

# Representing Interventions

CMPT 858

Nathaniel Osgood

4-7-2011

# Representing Interventions in AnyLogic & Vensim

- Interventions disturb the baseline operation of the system
- Interventions can be represented by several types of changes, namely modifications to:
  - Parameter/initial state values
  - Model structure
  - Incentives represented in model
  - System state at one or more particular points in time

# Running Interventions

- Typical: Run Baseline and alternatives each in series
  - Compare results (as if sensitivity analysis)
- Radical but effective (e.g. for cost-effectiveness arguments)
  - Vensim: Subscripting Vensim Model by intervention (Baseline/Intervention A/Intervention B), and having run in parallel
  - AnyLogic: Run several populations in parallel (each associated with a different intervention)

# Model Granularity can Limit Options in Representing Interventions

- Model specificity provides limits our ability to investigate targeted interventions
- Model granularity may force us to represent more detail with respect to an intervention



# Model Granularity

## & Intervention specificity

- All other things being equal, the more detailed the model, the greater detail with which we can – and sometimes must! – specify interventions
- Examples
  - A model stratified by age&sex permits vaccinations to be rolled out at different times according to these factors
  - A model incorporating network structure allows us to target our interventions at network “hubs”
  - A model in which contacts emerge from agents moving between locations would allow us to examine how changing those locations would affect contact patterns
  - Capturing history supports history-specific interventions

# Fine Grained Models

## Oblige Specifying Added Intervention Details

- More detail in a model generally requires making more specific statements about intervention effects
- Contrast changes to mixing assumptions
  - Unstratified aggregate model: Changing  $c$
  - Stratified aggregate model: Changing mixing matrix (abstracting over exactly how this is accomplished)
  - Individual-based model with Network: Change certain areas of network (e.g. add/delete/modify connections)
  - Individual-based model where contacts emerge from move: Change something about specific factors driving mobility patterns

# Common Phrasing of Interventions

## What would be Impact of....

- “Reducing uptake rate by 10%”?
- “Increasing cessation rate by 10%”
- “Lowering mortality rate by 2%”
- “Reducing mixing levels by 7%”
- “Increasing emergency room staff by 20%”
- “Reducing the rate of progression of diabetes by 10%”

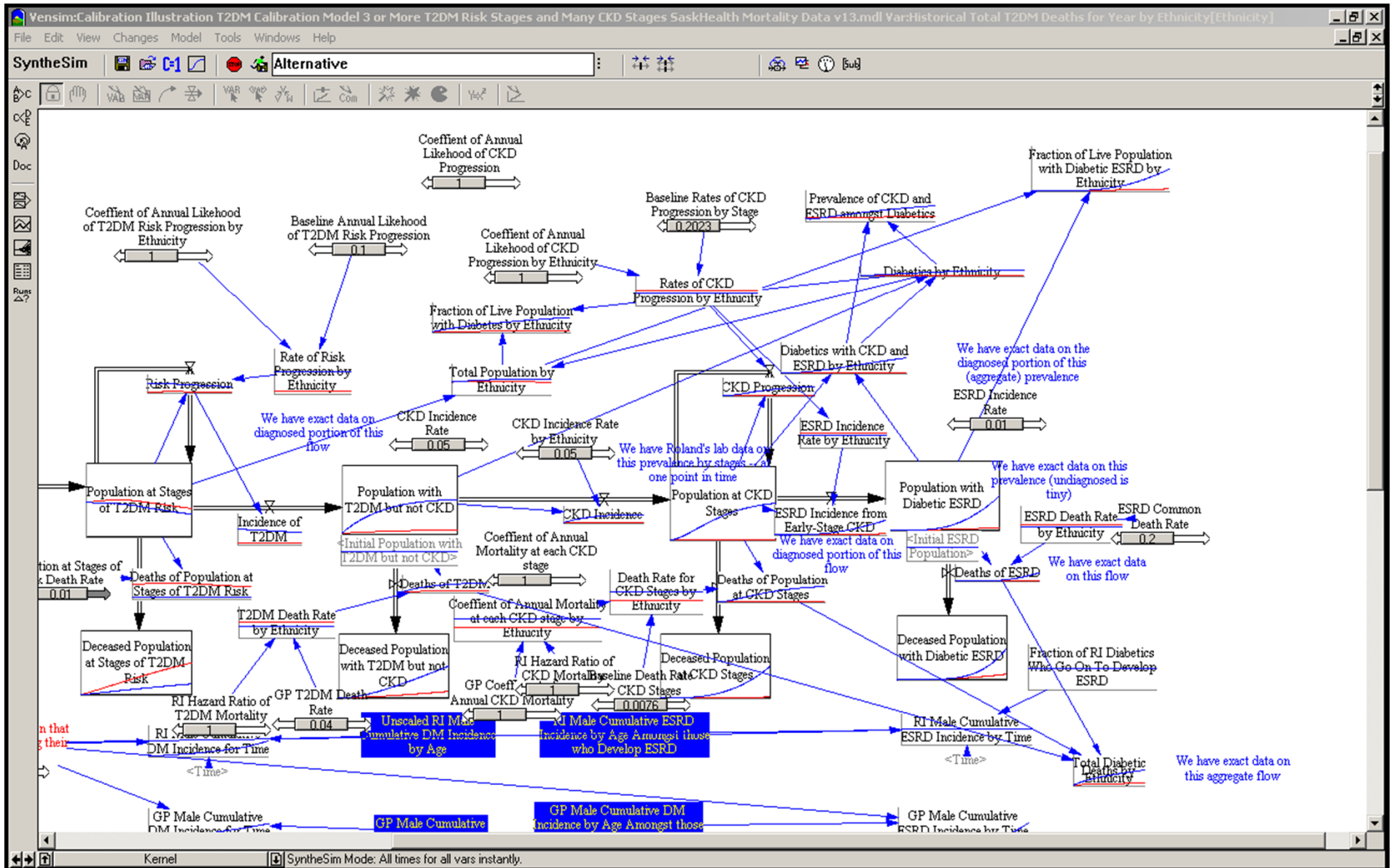
# Changing Parameter Values

- Frequently we can approximate an intervention's impact by changing behaviors already represented in the model
  - This is abstracting over the issue of the exact nature of how this is caused
- This might affect parameters or initial values
- Often several parameters may need to be changed together, e.g.
  - Higher smoking cessation rate, lower smoking relapse rate
  - Lower value of  $c$  & lower value of  $\beta$
- Be sure to restore parameters to their baseline values after experiments!

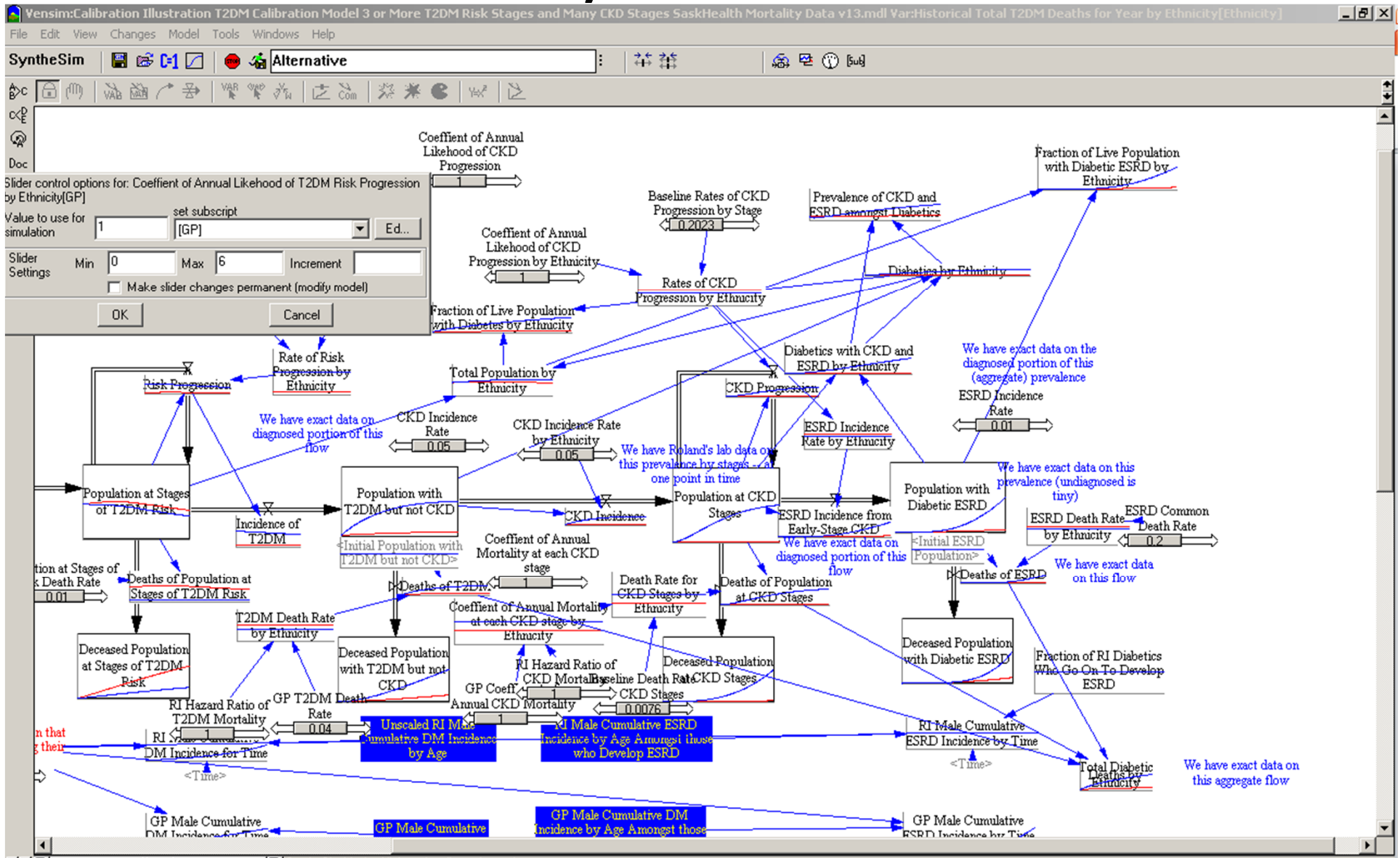
# Changing Parameters in Vensim

- In Synthesim
  - No worries that saving away the model will disturb baseline functioning!
  - Easy setting of values
    - (left click on variable to set exact value)
- Via “Gaming” variables (can adjust over time!)
- Via “changes” files (to remember *exact* changes across multiple parameters)
  - These can let you systematically save away parameter sets, each associated with a particular intervention scenario
- By modifying value of parameter within the model specification (to constant, formula, or time series)
  - Remember to restore (Indicate change with color eg red)

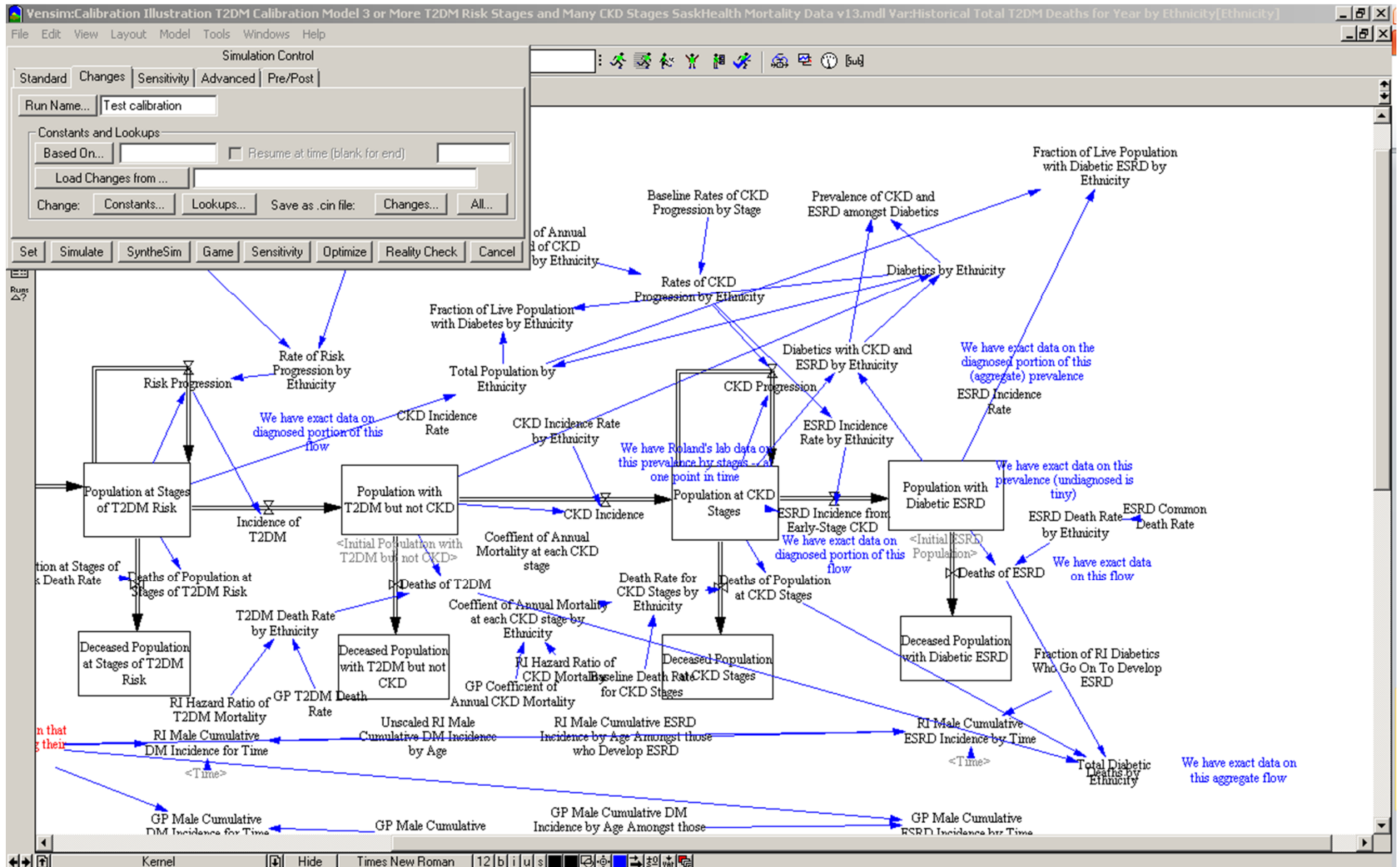
# Synthesim



# Setting the Value of a Parameter in Synthesim

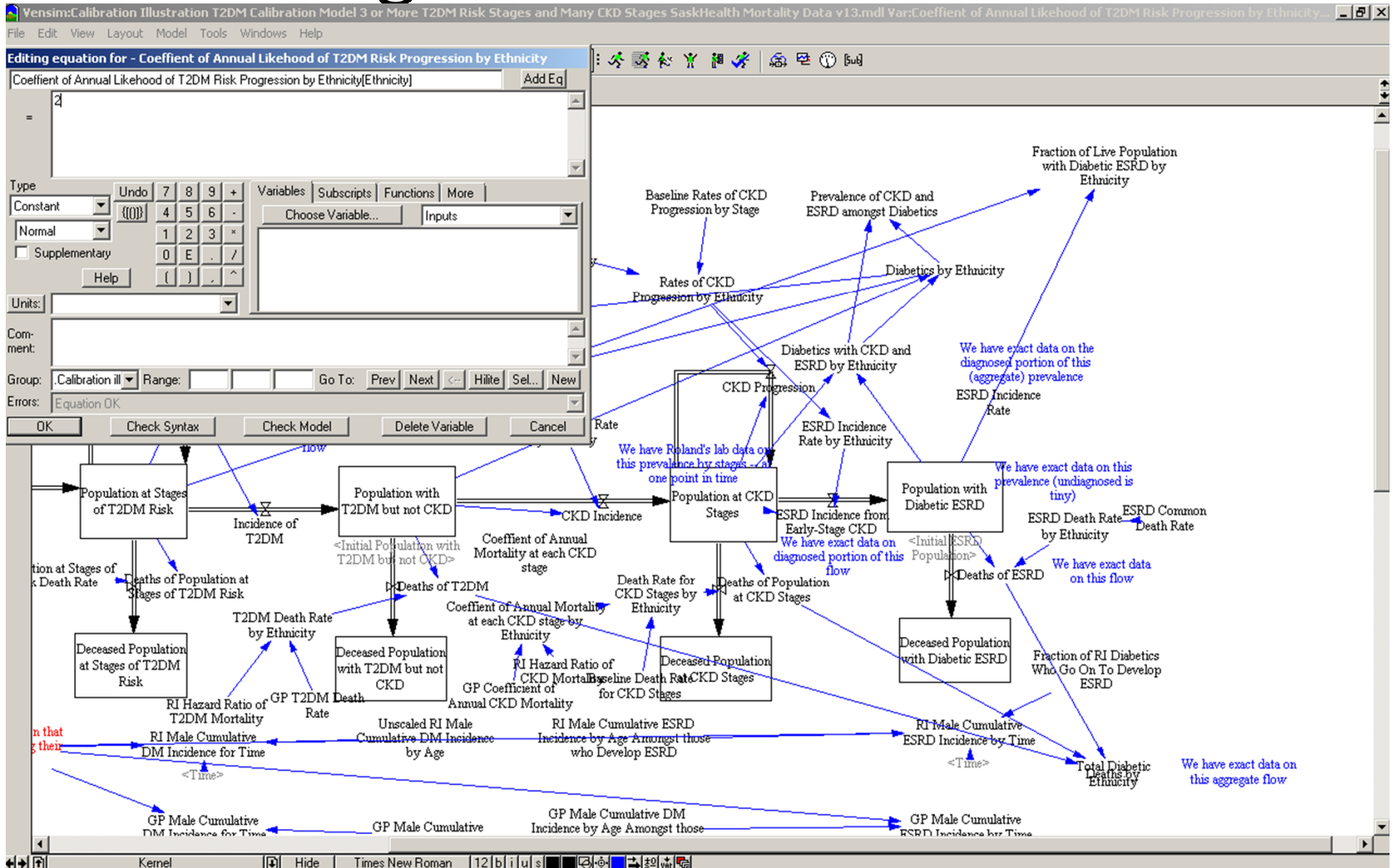


# Loading Changes File

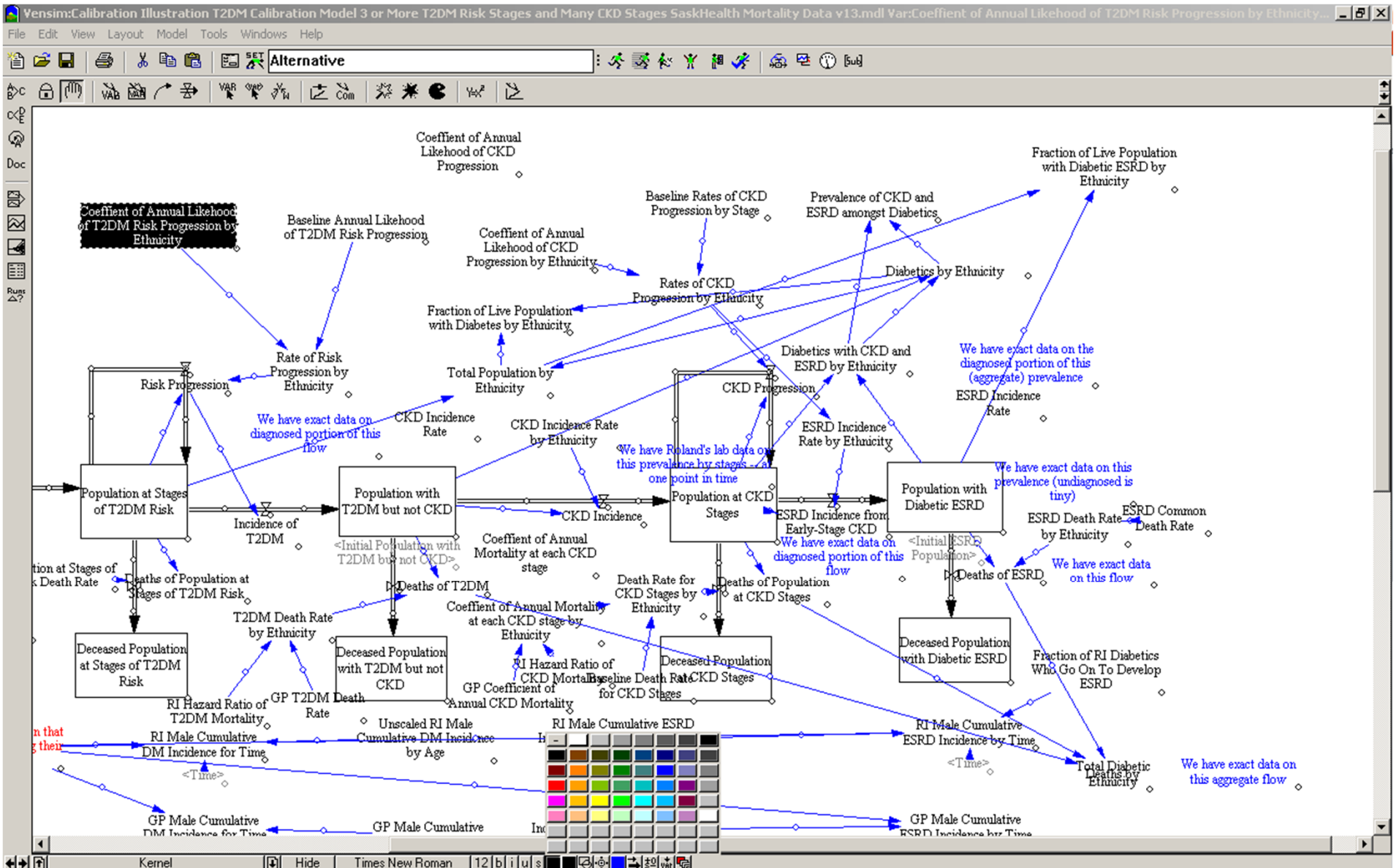




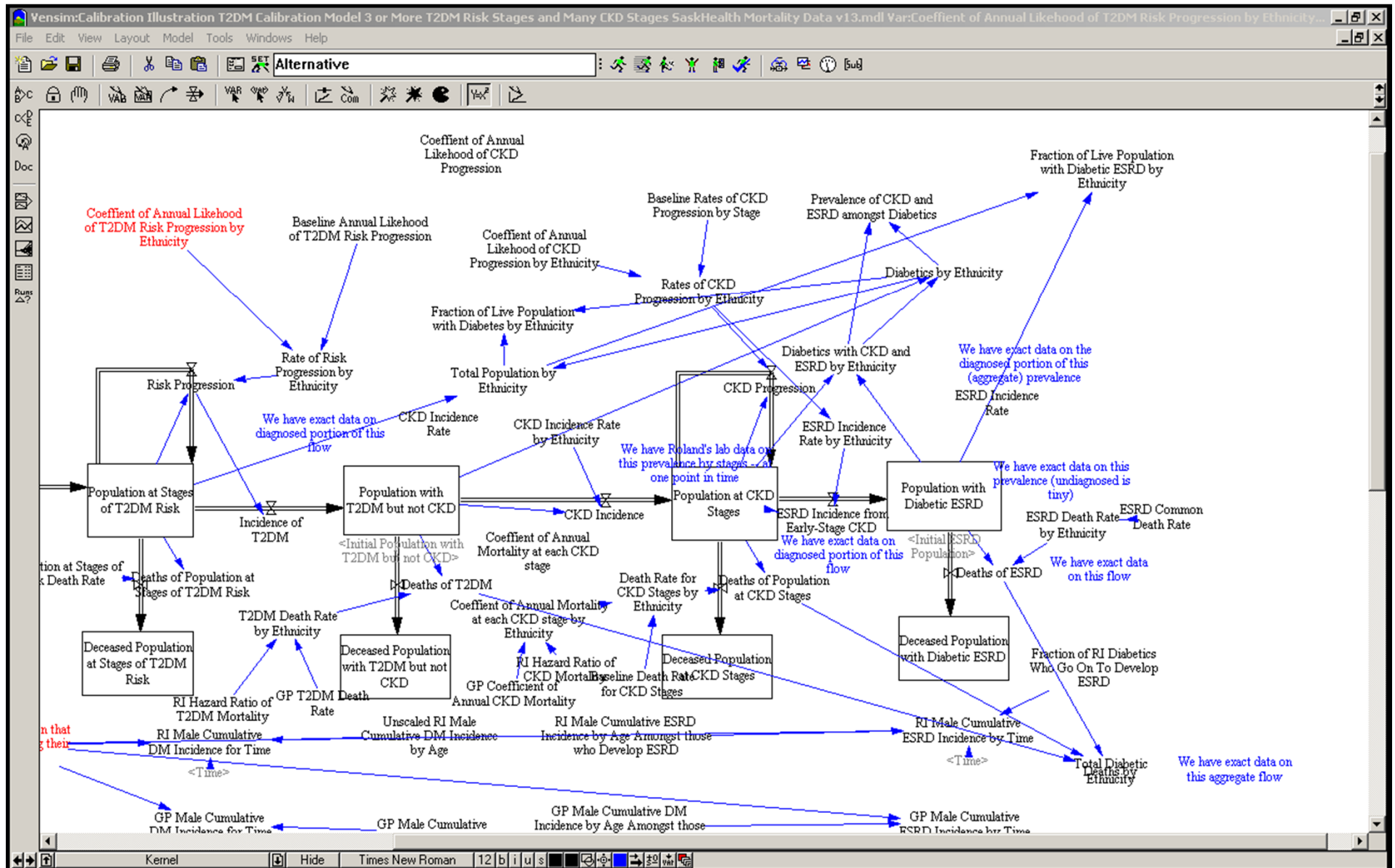
# Altering Variable Value in Model



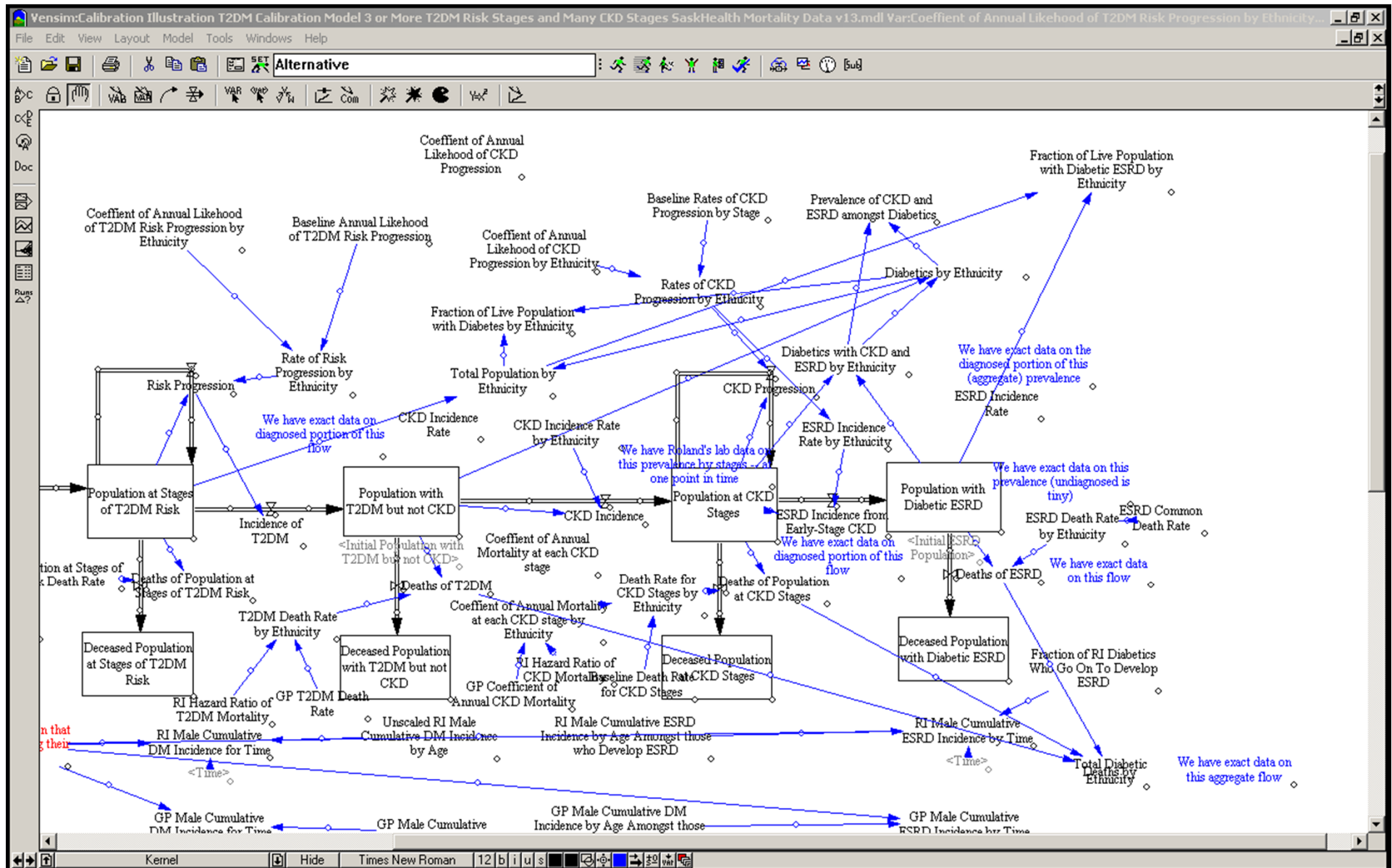
# Setting the Color of Parameter to Remind Us that it is Changed



# Following Color Change

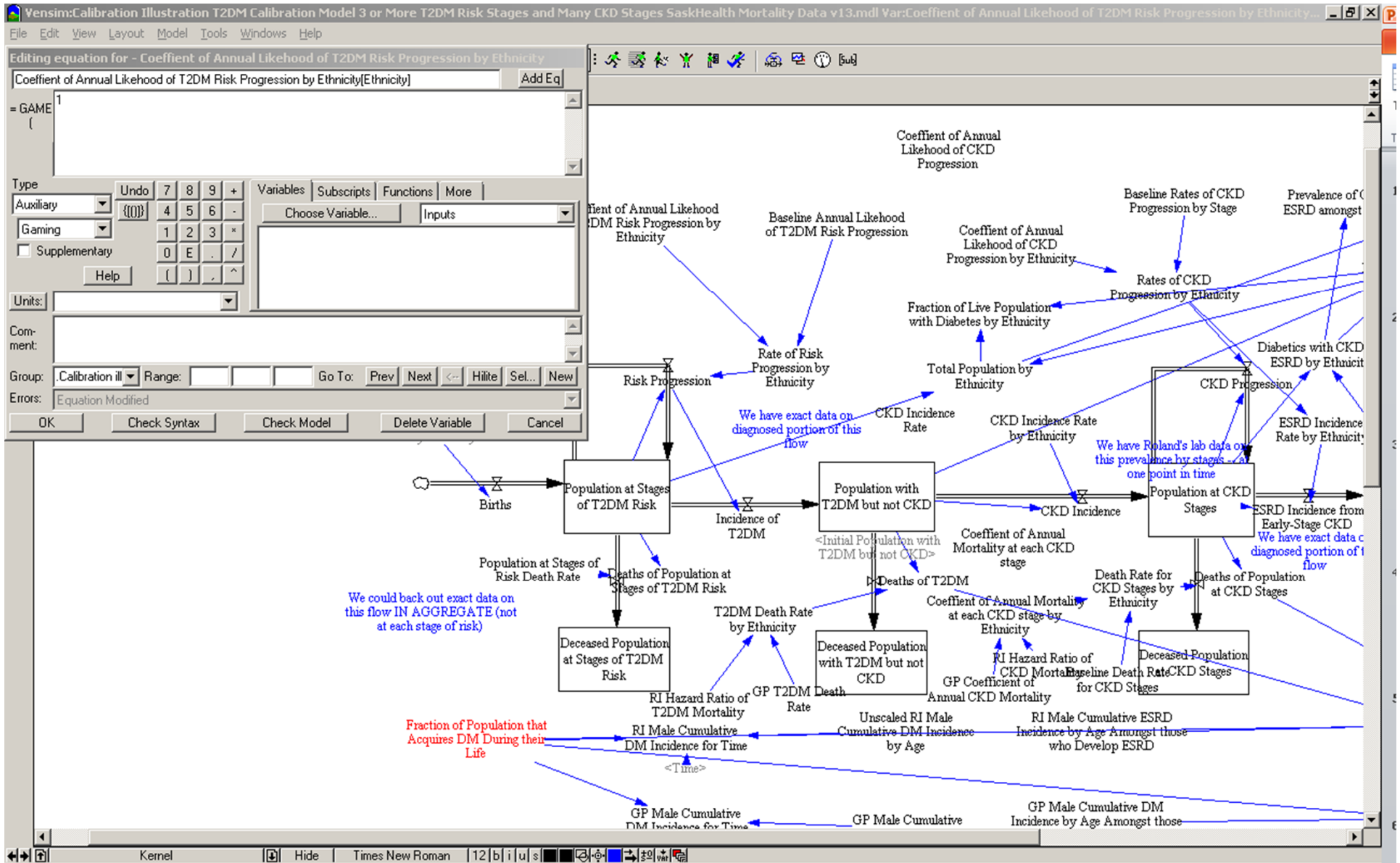


# Restore Color once Restore Value

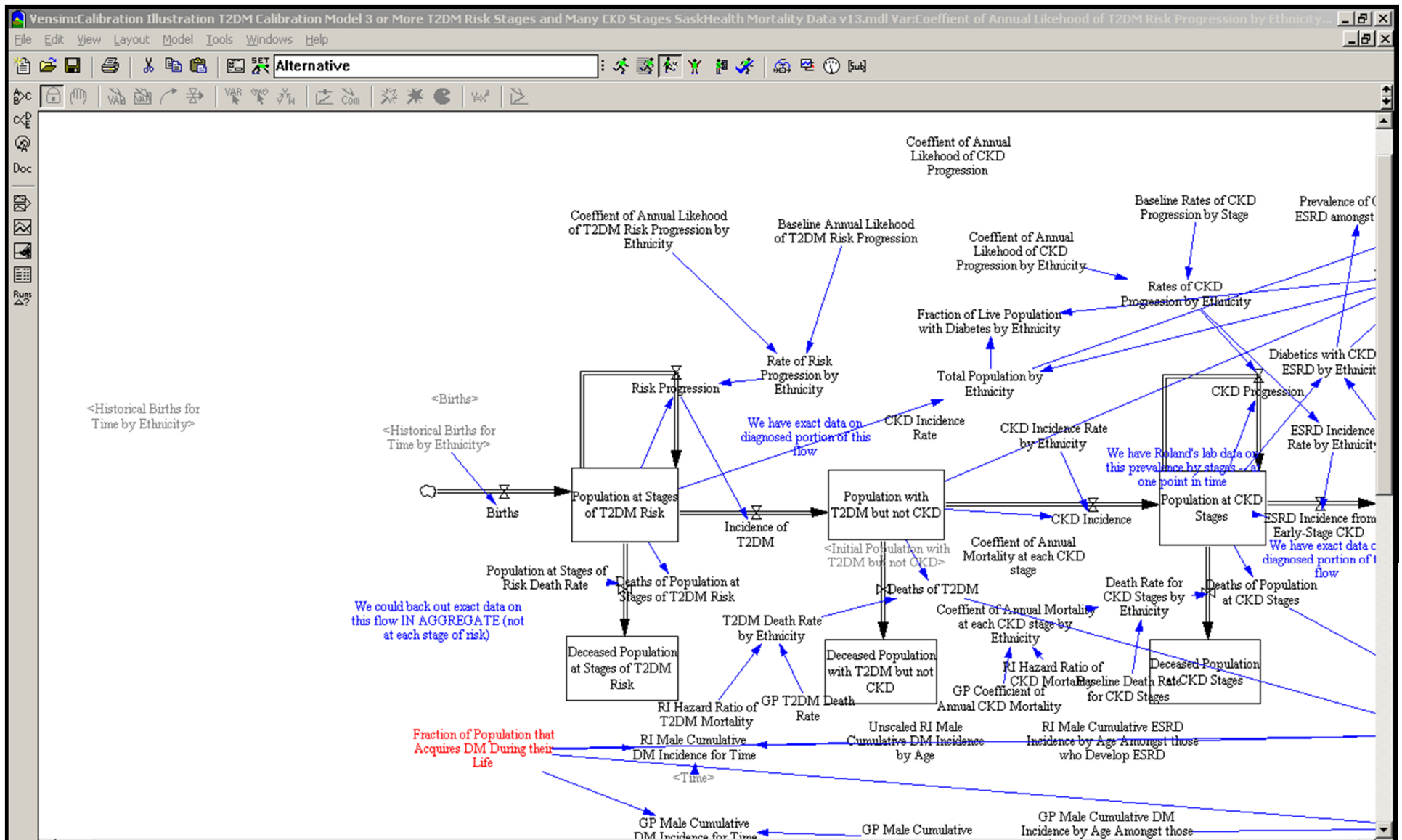




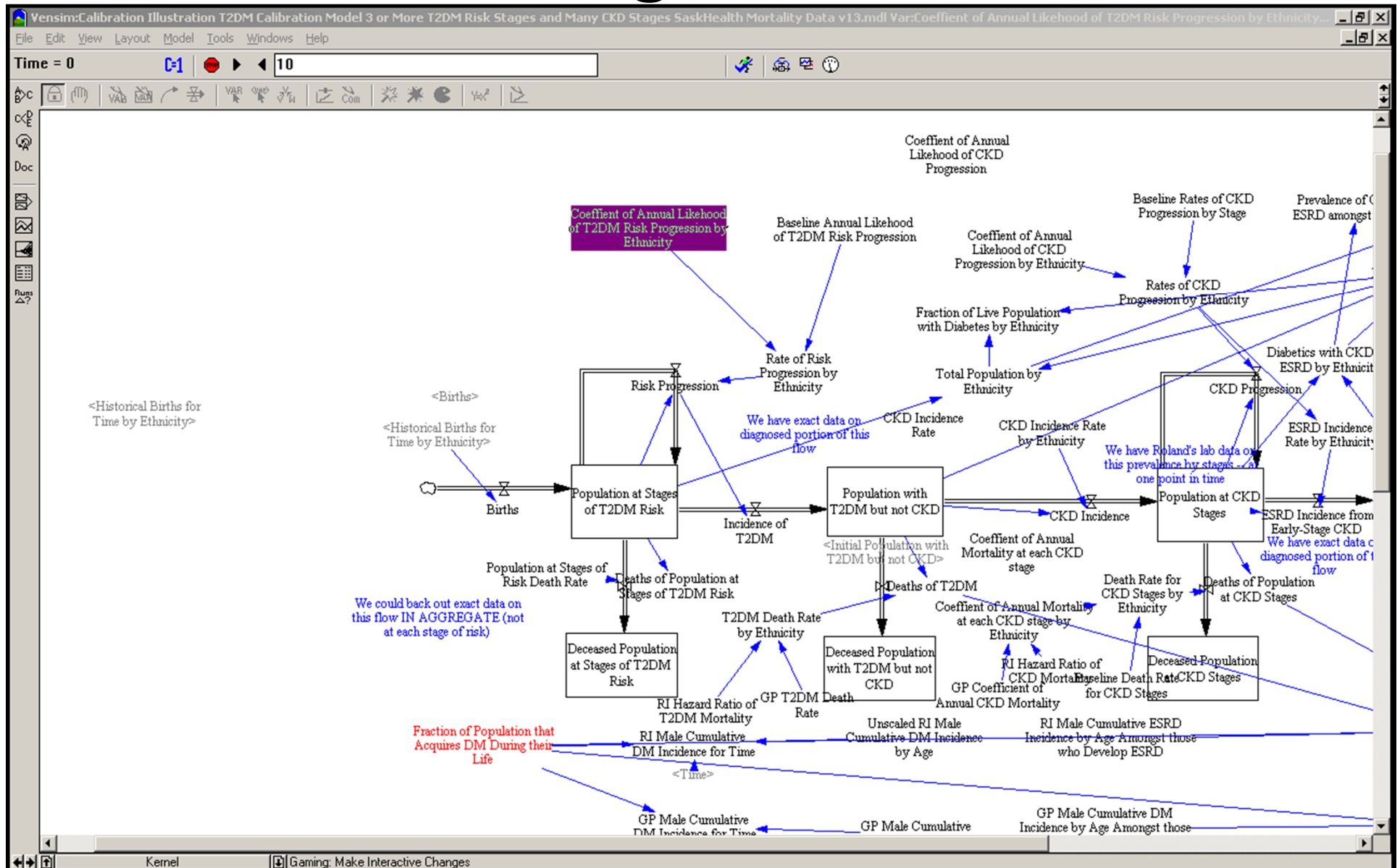
# “Gaming” Variables



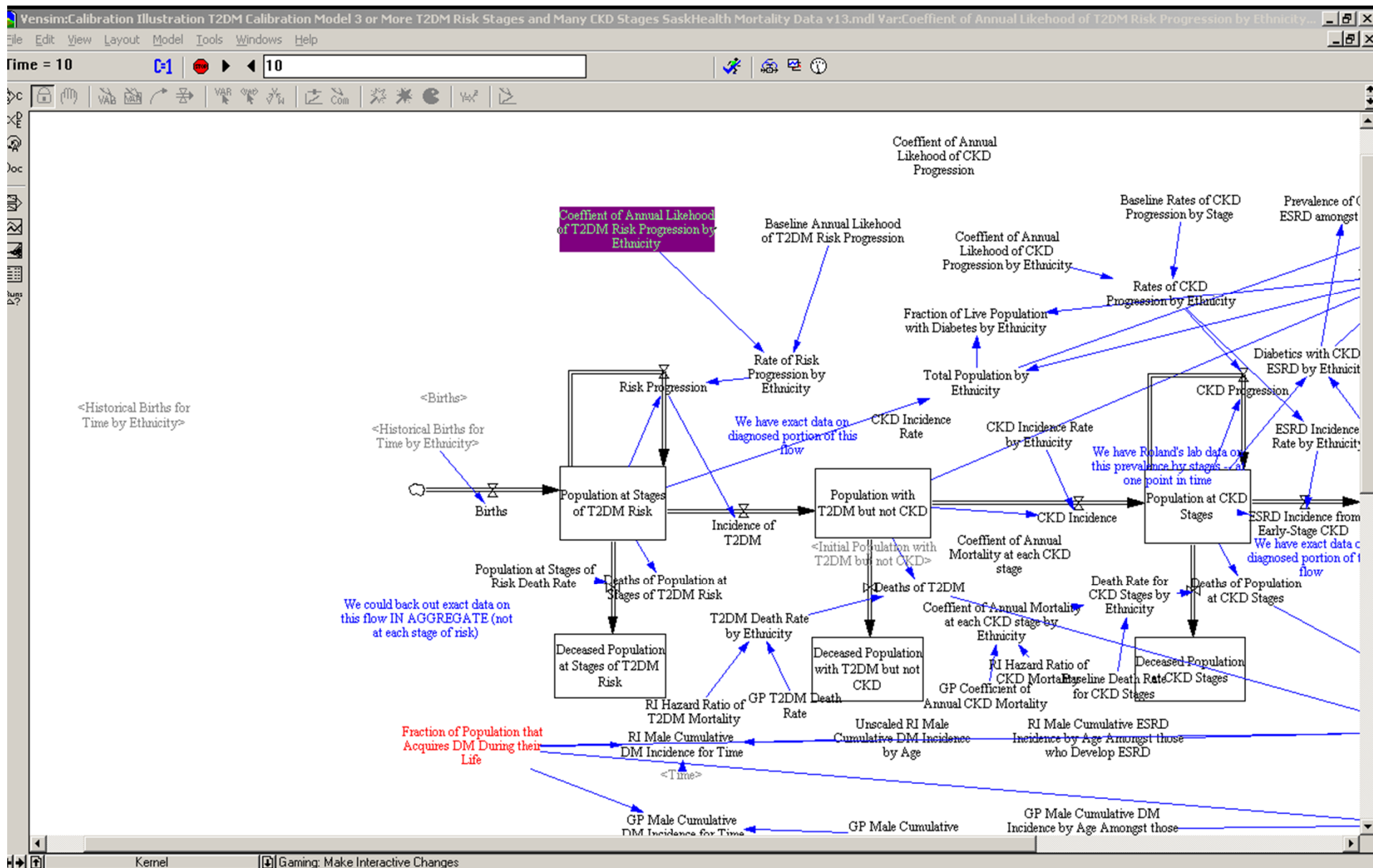
# Gaming



# Gaming Interface

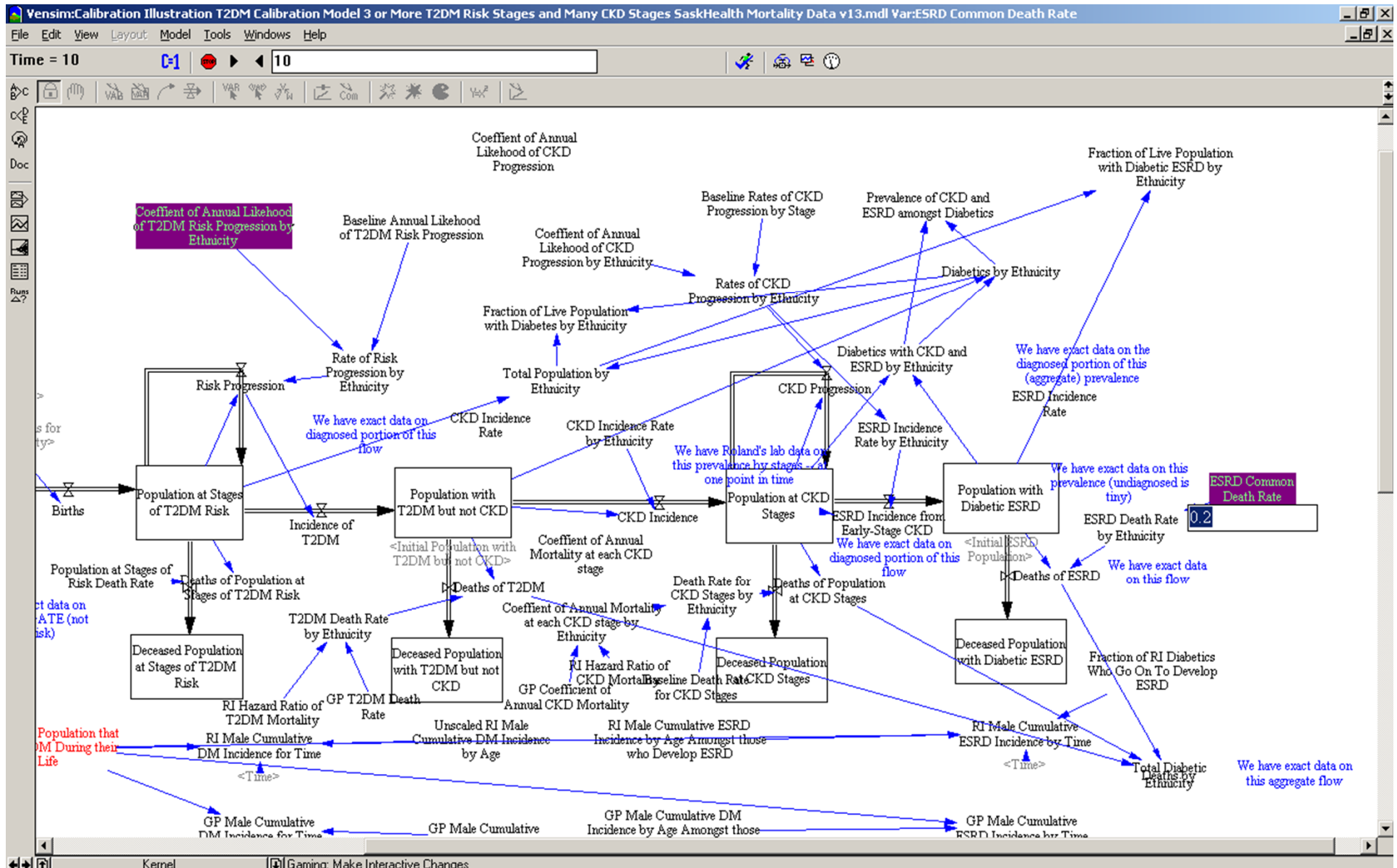


# Advancing time w/“Move Forward” Button

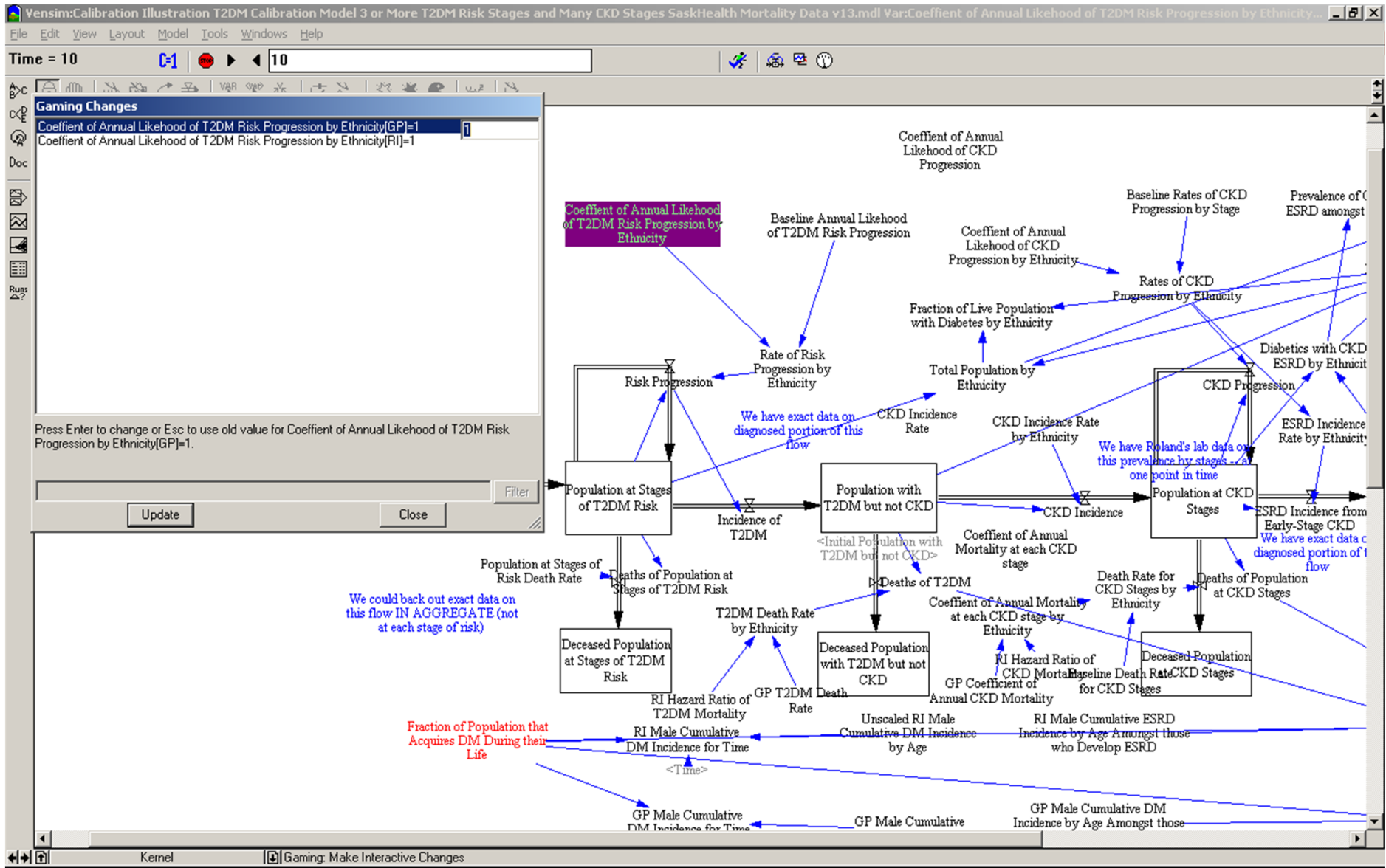




# Click to Change a Gaming Variable (Interface for a Non-Subscripted Variable)



# Click to Change a Gaming Variable (Interface for a Subscripted Variable)

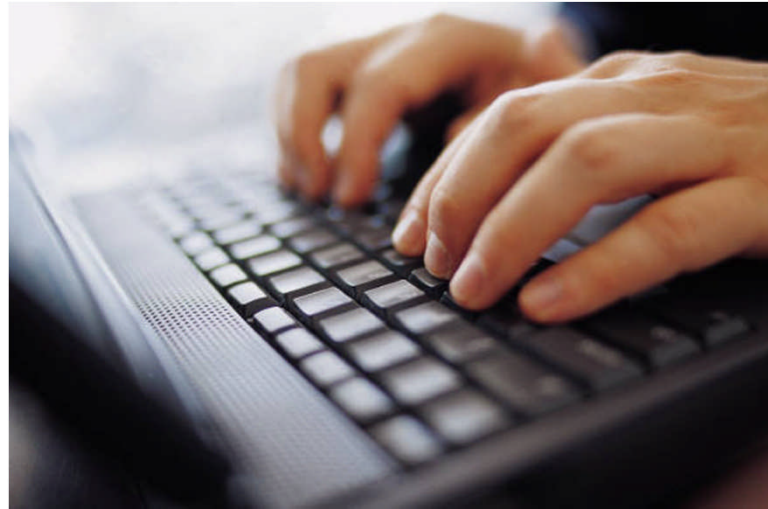


# Accomplishing “Live” Changes in AnyLogic via User Interface Elements

- Experiment User Interface normally just provides parameter values for starting up model
- Modifying an AnyLogic model’s operation during simulation itself can most easily be accomplished via a UI based in the Main object



## Hands on Model Use Ahead



Load Sample Model:

**Predatory Prey Agent Based**

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

# “Main” interface with Sliders

The screenshot displays the AnyLogic Advanced software interface, titled "AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main window shows a simulation titled "Predator Prey Agent Based Model" in a green box. The simulation area contains a control panel with four sliders for parameters: "Hare Births per Year" (value 123), "Hare Babies per Birth" (value 123), "Lynx Births per Year" (value 123), and "Lynx Babies per Birth" (value 123). To the right of the sliders is a black rectangular area representing the simulation environment. Below the sliders is a line graph showing the population dynamics of the model over time, with a red line and a green line. The graph has a y-axis ranging from 0 to 1.0 and an x-axis with vertical grid lines. The interface includes a menu bar (File, Edit, View, Model, Window, Help), a toolbar with various icons, and a palette on the right side with categories like Model, Parameter, Flow Aux Variable, Stock Variable, Event, Dynamic Event, Plain Variable, Collection Variable, Function, Table Function, Port, Connector, Entry Point, State, Transition, Initial State Pointer, Branch, History State, Final State, and Environment. The bottom of the interface shows a "Properties" window for the "Main - Active Object Class" with a "Name" field set to "Main" and an "Ignore" checkbox.

AnyLogic Advanced [EDUCATIONAL USE ONLY]

File Edit View Model Window Help

100%

Project Search

Model [simulation]

Predator Prey Agent Based Model

Change parameters on-the-fly

Hare Births per Year: 123

Hare Babies per Birth: 123

Lynx Births per Year: 123

Lynx Babies per Birth: 123

1

0.5

0

Properties Console

Main - Active Object Class

General Name: Main Ignore

Selection

# Slider Logic – Modifies Parameter

The screenshot displays the AnyLogic Advanced software interface. The main window shows a simulation titled "Predator Prey Agent Based Model" with a "[simulation]" indicator. The simulation area contains a text box with the heading "Change parameters on-the-fly" and four sliders. The sliders are labeled as follows:

- Hare Births per Year: 123
- Hare Babies per Birth: 123
- Lynx Births per Year: 123
- Lynx Babies per Birth: 123

The sliders are currently set to 123. The simulation area also features a black rectangular plot.

The Properties panel at the bottom shows the configuration for a "slider - Slider" component:

- General:** Name: slider,  Show Name,  Ignore,  Public,  Icon
- Advanced:** Orientation:  Horizontal,  Vertical
- Description:** Minimum Value: 1, Maximum Value: 6, Default Value: HareNatality, Enabled:
- Action:** `set_HareNatality( value );`

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

- associatedHealthcareProfessio
- residencyInstitution
- Functions
- categoryDescription
- Presentation
- AgentFactory
- Simulation: Main
- Predator Prey Agent Based
  - Hare
  - Lynx
  - Main
    - Parameters
    - Plain Variables
    - Functions
    - Embedded Objects
    - Analysis Data
    - Presentation
      - rect123
      - roundRect
      - AA text: Parame...
      - roundRect1
      - AA text1: Variab...

The right sidebar shows a palette of components including Parameter, Flow Aux Variable, Stock Variable, Event, Dynamic Event, Plain Variable, Collection Variable, Function, Table Function, Port, Connector, Entry Point, State, Transition, Initial State Pointer, Branch, History State, Final State, and Environment. Below the palette are buttons for Action, Analysis, Presentation, Connectivity, Enterprise Library, and More Libraries...



# Logic for Initial Values

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a simulation titled "Predator Prey Agent Based Model" with a subtitle "[experiment setup]". The simulation area contains two steps:

- Step 1. Set the initial number of hares and lynx**: This step features two sliders. The "Hares" slider is set to 123, and the "Lynx" slider is also set to 123. The sliders are horizontal and have a range from 1 to 10000.
- Step 2. Run the model**: This step contains a "Run" button.

The left sidebar shows a project tree with various components, including "Simulation: Main", "Plain Variables", "HaresInitial", "LynxInitial", and "Presentation". The bottom of the interface shows a "Properties" panel for a "slider - Slider" component. The "General" tab is active, showing the following settings:

- Name: slider
- Show Name:
- Ignore:
- Public:
- Icon:
- Orientation:  Horizontal  Vertical
- Minimum Value: 1
- Maximum Value: 10000
- Default Value: HaresInitial
- Enabled: `getEngine().getState() == Engine.IDLE`
- Action: `HaresInitial = (int)value;`

The right sidebar shows a "Palette" with various modeling elements, including "Parameter", "Flow Aux Variable", "Stock Variable", "Event", "Dynamic Event", "Plain Variable", "Collection Variable", "Function", "Table Function", "Port", "Connector", "Entry Point", "State", "Transition", "Initial State Pointer", "Branch", "History State", "Final State", and "Environment".

# Passing on Modified Parameter Values to the Simulation

The screenshot displays the AnyLogic Advanced software interface for an educational use. The main window shows a simulation titled "Predator Prey Agent Based Model" with an "experiment setup" label. The simulation area contains two steps: "Step 1. Set the initial number of hares and lynx" and "Step 2. Run the model". Under Step 1, there are sliders for "Hares: 123" and "Lynx: 123". A "Run" button is visible under Step 2.

The "Properties" window at the bottom shows the following parameters for the "Simulation - Simulation Experiment":

Category	Parameter Name	Value
General	HareMaxPerCell	15
Advanced	CellWidth	20
Model Time	Width	15
Presentation	LynxNatality	2
Window	LynxHuntingPeriod	0.01
Parameters	LynxNumberPerBirth	3
Description	LynxHungerDeathThreshold	0.03
	LynxLifeExpectancy	8
	HaresInitial	HaresInitial
	LynxInitial	LynxInitial



# Another Option

## Note: Slider Names Changed for Clarity

The screenshot displays the AnyLogic Advanced software interface for an educational use. The main workspace shows a simulation titled "Predator Prey Agent Based Model" with an "experiment setup" label. The simulation area contains two steps:

- Step 1. Set the initial number of hares and lynx**: This step features two sliders. The top slider is labeled "Hares: 123" and the bottom slider is labeled "Lynx: 123".
- Step 2. Run the model**: This step contains a single "Run" button.

The left sidebar shows a project tree with various elements, including "Simulation: Main", "Plain Variables", "HaresInitial", "LynxInitial", and "Presentation". The bottom panel displays the "Simulation - Simulation Experiment" properties, which are organized into a table:

Category	Parameter	Value
General	HareMaxPerCell	15
Advanced	CellWidth	20
Model Time	Width	15
Presentation	LynxNatality	2
Window	LynxHuntingPeriod	0.01
<b>Parameters</b>	LynxNumberPerBirth	3
Description	LynxHungerDeathThreshold	0.03
	LynxLifeExpectancy	8
	HaresInitial	sliderInitialHares.value
	LynxInitial	sliderInitialLynx.value

The right sidebar contains a "Palette" with various modeling elements such as "Parameter", "Flow Aux Variable", "Stock Variable", "Event", "Dynamic Event", "Plain Variable", "Collection Variable", "Function", "Table Function", "Port", "Connector", "Entry Point", "State", "Transition", "Initial State Pointer", "Branch", "History State", "Final State", and "Environment". At the bottom right, there are buttons for "Action", "Analysis", "Presentation", "Connectivity", "Enterprise Library", and "More Libraries...".

# Setting Initial Values

The screenshot shows the AnyLogic software interface for configuring a simulation. The window title is "experiment:" and the AnyLogic logo is in the top right corner. The main area has a light yellow background and contains the following text and controls:

**Predator Prey Agent Based Model** [experiment setup]


The predator prey problem is a simulation that attempts to predict the relationship in populations between a population of lynx and hares isolated on an island. Enjoy this old classic example with the help of the new Agent Based technology!

**Step 1.** Set the initial number of hares and lynx

**Hares:** 5,000

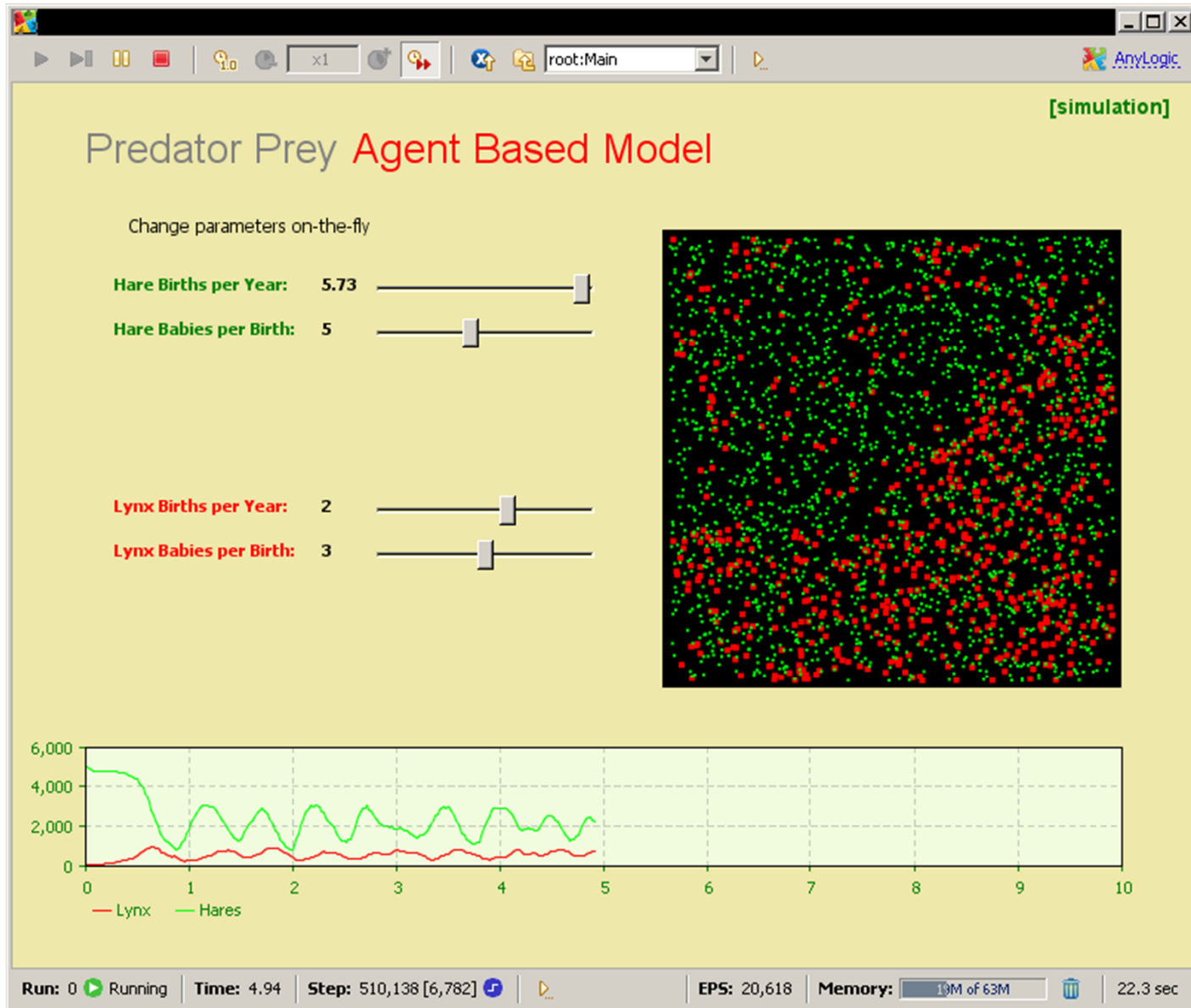
**Lynx:** 40

**Step 2.** Run the model

 technologies  
AnyLogic and this model is (c) XJ Technologies, www.anylogic.com. All rights reserved.

Run: 0  Idle | Time: 0.00 | Step: 0 [0] | EPS: 0 | Memory: 9M of 63M | 0.0 sec

# Run-Time Parameter Modifications



# Changing Parameters in AnyLogic

- Changing value of parameter explicitly in model
  - Avoid if possible -- could forget to restore
- Create a new experiment
  - Set the parameter value as a parameter for Main
  - Here, easiest if the operational parameter in Main!
  - If parameter is not located in “Main”, Main should “pass on” parameter value to e.g. the agent class
- Via an interface in the main class or agent class itself

# Structural Modifications

- Sometimes, capturing the effects of an intervention requires representing a different processes than are present in the baseline model
- e.g.
  - Vaccination
  - Quarantine
  - Intervention group
    - Educated
    - Given a treatment
  - Genetically immune mosquitoes

# Capturing Structural Modifications in Vensim

- Adding
  - Stocks (e.g. Vaccinated people, quarantined people)
  - Flows (e.g. to vaccinated stock, or quarantined stock)
  - Subscripts
    - e.g. Intervention group (may run in parallel with other group, subject to the same forces)
- Modifying existing flows
  - E.g. disabling smoking relapse when intervention is enabled

# Capturing Structural Modifications in AnyLogic

- Statechart based: Adding
  - States (e.g. Vaccinated, quarantined)
  - Transitions (e.g. to vaccinated state, or quarantined state, or to a new “cured” state)
- System Dynamics: flows
- Modifying an existing transition so that it is contingent on an intervention being disabled
- For targeted intervention, may wish to capture people as having been affected by the intervention

# Representing Intervention Mechanisms: Two Choices

- Some interventions are represented in a stylized fashion that abstracts away from *dynamics of intervention implementation*
  - Here, we just examine proximal & distal effects of certain modifications to baseline model assumptions, ignoring the issue of how these modifications would be achieved
- Some intervention representations include characterizing both the intervention effects & its dynamics e.g.
  - Dynamics of training teachers to deliver anti-smoking lessons in the classroom
  - Dynamics of vaccine production



# Endogenous Intervention Impacts on Behaviour: Current Practice

- Behaviour is exogenous to many models
- Models link behavior to distal impacts
- Modelers impose assumptions of how interventions affect behaviour
- Models offer value in understanding emergent, distal implications of behaviour change
- We gain little insight into the counter-intuitive behavioral impacts of intervention

# Example Behavioral Feedbacks

## Underlying Much Policy Resistance

- Cutting cigarette tar levels reduces cessation
- Cutting cigarette nicotine levels leads to compensatory smoking
- ARVs prolong lives of HIV carriers, but lower risk perception
- Availability of reduced-fat/calorie varieties undercuts changes to eating habits
- Antilock brakes lead to more risky driving

# Endogenous Intervention Impacts

## on Behaviour: Vision

- Modelers characterize intervention impacts on environment (e.g. prices, tax burden, \$ incentives, laws)
- Capture indiv preferences&mental models, learning
- Model endogenously compute individual, localized behavioural responses (cf discrete choice theory, psych. models)
- Models provide insight into both
  - Distal implications of interventions
  - Behavioral impacts of intervention (individual&collective)

# Additional Factors

- Accumulating costs for interventions
- Accumulating costs for other factors (so can see what intervention eliminates)