

Discrete Intra-Agent Dynamics: Statecharts

Nathaniel Osgood

MIT 15.879

March 7, 2012



Hands on Model Use Ahead



Load Previous Built [& Provided] Model:
MinimalistNetworkABMModel

Adding "Color" Variable

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: MalariaV2

- Main
 - Parameters
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Presentation
 - Simulation: Main
 - Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
 - Network Modification of SIR AB
 - Main
 - Person

Model:

- Parameter
- Flow Aux ...
- Stock Vari...
- Event
- Dynamic ...
- Plain Vari...
- Collectio...
- Function
- Table Fun...
- Port
- Connector
- Entry Point
- State
- Transition
- Initial Stat...
- Branch
- History St...
- Final State
- Environm...

Console: color - Plain Variable

General

Name: color Show Name Ignore Public Show At Ru

Access: public Static Constant Save in snapshot

Type: boolean int double String Other: Color

Initial Value: BLACK

This is the name of a Java class!

Make sure this is in lower case!

Fill in the type and Initial Value (watch for correct case!!)

Discrete Agent Dynamics

- Frequently we can represent agent behaviour using as transitioning among a set of mutually exclusive and collectively exhaustive states in a “state chart”
- For a given simple statechart, the agent is in exactly one state at a time
- Fixed transitions between states define possible evolution
- The transitions between states occur instantaneously, based on some condition

Add Entry Point of State chart

The screenshot displays the AnyLogic Advanced interface. The main workspace shows a statechart entry point labeled "InfectionStatechart" with a red oval highlighting it. A red arrow points from this oval to the "Name" field in the "InfectionStatechart - Statechart Entry Point" properties window, which is set to "InfectionStatechart".

The associated text is the name of the statechart!

The interface includes a Project tree on the left, a Console and Properties window at the bottom, and a toolbar on the right. The Properties window shows the following configuration:

- General: Name: InfectionStatechart, Show Name: , Ignore: , Public: , Show At Runtime:
- Description: Action: (empty field)

The Problems window at the bottom left shows several error messages:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging statechart entry

The status bar at the bottom indicates "Selection" and "Cursor: X=186, Y=63".

Add in “Susceptible” State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a statechart diagram for an 'InfectionStatechart'. A yellow oval labeled 'Susceptible' is being added to the statechart, indicated by a red highlight and a blue arrow pointing to the statechart's entry point. The diagram also includes a 'color' variable and a 'Person' object.

The left sidebar shows a project tree with the following structure:

- MalariaV2
 - Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
 - TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Unresolved
 - Presentation
 - Simulation: Main
 - Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
 - Network Modification of SIR AB
 - Main

The bottom panel shows the 'InfectionStatechart - Statechart Entry Point' configuration. The 'General' tab is active, showing the name 'InfectionStatechart' and checkboxes for 'Show Name', 'Ignore', 'Public', and 'Show At R...'. The 'Description' tab is also visible, showing an empty 'Action' field.

The bottom status bar indicates the cursor position: Selection | Cursor: X=151, Y=127.

Connect with Entry Point

The screenshot shows the AnyLogic Advanced interface. The main workspace displays a statechart diagram with a state named "Susceptible" highlighted in yellow. A red arrow points from the text overlay to this state. The diagram also includes a variable "color" and a statechart element "InfectionStatechart".

**When this really connects,
The circle should be green
(see tip at end of presentation)**

The interface includes a Project Explorer on the left, a Properties panel at the bottom, and a toolbar on the right. The Properties panel shows the "InfectionStatechart" element with fields for Name, Show Name, Ignore, Public, and Show At Run.

Fill In Code to Color Green when Enter State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a statechart diagram with a state named "Susceptible" highlighted in yellow. A transition arrow points to this state from a transition labeled "InfectionStatechart". A variable "color" is also visible in the diagram.

The Properties panel for the "Susceptible - State" is open, showing the following configuration:

- Name: Susceptible
- Show Name:
- Ignore:
- Public:
- Show At Run:
- Fill Color: Default
- Entry Action: color=GREEN
- Exit Action:

The Problems panel at the bottom left shows several error messages:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Adding in “Infective” State

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a statechart diagram with two states: 'Susceptible' (a yellow rounded rectangle) and 'Infective' (an orange rounded rectangle). The 'Infective' state is highlighted with a red oval. A transition arrow points from the 'Infective' state to the 'Susceptible' state, labeled 'Infections'. A variable 'color' is shown in the workspace. The left sidebar contains a project tree for 'MalariaV2'. The bottom panel shows the 'Properties' window for the 'Infective - State' element, with fields for Name, Fill Color, Entry Action, and Exit Action. The 'Problems' window at the bottom left lists several errors related to state membership and hanging transitions.

Project: MalariaV2

- Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Unresolved
 - Presentation
 - Simulation: Main
- Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
- Network Modification of SIR AB
 - Main

Console: Infective - State

General

Name: Infective Show Name Ignore Public Show At R...

Description

Fill Color: Default

Entry Action:

Exit Action:

Problems

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Set to Color Red when Enter State

The screenshot displays the AnyLogic Advanced interface. The central workspace shows a statechart titled "InfectionStatechart" with two states: "Susceptible" and "Infected". The "Infected" state is highlighted with a yellow fill and a red border. A parameter named "color" is shown at the top left of the workspace.

The Properties window at the bottom right is open to the "General" tab for the "Infected - State". The "Name" is set to "Infected". The "Fill Color" is set to "Default". The "Entry Action" field contains the text "color=RED", which is highlighted with a red oval. The "Exit Action" field is empty.

The left sidebar shows a project tree with the following structure:

- MalariaV2
 - Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
 - TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Unresolved
 - Presentation
 - Simulation: Main
 - Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
 - Network Modification of SIR AB
 - Main

The bottom left pane shows a "Problems" list with the following entries:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition
- Element doesn't belong to any state
- Hanging transition

Discrete Agent Dynamics: Transitions

- Many transition conditions are possible
 - Timeout: Spending some period of time in the state
 - Fixed rate: Leave state with some fixed change per unit time
 - This is similar to “first order interarrival time”, and is conceptually linked to the operation of first-order delays in stock & flow diagrams
 - Variable rate: If desired, we can change the rate over time – but Anylogic only “notices” changes when eq agent re-enters the state
 - Message received: We can transition when a message (any message or particular type of message) is received
 - Predicate: Only transition when condition becomes true
- These transitions can be conditionally “routed” via branches
 - Conditions can determine to what destination state a particular transition will travel

Adding Fixed Rate Transition

The screenshot displays the AnyLogic Advanced interface. The main workspace shows a statechart with two states: 'Susceptible' and 'Infected'. A transition arrow points from 'Susceptible' to 'Infected'. A red arrow points from the text 'When this really connects on both sides, circles should be green' to the transition arrow. The 'Infection - Transition' properties window is open, showing the transition name 'Infection', triggered by 'Rate', and a rate of '0.01'. A red arrow points from the text 'This implies mean time Susceptible = 100' to the '0.01' rate value. The left sidebar shows a project tree with 'MalariaV2' and 'TestModel2*' folders. The bottom left shows a 'Problems' window with two error messages: 'Element doesn't belong to any state' and 'Hanging transition'.

When this really connects on both sides, circles should be green

This implies mean time
Susceptible = 100

Tip: Beware Loose Connections

The screenshot displays the AnyLogic Advanced interface. The main workspace shows a state transition diagram with two states: 'Susceptible' and 'Infective'. A transition arrow points from 'Susceptible' to 'Infective', triggered by a rate of 0.1. A variable 'color' is associated with the transition. A loose connection is visible as a vertical line extending upwards from the transition area.

The left sidebar shows the project structure for 'MalariaV2', including 'Main', 'Parameters', 'Functions', 'Events', 'Embedded Objects', 'Analysis Data', 'Presentation', 'Mosquito', 'Person', 'Simulation: Main', 'TestModel2*', 'Spatial SIR with Waning Immunity', and 'Network Modification of SIR AB'.

The bottom-left pane shows the 'Problems' window with the following errors:

- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Hanging transition

The bottom-right pane shows the 'Properties' window for 'transition1 - Transition'. The 'General' tab is active, showing the following settings:

- Name: transition1
- Show Name:
- Ignore:
- Public:
- Show At Runtime:
- Triggered by: Rate
- Rate: 0.1
- Action: (empty)
- Guard: (empty)

The status bar at the bottom indicates 'Selection' and 'Cursor: X=64, Y=86'.

Corrected

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: MalariaV2

- Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Person
 - Simulation: Main
- Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
- Network Modification of SIR AB
 - Main
 - Person
 - Plain Variables
 - color
 - Statecharts
 - Functions

Console: transition1 - Transition

General

Description

Name: transition1 Show Name Ignore Public Show At R...

Triggered by: Rate

Rate: 0.1

Action:

Guard:

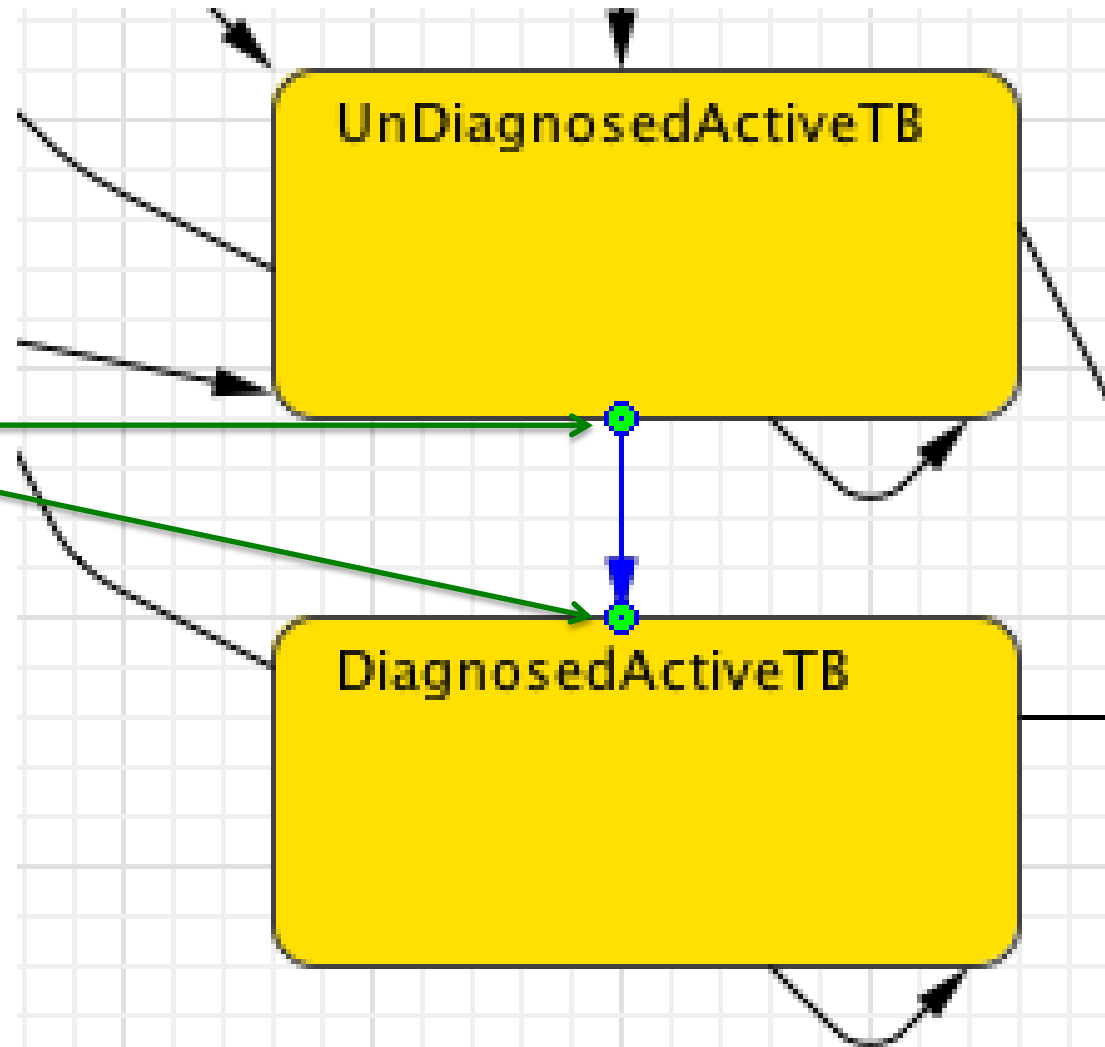
Problems

- Description
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state
- Element doesn't belong to any state

Selection

Tip: Confirming Transition Connectivity

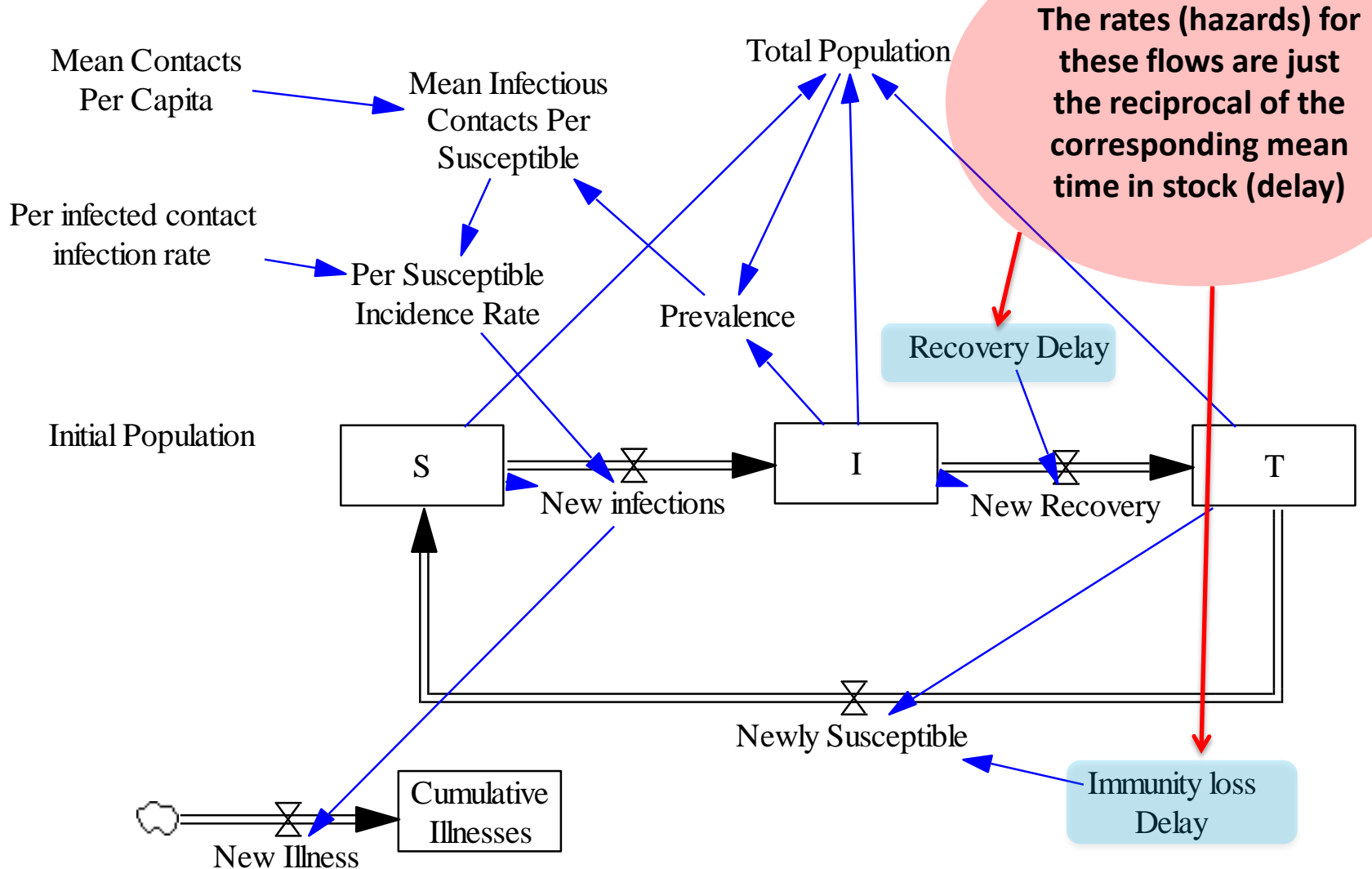
- Ensure that both sides of the transition show **green circles** when connected
 - Otherwise, may appear connected but will actually be disconnected!



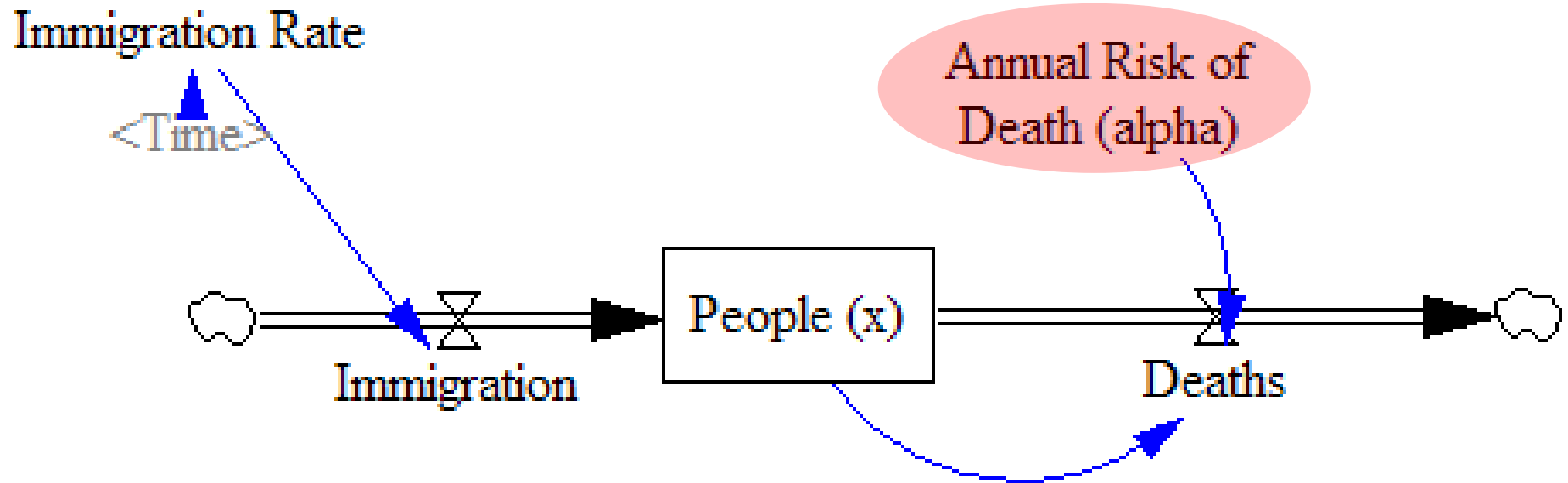
Rates & Flows

- Some may have seen fixed rates before – in the form of “transition rates” in Compartment models
- Within a Compartment/SD model, a flow out of a stock was commonly set by the multiplication of the
 - State variable (Stock)
 - Some rate of transition
- We use different names for these rates
 - “Transition rates”
 - “Likelihood of transition per *Unit Time*”
 - Transition (e.g. “infection”, “mortality”) “hazard”

First Order Delays in Action: Simple SIT Model



Example Fixed Transition Rate/Hazard

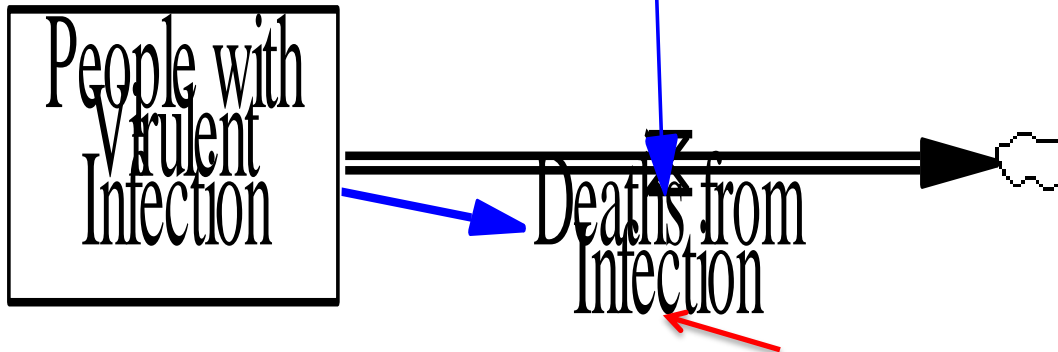


Example Fixed Transition Rate/Hazard

The transition rate is the *reciprocal* of this number i.e.

$$\frac{1}{\text{Mean time until Death}}$$

Mean time until Death



People with Virulent Infection/Mean time until Death

=

People with Virulent Infection*(1/Mean time until Death)

i.e. People with Virulent Infection*Rate

Fixed Rates: Transition “Hazards”

- With “fixed rates”, we are specifying rates of transitions
- Because we are dealing with the chance that each individual transitions, we don’t need to multiply by the number of people at risk
 - Here, there is just 1 person at risk!
- As in Compartment models, these rates can change over time, but the statechart needs to be “made aware” of these changes (see later)
 - Leave & go back into current state (circular transition)
 - Trigger “change” event in Agent

Adding Infection Clearance Transition

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a statechart diagram with two states: 'Susceptible' and 'Infected', both represented by yellow rounded rectangles. A transition arrow points from 'Susceptible' to 'Infected'. A red circle highlights the 'InfectionClearance' transition, which is currently inactive. The transition configuration panel at the bottom shows the following settings:

- Name: InfectionClearance
- Triggered by: Rate
- Rate: 0.1
- Action: (empty)
- Guard: (empty)

The left sidebar shows a project tree with folders for 'MalariaV2', 'TestModel2*', and 'Spatial SIR with Waning Immunity'. The right sidebar contains a 'Model' palette with various components like Parameter, Flow Aux, Stock Vari, Event, Dynamic, Plain Vari, Collectio, Function, Table Fun, Port, Connector, Entry Point, State, Transition, Initial Stat, Branch, History St, Final State, and Environm. The bottom status bar indicates 'Selection' and 'Cursor: X=100, Y=207'.

Run the Model!

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a statechart diagram with two states: "Susceptible" and "Infected", both represented by yellow rounded rectangles. A transition labeled "InfectionS" is shown with a black circle at its start and an arrow pointing to the "Susceptible" state. A variable "color" is shown with a yellow circle icon. The left sidebar contains a project tree for "MalariaV2", with "Main" selected. The bottom panel shows the "InfectionClearance - Transition" configuration, with "Name" set to "InfectionClearance", "Triggered by" set to "Rate", and "Rate" set to "0.1". The "Action" and "Guard" fields are empty. The bottom status bar indicates "Selection".

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project: MalariaV2

- Main
 - Parameters
 - Functions
 - CountInfectiveHum...
 - CountInfectiveMosc...
 - PersistSimulationDa...
 - SelectRandomPerso...
 - SetParameters
 - SetSimulationOutpu...
 - getHumanPopulatio...
 - getMosquitoPopulat...
 - Events
 - Embedded Objects
 - Analysis Data
 - Presentation
 - Mosquito
 - Person
 - Simulation: Main
- TestModel2*
 - Main
 - Person
 - Plain Variables
 - Statecharts
 - InfectionStatechart
 - Presentation
 - Simulation: Main
- Spatial SIR with Waning Immunity
 - Main
 - Person
 - Simulation: Main
- Network Modification of SIR AB
 - Main
 - Person

Model:

- Parameter
- Flow Aux ...
- Stock Vari...
- Event
- Dynamic ...
- Plain Vari...
- Collectio...
- Function
- Table Fun...
- Port
- Connector
- Entry Point
- State
- Transition
- Initial Stat...
- Branch
- History St...
- Final State
- Environm...

Console: Properties

InfectionClearance - Transition

General

Description

Name: InfectionClearance Show Name Ignore Public Show At Ru...

Triggered by: Rate

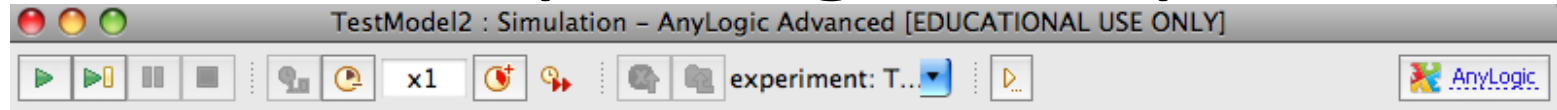
Rate: 0.1

Action:

Guard:

Selection

Completing Set-Up



TestModel2

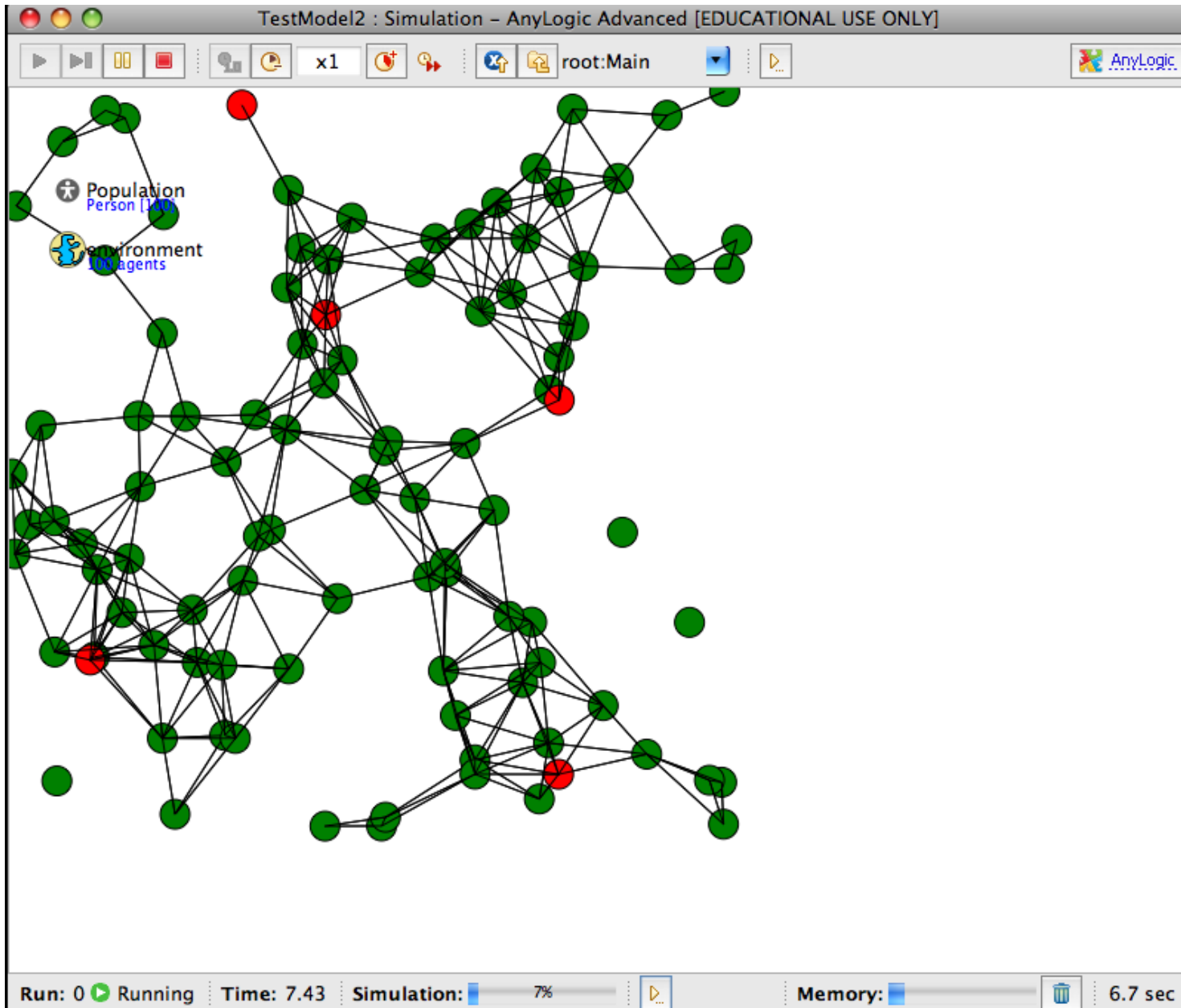
Experiment setup page

Run the model and switch to Main view

Press this button to start model execution

Run: 0 Idle Time: 0.00 Simulation: Stop time not set Memory: 7M of 63M 0.0 sec

Model Presentation



Transition Type: Fixed Residence Time (Timeout)

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a statechart for TB progression. The states are represented by yellow rounded rectangles: "TBSusceptible", "LTBI", "UnDiagnosedActiveTB", and "DiagnosedActiveTB". Transitions are shown as arrows between these states, with decision diamonds for "WhetherInfected" and "WhetherPrimaryProgression". A "Diagnosis" transition is shown as a blue arrow from "UnDiagnosedActiveTB" to "DiagnosedActiveTB".

The "Diagnosis - Transition" properties window is open at the bottom, showing the following configuration:

- Name: Diagnosis
- Show Name:
- Ignore:
- Public:
- Show:
- Triggered by: Timeout
- Timeout: `get_Main().DaysUntilDiagnosis/DaysPerTimeUnit`
- Action: `traceln("Diagnosis performed");`
- Guard: (empty)

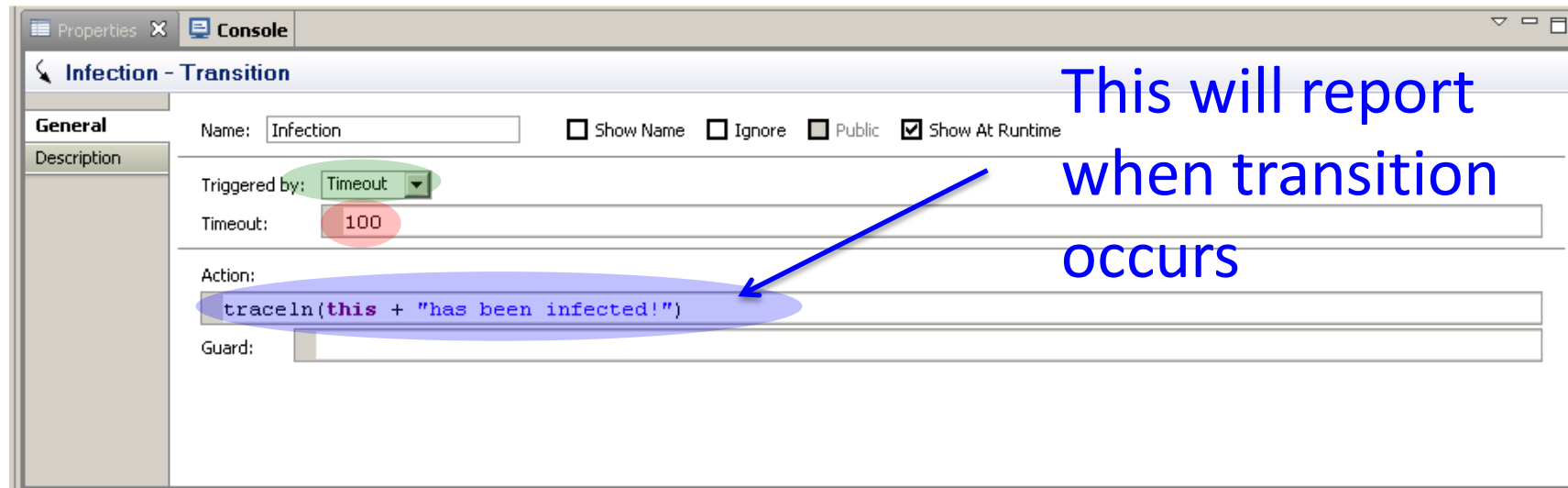
The left sidebar shows a project tree with folders for "Main" and "Person". The "Main" folder contains parameters, functions, environments, and statecharts. The "Person" folder contains parameters, plain variables, dynamic variables, and statecharts. The "Problems" window at the bottom left shows several error messages related to "Engine.log cannot be resolved" and "Cannot make a static reference to the non-static method getCurrentState()".

Example of Processes Associated with Fixed Timeouts

- Aging
- Tightly defined time constants associated with natural history
 - While these may be described as associated with a broad distribution (e.g. with a 1st or 2nd order delay), much of that variability may be due to heterogeneity
 - *For a given person, these may be quite specific in duration ⇒ Can capture through a timeout*

What Happens if this Depends on a Timeout?

- Set the “Infection” transition to Trigger based on a “Timeout”



- Make the “Timeout” 100

Now run the model, and observe the difference



Hands on Model Use Ahead



Load model: TBv1.alp

Transition Type: Variable Rate

The screenshot displays the AnyLogic Advanced software interface for modeling a Tuberculosis (TB) progression statechart. The main workspace shows a statechart with the following states and transitions:

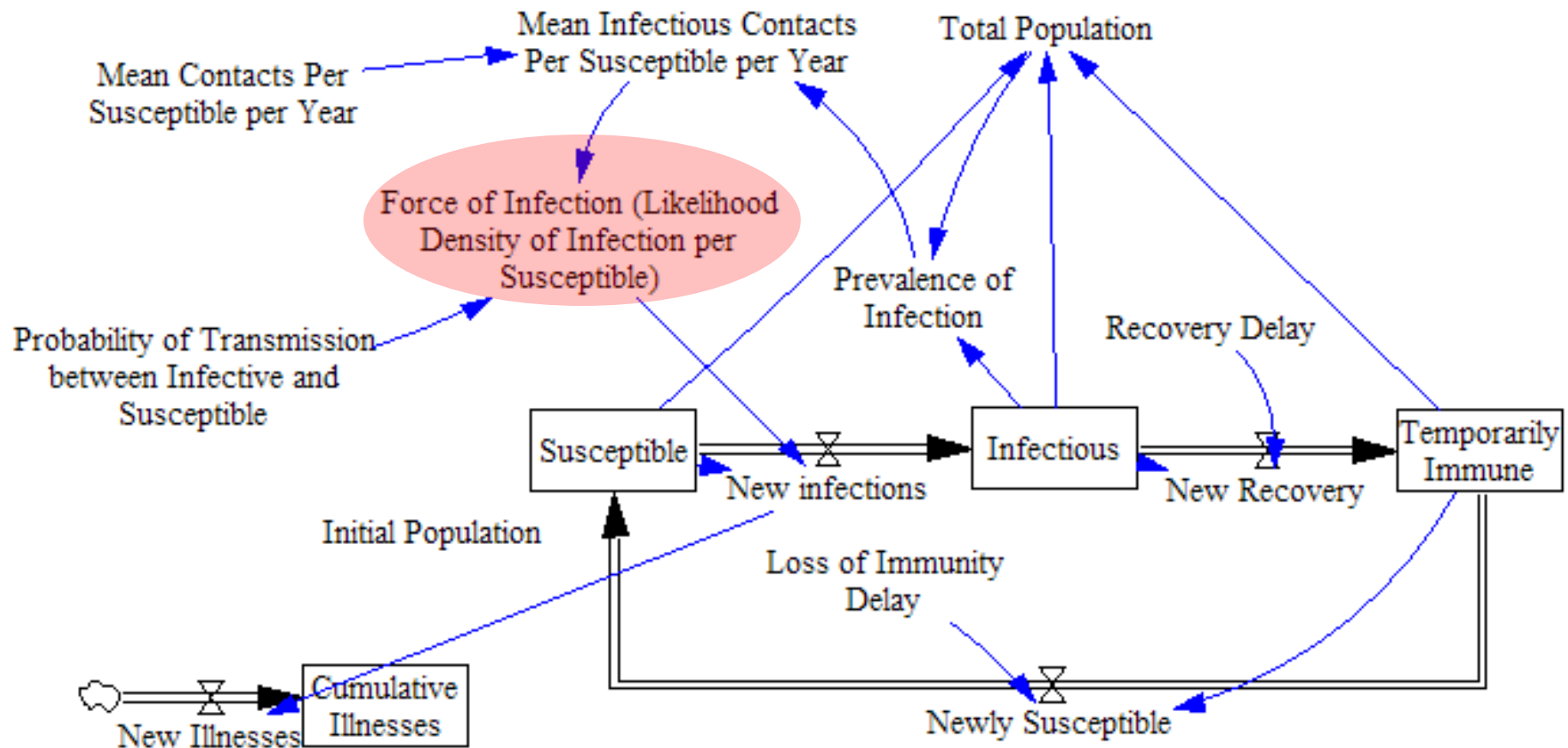
- States:** TBSusceptible, LTBI, UnDiagnosedActiveTB, DiagnosedActiveTB.
- Transitions:** WhetherInfected (diamond), WhetherPrimaryProgression (diamond), Reactivation (variable rate transition), and Death (circle).
- Initial State:** TBSusceptible.
- Final State:** Death.

The **Reactivation - Transition** properties are shown in the bottom panel:

- Name:** Reactivation
- Triggered by:** Rate
- Rate:** ReactivationRateForCKDStage()
- Action:** `traceln("Reactivated");`
- Guard:** (empty)

The left sidebar shows the project structure for **TBv1***, including parameters, functions, environments, and statecharts. The bottom-left pane displays a list of errors, such as "Engine.log cannot be resolved" and "Type statechart cannot be resolved".

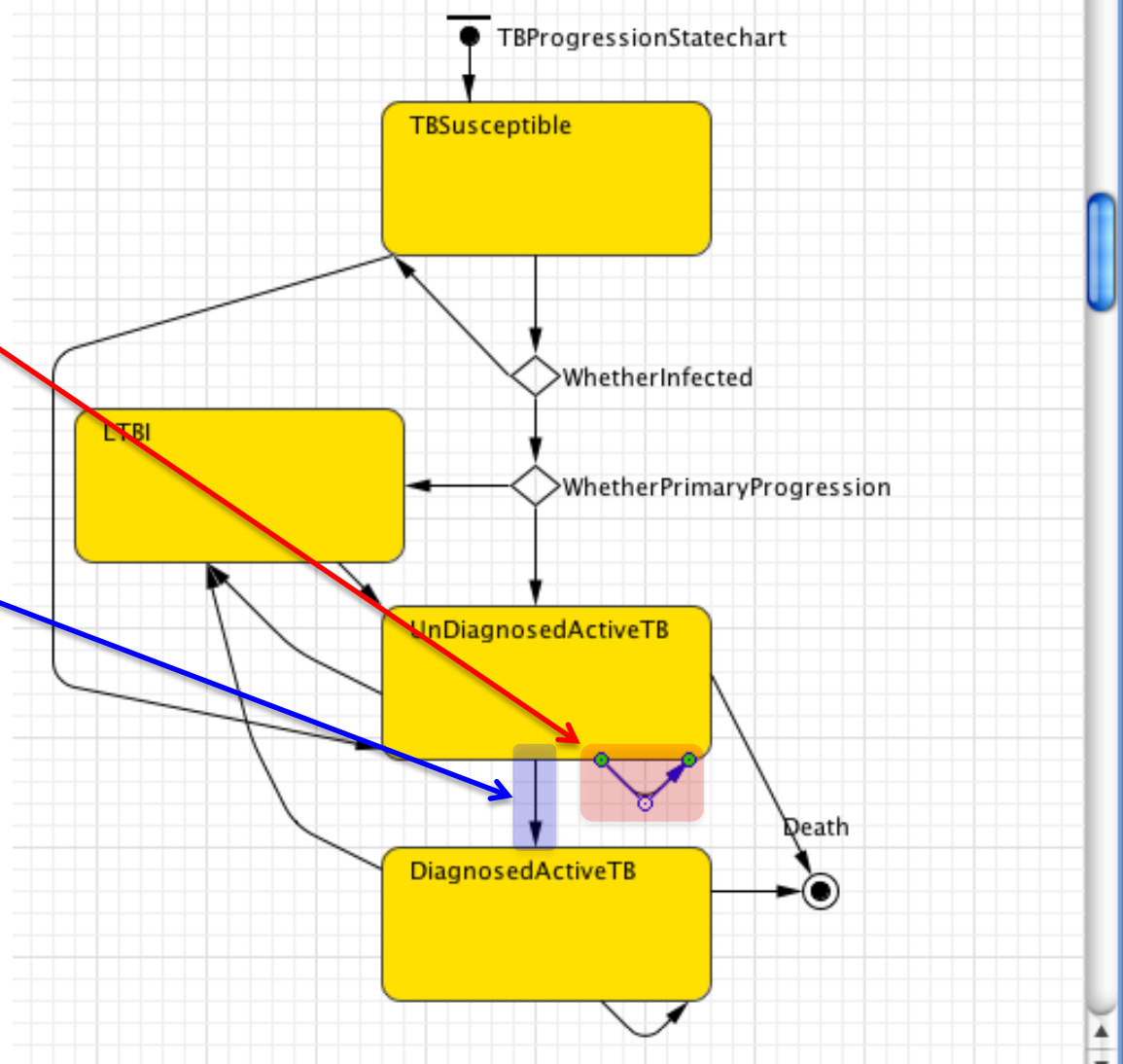
Example Transition Rate/Hazard



Special Elements: Self-Transition

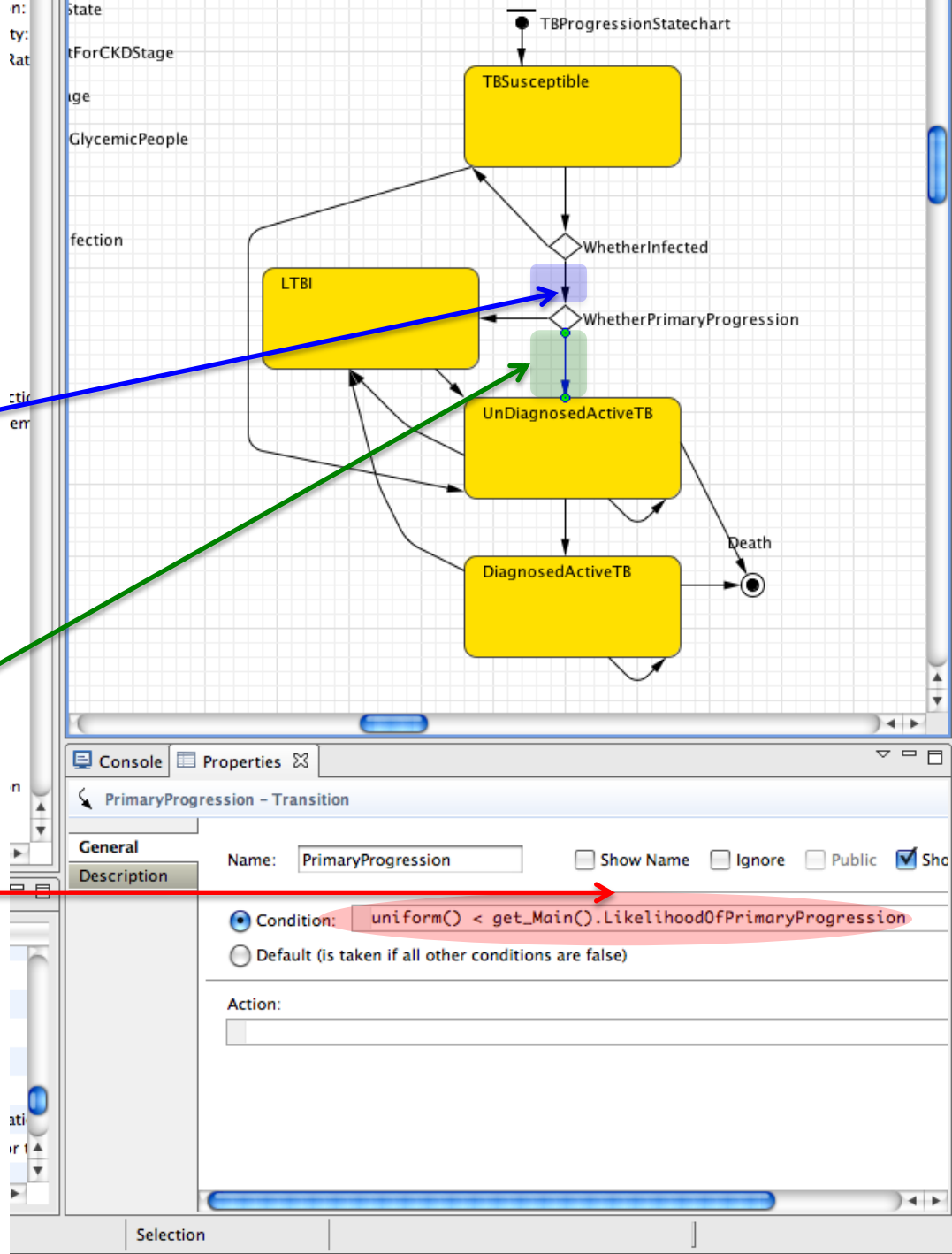
(Use if Wish To Have State Register Changing Out-transition rates)

The **self-transition** will “make the state realize” that the rate associated with any out transition (e.g. **this one**) has changed

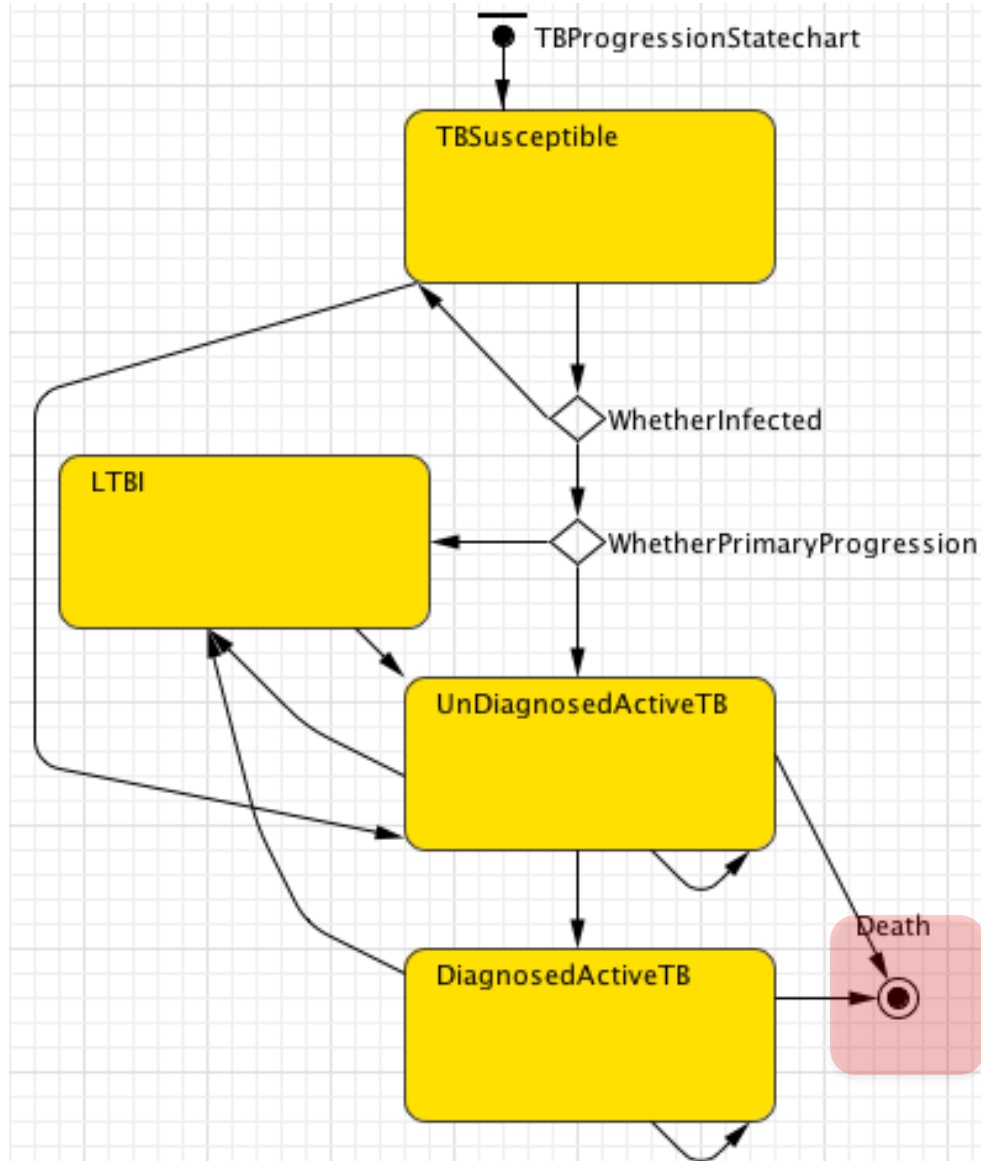


Example Conditional Transition

The **incoming** transition into “WhetherPrimaryProgression” will be routed to this **outgoing** transition if **this condition** is true



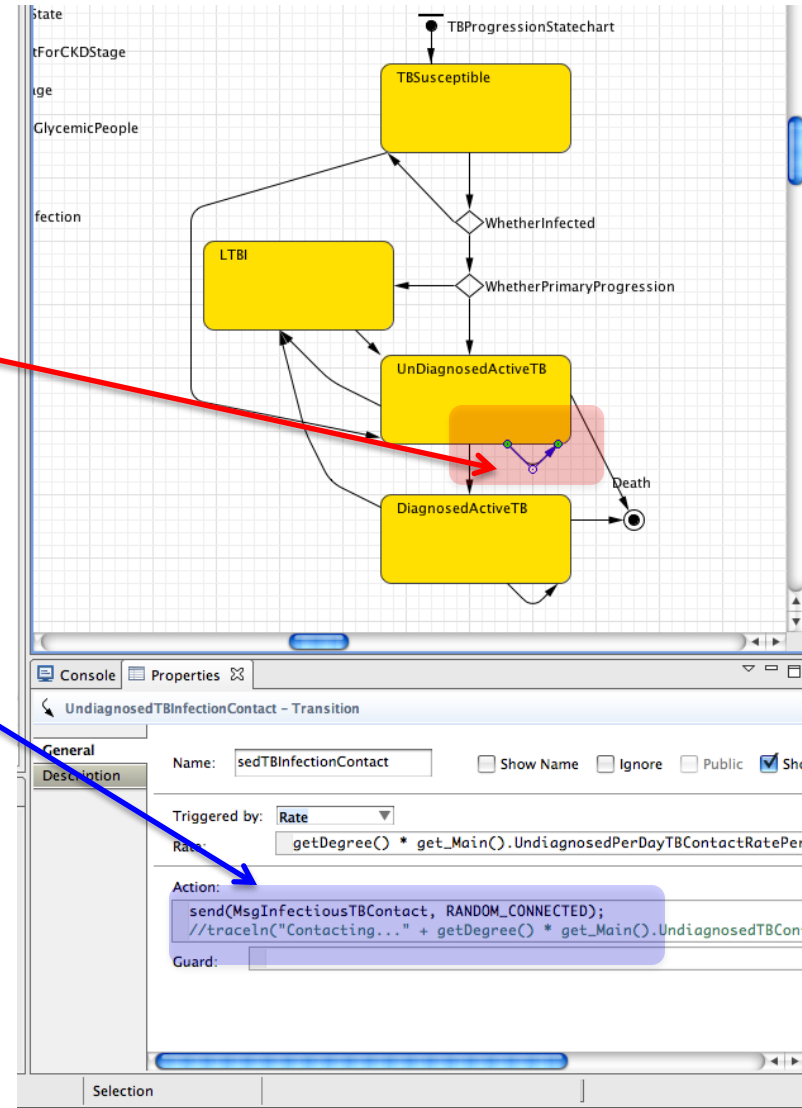
Special Elements: Exit Point



Special Elements: Self-Transition

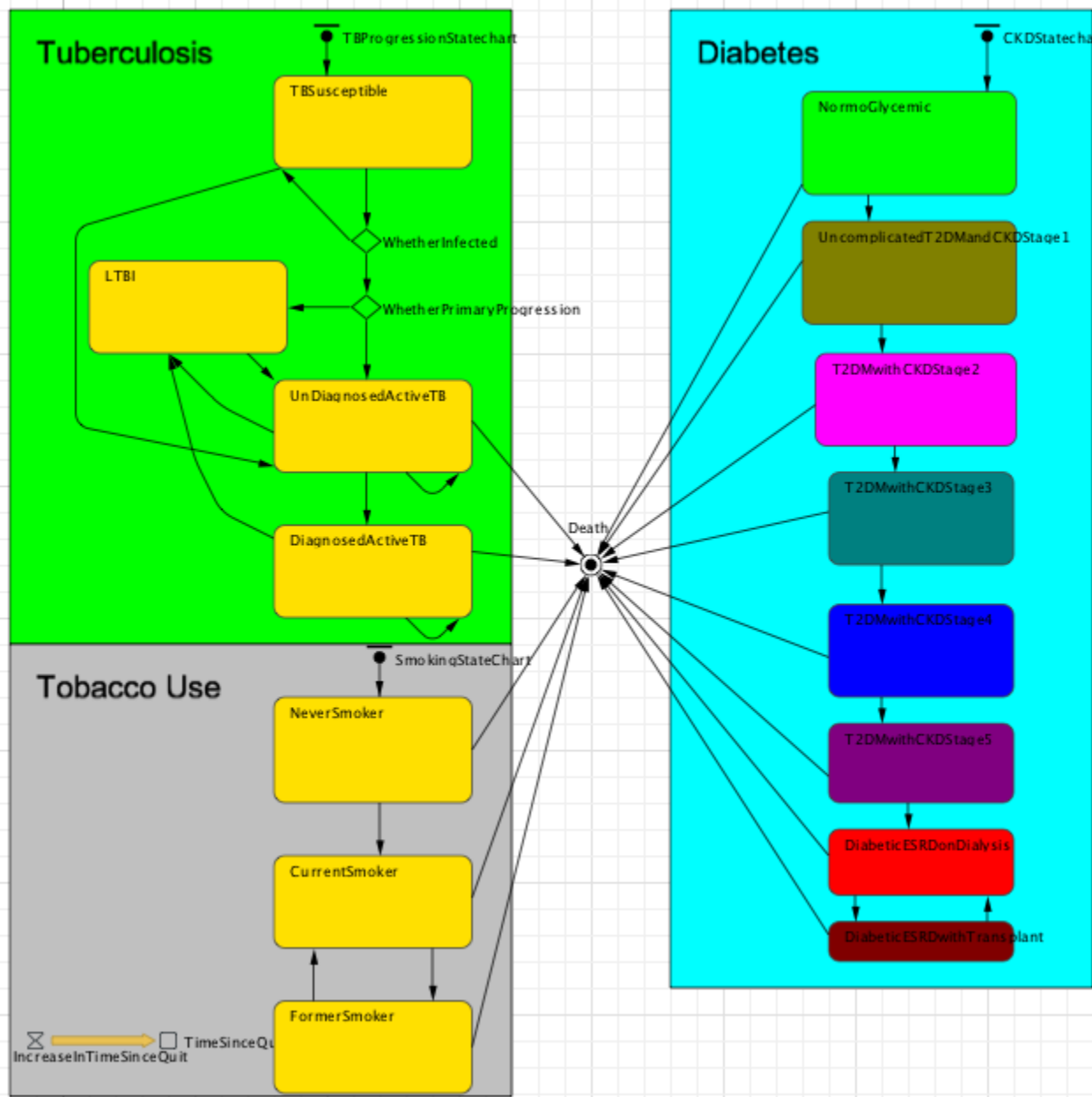
(Use if Wish To Trigger an Action w/o Leaving State)

The **self-transition** will invoke **this action** when it occurs



Parallel Statecharts

- By default, each statechart evolves independently.
- If coupling is desired, can make transitions/actions dependent on state of other statecharts

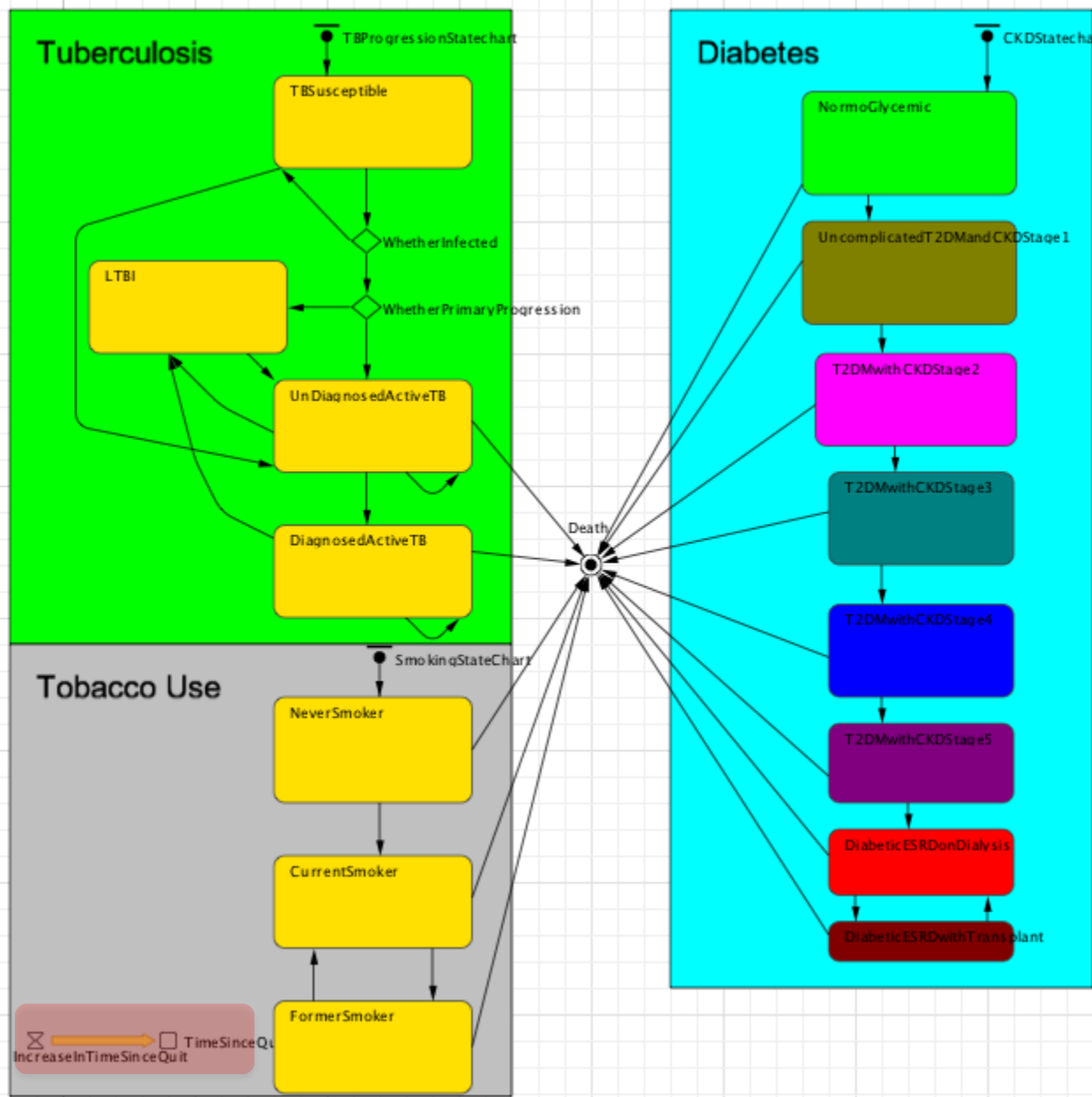


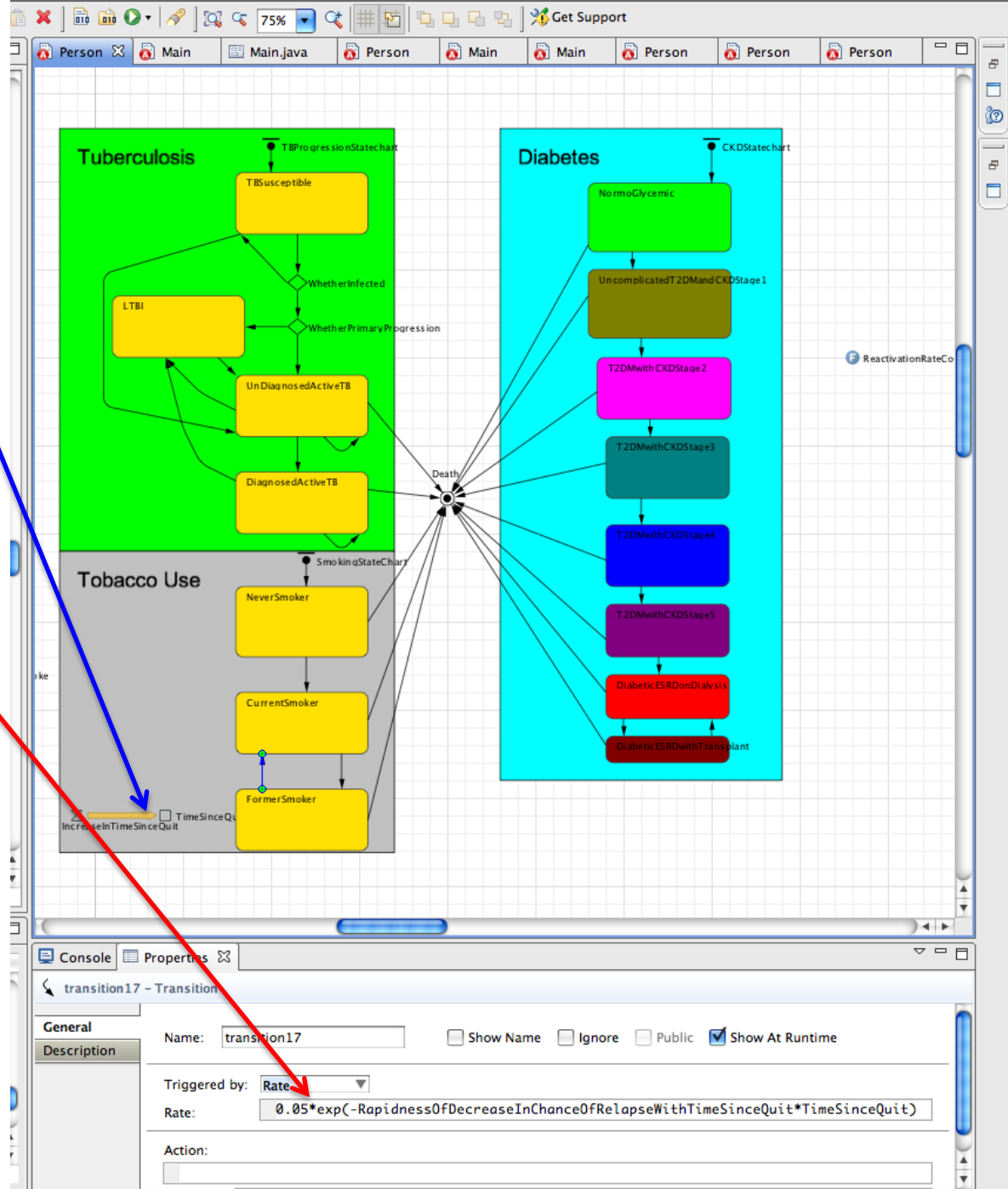
Comparison with Aggregate Stock & Flows

- As for aggregate stocks & flow, individuals' states are discrete
- Unlike aggregate stocks & flows
 - One state within a given statechart is active at a time
 - For parallel flows (e.g. comorbidities), there is no need for considering all combinations of the possible states
 - We can keep track of how long an individual is in a given state & adjust the transition rate accordingly

Parallel Transitions

- Example recording the residence time in a state (via a stock with unit inflow -- i.e. just accumulates the time present in that state)





- The **residence time** in the state determines the **transition rate** out of that state.
- Transition rates depending on residence time are generally not possible with aggregate models



Hands on Model Use Ahead



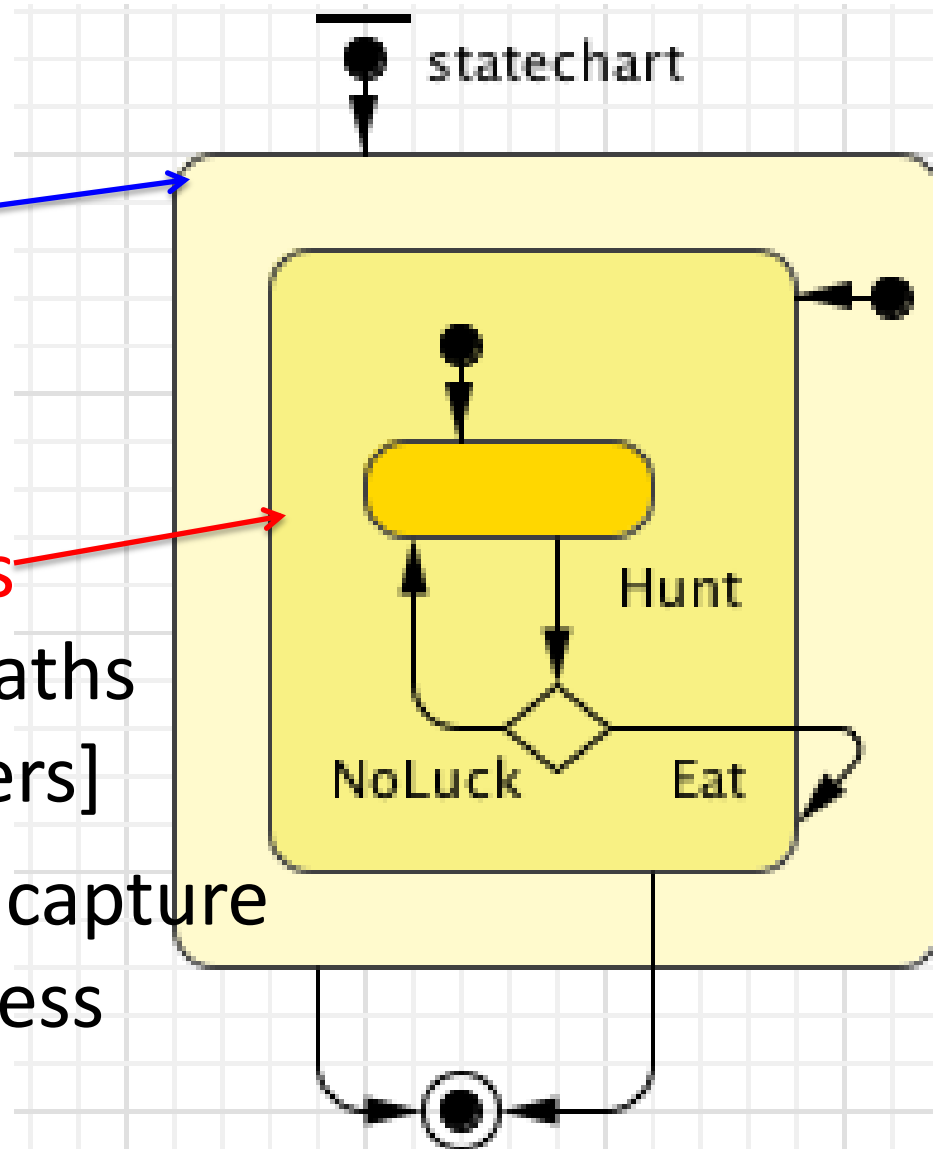
Load Sample Model:

Predator-Prey Agent Based

(Via “Sample Models” under “Help” Menu)

Advanced Element: Hierarchical States

- The **outermost state** captures time since born (for natural deaths)
- The **middle-state captures** time since last ate (for deaths by hunger). [Eating reenters]
- The inner state transition capture hunting frequency & success



Natural Death Transition

The image displays a software development environment, likely Eclipse, with a statechart editor. The main window shows a statechart diagram for a Lynx agent. The statechart is contained within a yellow box labeled 'statechart'. It features a start node leading to a state, which then transitions to a state labeled 'NoLuck'. From 'NoLuck', there is a transition labeled 'Hunt' leading to a state labeled 'Eat'. From 'Eat', there is a transition labeled 'Eat' leading to a state labeled 'cell'. The statechart is triggered by an event 'HaveBabies' and a parameter 'Width'. A variable 'CellWidth' is also shown. The statechart is associated with a 'cell' variable.

The left sidebar shows a project tree with the following structure:

- Project
- ↳ Contact
- ↳ Presentation
 - ↳ pixel
- ↳ Simulation: Main
 - ↳ Presentation
- ↳ Predator Prey Agent Based
 - ↳ Hare
 - ↳ Parameters
 - ↳ Plain Variables
 - ↳ Statecharts
 - ↳ statechart
 - ↳ statechart
 - ↳ Alive
 - ↳ DiedBecauseO
 - ↳ Dead
 - ↳ WasEatenByAL
 - ↳ Lynx
 - ↳ Parameters
 - ↳ Plain Variables
 - ↳ Statecharts

The bottom right pane shows the properties for a transition named 'tran3':

General	Description
Name:	tran3
Triggered by:	Timeout
Timeout:	get_Main().LynxLifeExpectancy
Action:	
Guard:	

The bottom left pane shows a list of errors:

Description
Engine.log cannot be resolved
Engine.log cannot be resolved
Engine.log cannot be resolved
Engine.log cannot be resolved

Death By Hunger

(Note that Depends on Time in State – i.e. time Since last ate)

The image shows a UML modeling tool interface with the following components:

- Project Explorer (Left):**
 - Simulation: Main
 - Predator Prey Agent Based
 - Hare
 - Parameters
 - Plain Variables
 - Statecharts
 - statechart
 - statechart
 - Alive
 - DiedBecauseO
 - Dead
 - WasEatenByAL
 - Events
 - HaveBabies
 - Presentation
 - Lynx
 - Parameters
 - Plain Variables
 - Statecharts

- Diagram Area (Center):**
- Statechart diagram for 'statechart'.
- States: 'Alive' (yellow circle), 'Dead' (black circle).
- Transitions: 'NoLuck' (from Alive to Dead), 'Eat' (from Dead to Alive), 'Hunt' (from Alive to Alive).
- Association: 'cell' (orange circle) is associated with the statechart.
- External elements: 'Width', 'CellWidth', 'HaveBabies' (lightning bolt), 'statechart' (input).
- Properties Panel (Bottom Right):**
- Selected transition: 'tran4 - Transition'.
- General: Name: tran4.
- Description: Triggered by: Timeout; Timeout: get_Main().LynxHungerDeathThreshold.
- Action: (Empty field).
- Guard: (Empty field).
- Problems Panel (Bottom Left):**
- Multiple error messages: 'Engine.log cannot be resolved'.

Eating Transition Leaves & Reenters Middle State

The screenshot displays a statechart editor interface. On the left is a project tree showing a statechart for a 'Hare' agent. The main workspace shows a statechart with a state containing a 'NoLuck' state and an 'Eat' transition. The 'Eat' transition is highlighted with a red box. Below the workspace, the Properties window for the 'Eat' transition is visible, showing its name and a condition.

Properties Window: Eat - Transition

General

Name: Show Name Ignore Public Show

Description

Condition: `randomTrue(min(1, get_Main().HaresInCell[cell].size() / (`

Default (is taken if all other conditions are false)

Action:

```
Main m = get_Main();  
Hare prey = m.HaresInCell[cell].get( uniform_discr( 0, m.HaresInCell[ce  
prey.statechart.receiveMessage( new Object() );
```

Tips on Statechart Code

- Each State & Transition has an integer index
 - This by accessed via a (static) constant holding the name of state within the statechart class (*statechart.StateName*)
- To determine length of time spent in state
 - *Statename.getLocalTime(StateIndex)*
- To determine current state
 - *statechart.getActiveSimpleState()*
- To find out if a state (either simple or composite) is currently active
 - *statechart.isStateActive(StateIndex)*