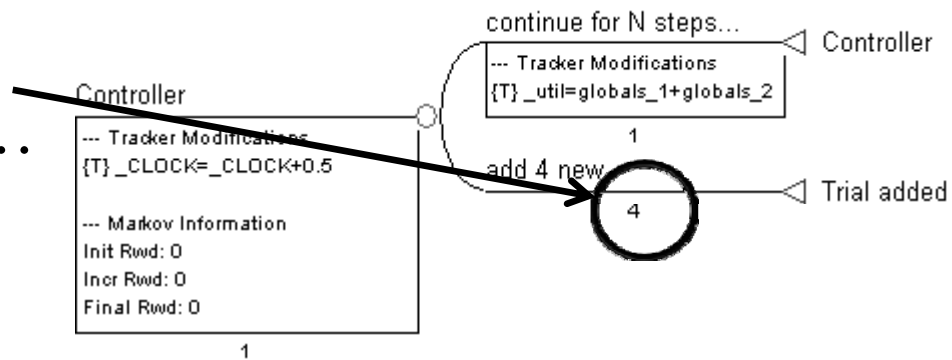


Discrete Event Simulation

_CLOCK example

- “Controller” state
 - Initial probability = 1, and trial #1 starts here at initial cycle (t_0)
 - If init. prob. instead = 2, then 2 trials would start here
 - Transition probabilities are non-coherent:
 - $p_{\text{Continue}} = 1 + p_{\text{Add}} = 4 \rightarrow p_{\text{Total}} = 5$
 - “continue...” branch
 - Current trial always follows top/first 1.0 of p_{Total}
 - **So trial #1 stays in “Controller”**
 - $\{T\} _util \rightarrow$ performs a reporting/debugging function

– “add 4 new”
generates trials...





Discrete Event Simulation

- “Controller” state (continued)
 - “add...” event injects new trials
 - Parent trial’s {T} `_CLOCK` increment effectively means trials added every 0.5 time units
 - New trial’s {T} `_CLOCK` initialized equal to parent trial’s
 - New trial will start its random walk at “add 4 new” (e.g., executing any tracker modifications, or subsequent chance events)

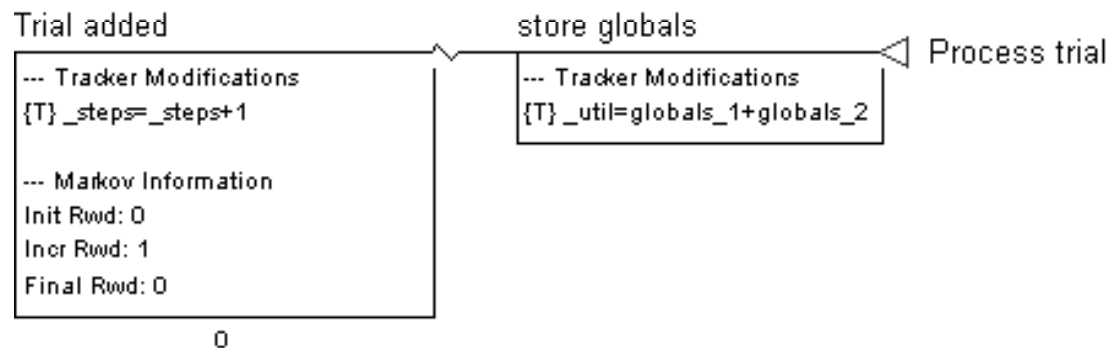
`_CLOCK` example



Discrete Event Simulation

- “Trial added” state
 - Zero initial probability (no trials here at t_0)
 - New trials sent here; temporary state
 - $\{T\}$ _steps \rightarrow not a time counter
 - Counts # cycles (events?) for each trial/individual
 - But cycles have no particular length; not clinically meaningful, more for debugging
 - $\{T\}$ _util: stores clock “history” in Global matrix

_CLOCK example

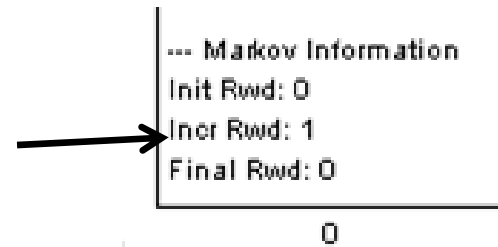


0

Discrete Event Simulation

- “Trial added” state (continued)
 - Incremental reward doesn’t do much

- Each trial added gets a single incr. reward =1



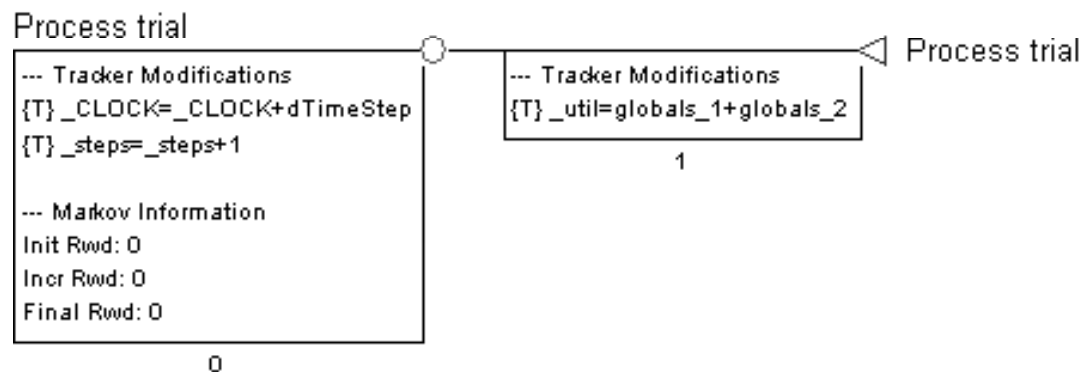
- Total final cohort reward equals size (# trials added)
- State is temporary; every trial continues to “processing”



Discrete Event Simulation

_CLOCK example


- “Process trial” state
 - Trial with lowest `_CLOCK` time runs a cycle/event
 - At start, each trial samples its `dTimeStep` (i.e., “cycle length”); `_CLOCK` advances by `dTimeStep`
 - Effectively running in parallel, but simulation processing one at a time
 - In example, termination based on `_stage` (which is NOT per trial, but total cycles across all trials)



Discrete Event Simulation

- Reporting/debugging trials:
 - Uses Global matrices to mimic report similar to full cohort trace

_CLOCK example

Variable	Definition	Notes
globals_1	GlobalN(1;_trial;_steps; _CLOCK)	Matrix #1 records trials' _clock changes
globals_2	GlobalN(2;1;_stage+1;_trial) + GlobalN(2;2;_stage+1;_CLOCK) + GlobalN(2;3;_stage+1;_parallel_trials_clock)	Matrix #2 record per _stage: _trial; trial's _CLOCK; low _CLOCK
globals_2_titles	Command("GlobalMatrixN"; 2; "ColumnLabels"; "Stg0";"Stg1";"Stg2";"Stg...")	Set some column titles
_node_action	command("Excel"; "ExportGlobalMatrixN"; 1; ""; "Matrix 1 - trial steps") + command("Excel"; "ExportGlobalMatrixN"; 2; ""; "Matrix 2 - overall steps")	Output matrices (to Excel in this case). Linked to: 

Discrete Event Simulation

- Reporting/debugging output to Excel:

- Matrix #1:

- No column titles here
 - Row for trial 1 (controller) isn't filled in; real trials each record their `_CLOCK` change
 - Groups of 4 new trials added starting at $t_{0.5}$ and every 0.5 time units
 - Their `_CLOCK`'s indicate their sampled "cycle length"

	A	B	C	D	E
1	C1	C2	C3	C4	C5
2	0	0	0	0	0
3	0.5	1.077355	1.65471	2.232065	0
4	0.5	0.976082	1.452165	1.928247	2.40433
5	0.5	1.29464	2.08928	0	0
6	0.5	0.979427	1.458855	1.938282	2.41771
7	1	1.593773	2.187545	0	0
8	1	1.496316	1.992633	2.488949	0
9	1	1.617029	2.234059	0	0
10	1	1.745709	2.491418	0	0

_CLOCK example

- Matrix #2:

- Under titles, first row shows order in which trials processed

	A	B	C	D	E	F	G	H
1	Stg0	Stg1	Stg2	Stg...	C5	C6	C7	C8
2	1	1	2	2	3	3	4	4
3	0.5	1	0.5	1.077355	0.5	0.976082	0.5	1.29464
4	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5



Discrete Event Simulation

- Global matrices:
 - Can also use for public/shared “state”
 - Next example demonstrates this (in context of competing for limited resources, e.g. organ transplant)



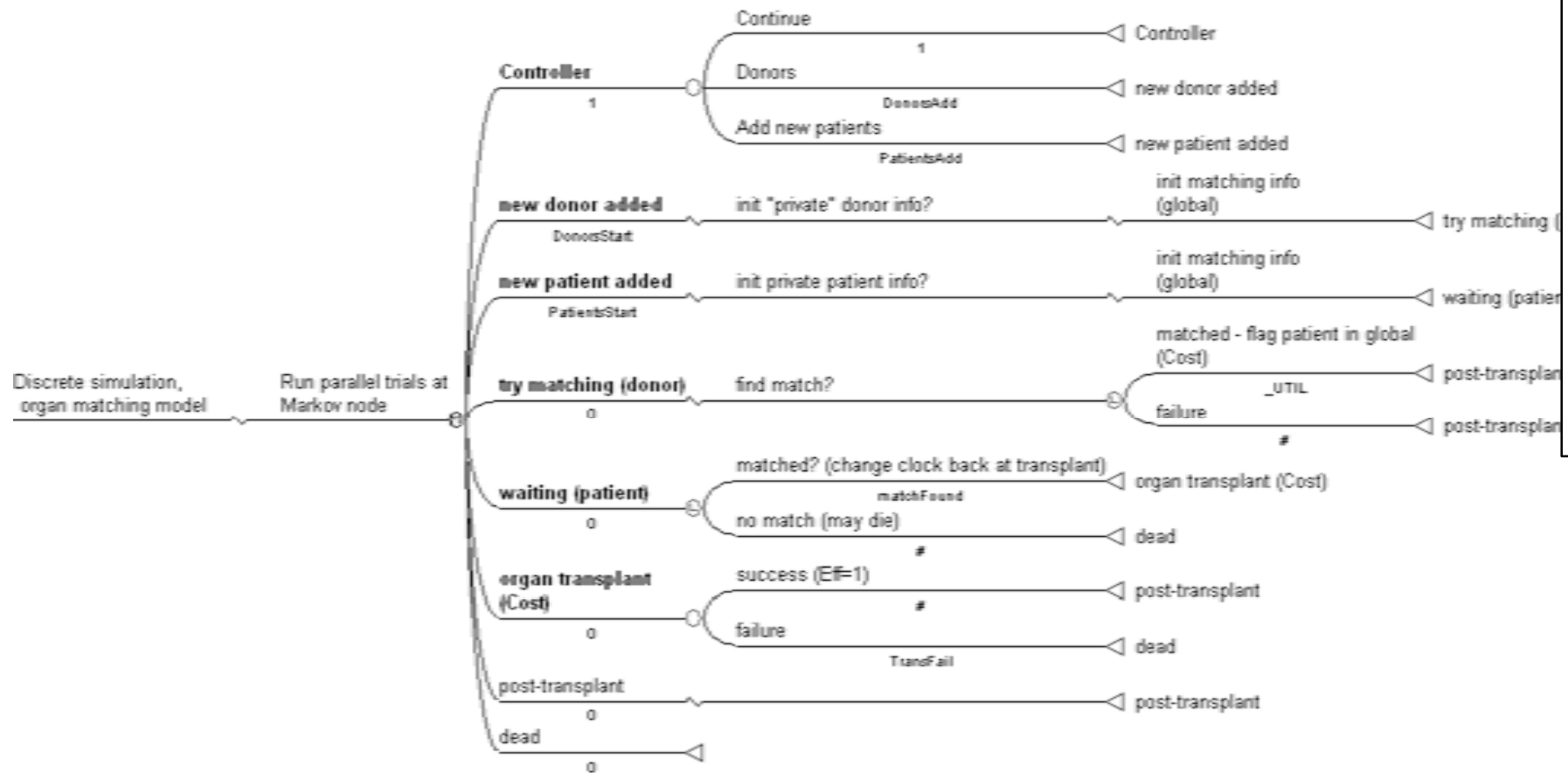
Discrete Event Simulation

- Queueing/competition example
 - “Organ allocation.pkg”
 - Less abstract model, real problem...
 - Uses all the mentioned “DES” tools, e.g.:
 - Microsimulation for trackers (private)
 - Parallel trials, for competition/interaction
 - {T} _CLOCK synchronization
 - GlobalN() for public, global state
 - Non-coherent probabilities, to allow for open, dynamic population
 - Python (e.g., for sorting/prioritizing list of recipients by characteristics)



Discrete Event Simulation

Organ allocation example





Discrete Event Simulation

- New features (v2008):
 - Dynamic-sized microsimulation
(based on existing parallel trials feature)
 - Illustrates non-coherent initial probabilities
 - “Managed” {T} _CLOCK tracker variable, for triggering events
 - Trial ignored until “system time” catches up to individual _CLOCK
 - Other supporting changes:
 - New functions; keywords; efficiencies added to GlobalN() matrixes; Python user-defined functions



Discrete Event Simulation

- Download example
 - Run microsimulation (parallel trials will be default) at the Markov node
 - Click on `_node_action` button to export resulting global matrices/simulation trace outputs to Excel



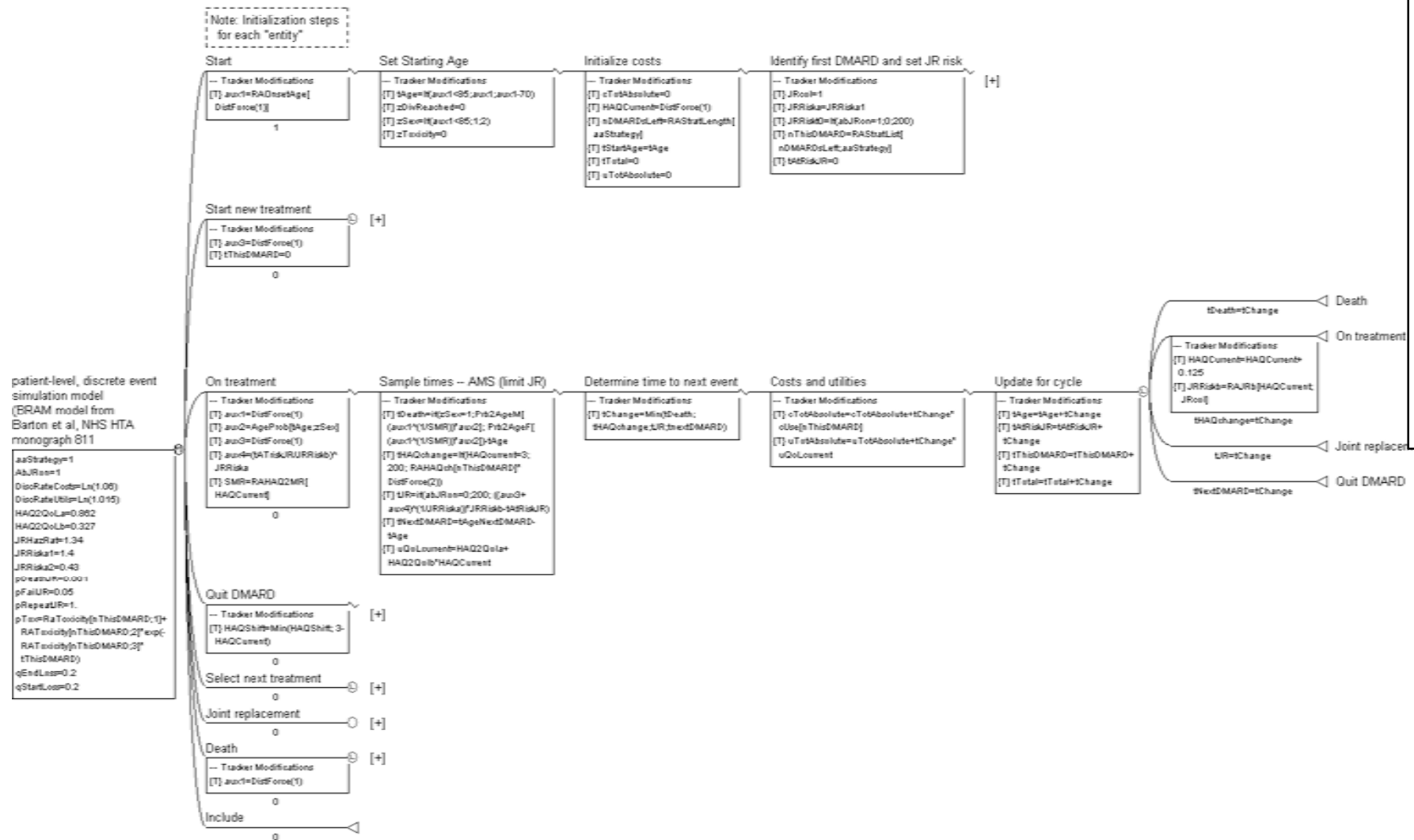
Discrete Event Simulation

- Sequential, event-advance example
 - BRAM (Birmingham UK Rheumatoid Arthritis Model) simulation
 - Study done c.2003 using DATA Pro (pre-TreeAge Pro speed-ups, features)
 - Not parallel trials (no need for “agent” interaction)
- Discrete Event Simulation using microsimulation
 - Competing risks using sampled times to events
 - So, continuous time jumps to next event
 - Keep patient time in a {T} tracker)
- Lots of trackers (in addition to time)



Discrete Event Simulation

BRAM simulation example





Discrete Event Simulation

- See NHS monograph #811 for complete details on their model



Dynamic Cohort Models

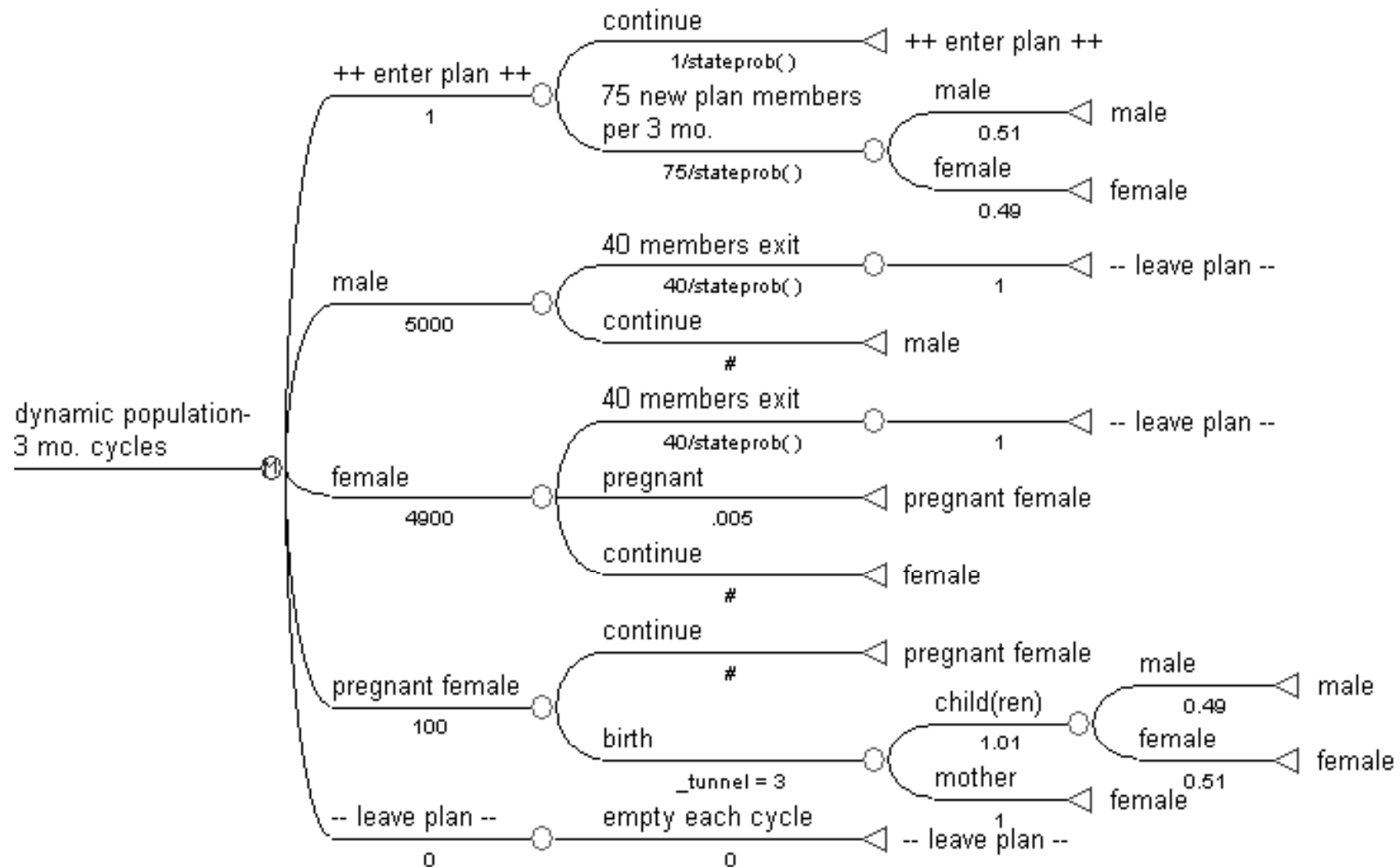
- Switching gears for a moment...
- Using dynamic cohorts models
 - If you don't need {T} trackers, and _stage and fixed cycle length are fine...
 - But you do need:
 - “Agent” interaction; and/or
 - Open/dynamic-sized cohort



Dynamic Cohort Models

- Dynamic cohorts models
 - Epidemiology
 - Modeling infectious disease
 - Use of StateProb(n) function in determining force of infection (based on counts in Susceptible, Infectious, etc.)
 - Budget-impact analysis
 - Modeling changing size of treatment population
 - Dynamic cohort analysis models useful
 - In microsimulation, parallel trials important
 - Use instead of sequential, to support StateProb() function
 - Model dynamic/open population via non-coherent probabilities (parallel trials allow “creation” of trials)

Dynamic Cohort Models



Dynamic Cohort Models

- Model functionality described elsewhere
 - Chapter 35, TreeAge Pro PDF user's manual; *“Modeling dynamic populations during cohort analysis ...”*

Dynamic Markov models: budget impact analysis and infectious disease modeling

This section deals with the use of a dynamic “population” in a cohort analysis model. Starting with TreeAge Pro 2008, similar techniques are also available with Markov microsimulation models using parallel trials; see Chapter 50 for details.

Budget impact analysis and infectious disease simulation

In decision analysis or cost-effectiveness analysis, a Markov node can be used to calculate expected cost and effectiveness for a closed population (or subgroup) of indeterminate size. In a budget impact analysis, however, the goals are different: a modified approach to the Markov cohort may be required, including modeling an open, dynamic population.

Another context where open, dynamic cohorts are relevant is in modeling infectious disease, when force of infection, herd immunity, and other population impact factors may depend in part on the size of the infectious and/or susceptible population (versus a cohort of indeterminate size).

For additional background on budget impact modeling, refer to: <http://www.ispor.org/wp/wp-content/uploads/impact.asp>

Modeling dynamic “populations”

An equivalent option might be to do the multiplication via tree probabilities. This would require that, in every path (probabilistically at the first, a single probability be multiplied by the population size.

It has been noted numerous times in this manual that TreeAge Pro requires that the branch probabilities in situations where it is useful to do so, this restriction can be removed.

TreeAge Pro includes a tree preference to turn off the error checking that normally protects against mistakenly assigning non-coherent probabilities.

It is not recommended that the option to allow non-coherent probabilities be used without careful consideration of its implications and the hazards of turning off probability error checking.

► To disable errors when using non-coherent probabilities:

User Calculation Settings

- Allow non-coherent transition settings
- Prevent Markov Model Calc. To establish or update starting size
- Allow probabilities not sum to 100%
- Exclude branch of value of 0
- Automatically resolve probabilities
- Show terminal node size to act to return path after event if empty

► Choose Edit > Preferences... and select the Other Calc Settings category.

► Check the option labeled Allow probabilities to not sum to 100%.

Modeling dynamic “populations” during cohort analysis

A simple example, the Dynamic Population model from TreeAge Pro’s Tutorial Examples/Healthcare subcategory, is shown on the following page.

This model illustrates the use of non-coherent probabilities to model a growing population (e.g., for budget impact analyses). It highlights two key aspects of using non-coherent probabilities in Markov models:

- discrete initial sizing of the cohort (e.g., 10-25 million) using non-coherent initial probabilities
- population growth (e.g., through births or migration) using non-coherent transition probabilities

In some models, just setting a size on the initial

infection could depend on the number in infected states:

$$p_{inf} = 2 * StateProb(6,10) / StateProb(1,10)$$

In the Dynamic Population example, only population growth is modeled (not disease). The context is a health plan with changing membership. The starting membership is specified with numbers of individuals starting in the states — a total of 10,000 individuals in the example (ignore the “++ entry/exit ++” state). During 10 years of Markov calculations (40 three-month cycles), individuals can be added to the population in two ways:

- entry from other populations, outside the health plan (i.e., new enrollment)
- internal population growth (i.e., births and other changes in family size)



Dynamic Cohort Models

- Final note:
 - Translating a cohort model using a dynamic/open population or StateProb() ...
 - ... Into a microsimulation model:
 - Parallel trials are required, instead of sequential, to support StateProb() function and/or resource competition, time synchronization
 - Can model dynamic/open population via non-coherent probabilities

Summary

- Questions?



- Send an e-mail to support@treeage.com
- Search Google for more TreeAge simulation/DES models...

Webinar Series

- Material originally presented as part of TreeAge Software webinar series
- Webinar schedule and materials available at:

<http://server.treeage.com/treeagepro/training/webinars.asp>