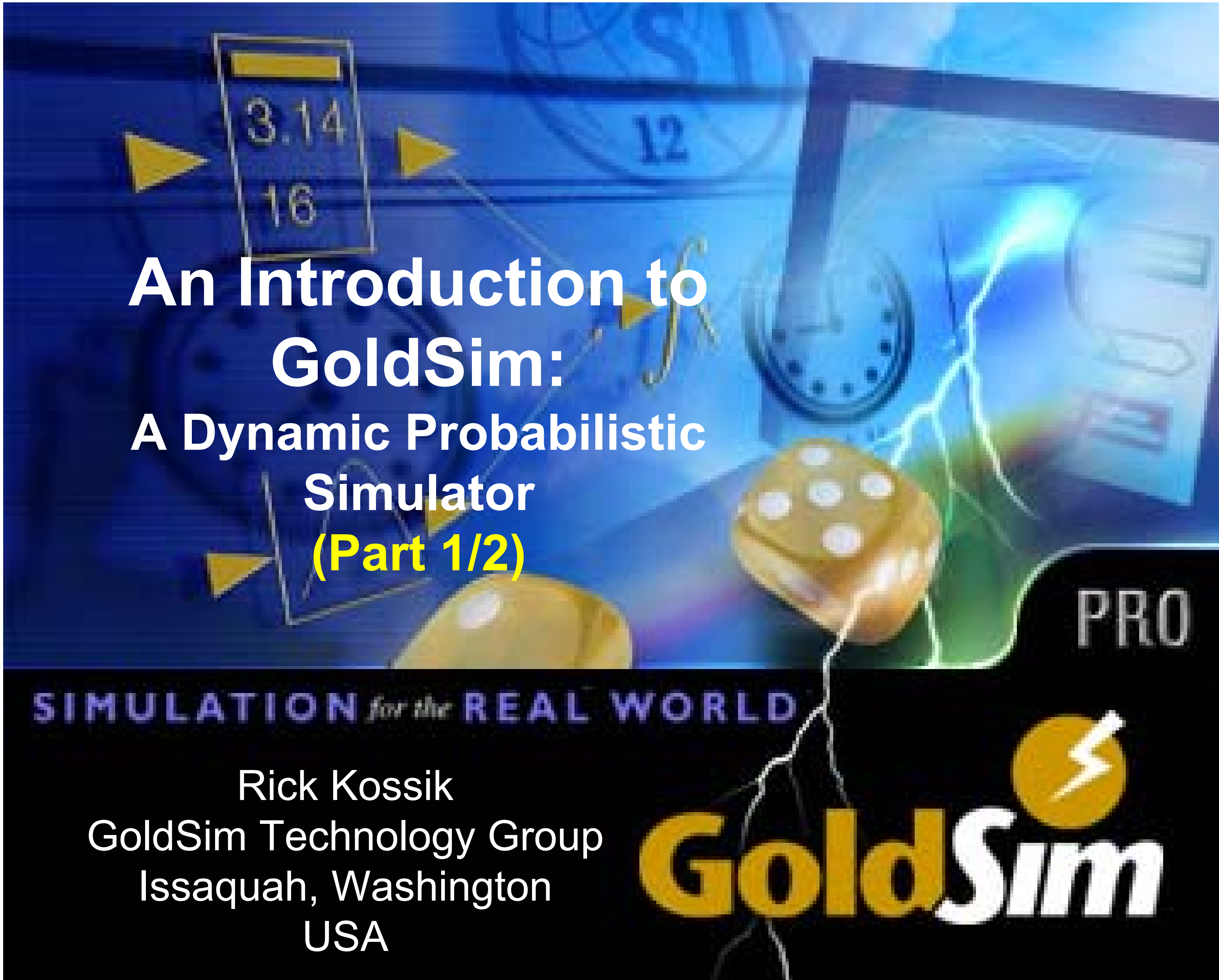
An Introduction to GoldSim:

A Dynamic Probabilistic Simulator

(Part 1/2)



Rick Kossik
GoldSim Technology Group
Issaquah, Washington
USA

Outline

- What is GoldSim and where did it come from?
- A summary of the major differences between GoldSim and traditional SD codes
- Basic GoldSim Features
- Overview of Advanced GoldSim Features
- Overview of GoldSim Extension Modules
- Can GoldSim complement traditional SD codes?
- Questions and Discussion

Outline

- **What is GoldSim and where did it come from?**
- **A summary of the major differences between GoldSim and traditional SD codes**
- **Basic GoldSim Features**
- **Overview of Advanced GoldSim Features**
- **Overview of GoldSim Extension Modules**
- **Can GoldSim complement traditional SD codes?**
- **Questions and Discussion**

GoldSim Technology Group

- **Originally a division of Golder Associates**
  – International civil and environmental engineering firm
- **Began developing GoldSim in 1990**
  – First customers were US Department of Energy and analogous government organizations in Europe and Asia
  – Focused on risk analysis for complex engineered systems associated with waste management
- **Started marketing software in 2002**
  – Rapidly expanded into other related engineering arenas (mining, water resources, failure analysis, long-term strategic planning)
- **Became independent company in February 2004**
  – GoldSim represents over 50 man-years of development
  – Over 1,000,000 lines of code (C++)

What were the main drivers behind the development of GoldSim?

- **Systems being evaluated had lots of uncertainty and involved stochastic processes**
- **Clients required predictions of future performance in order to optimize system design and meet regulatory requirements**
- **Evaluations needed to be transparent and easy to explain to multiple audiences.**

Goal was to create a probabilistic simulation framework that could be applied to complex and diverse engineering and scientific problems

Outline

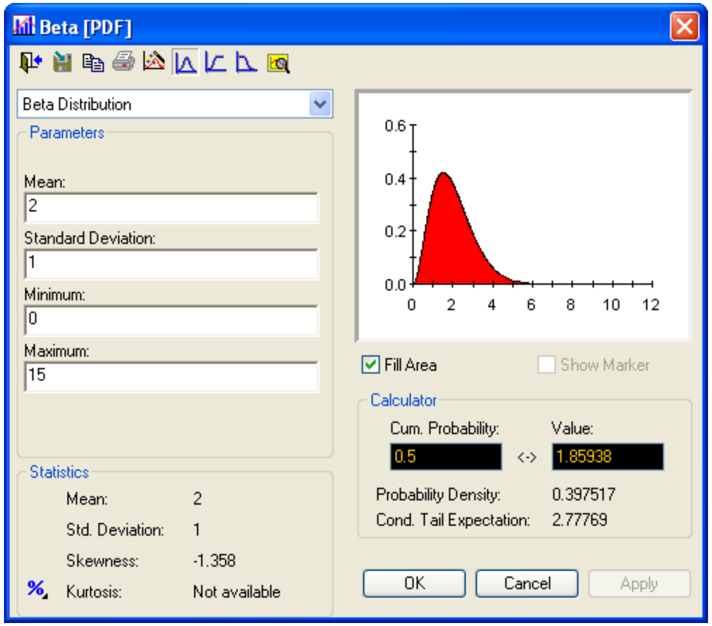
- What is GoldSim and where did it come from?
- **A summary of the major differences between GoldSim and traditional SD codes**
- Basic GoldSim Features
- Overview of Advanced GoldSim Features
- Overview of GoldSim Extension Modules
- Can GoldSim complement traditional SD codes?
- Questions and Discussion

GoldSim TECHNOLOGY GROUP

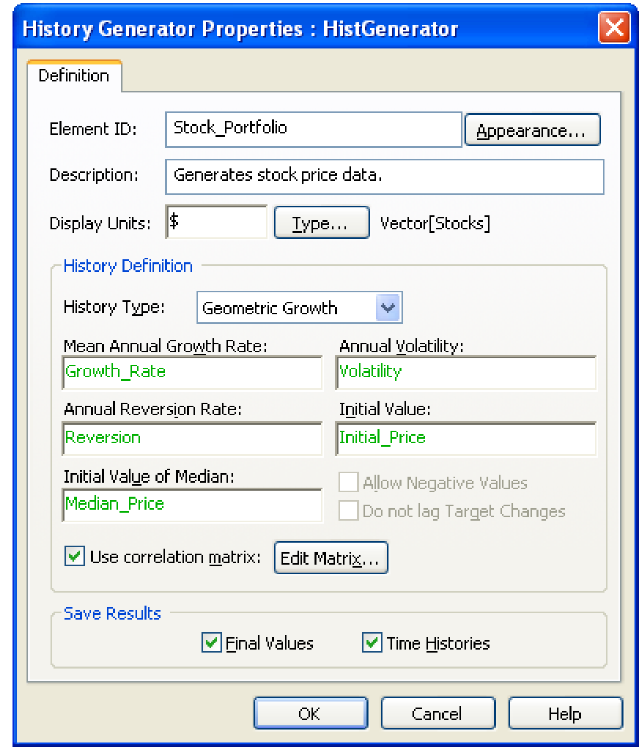
What are the key differences between GoldSim and traditional SD tools?

- **GoldSim puts much greater emphasis on probabilistic simulation and producing probabilistic predictions of performance**

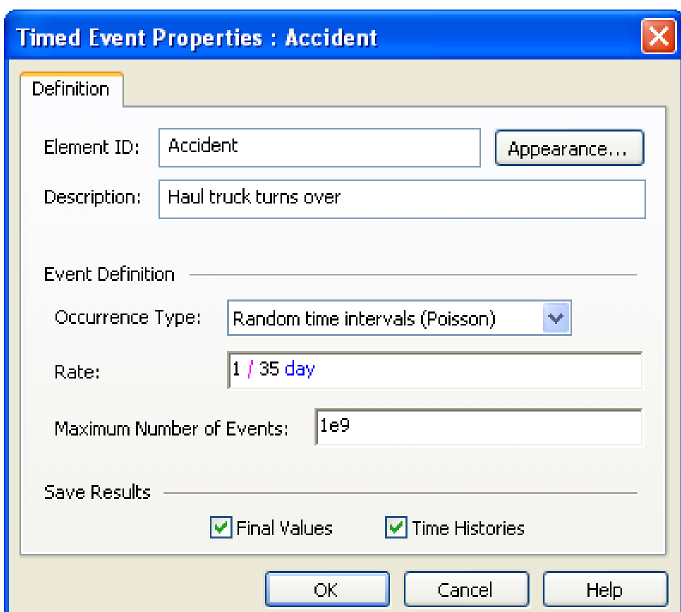
Define uncertain variables



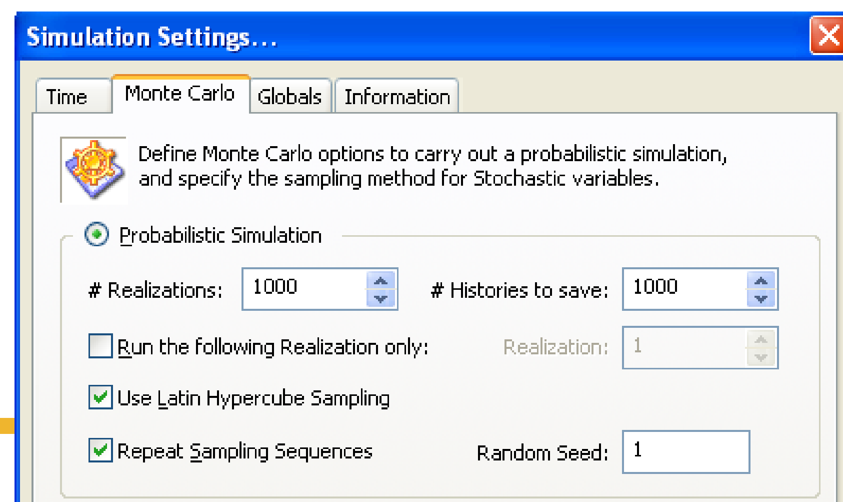
Define stochastic time series

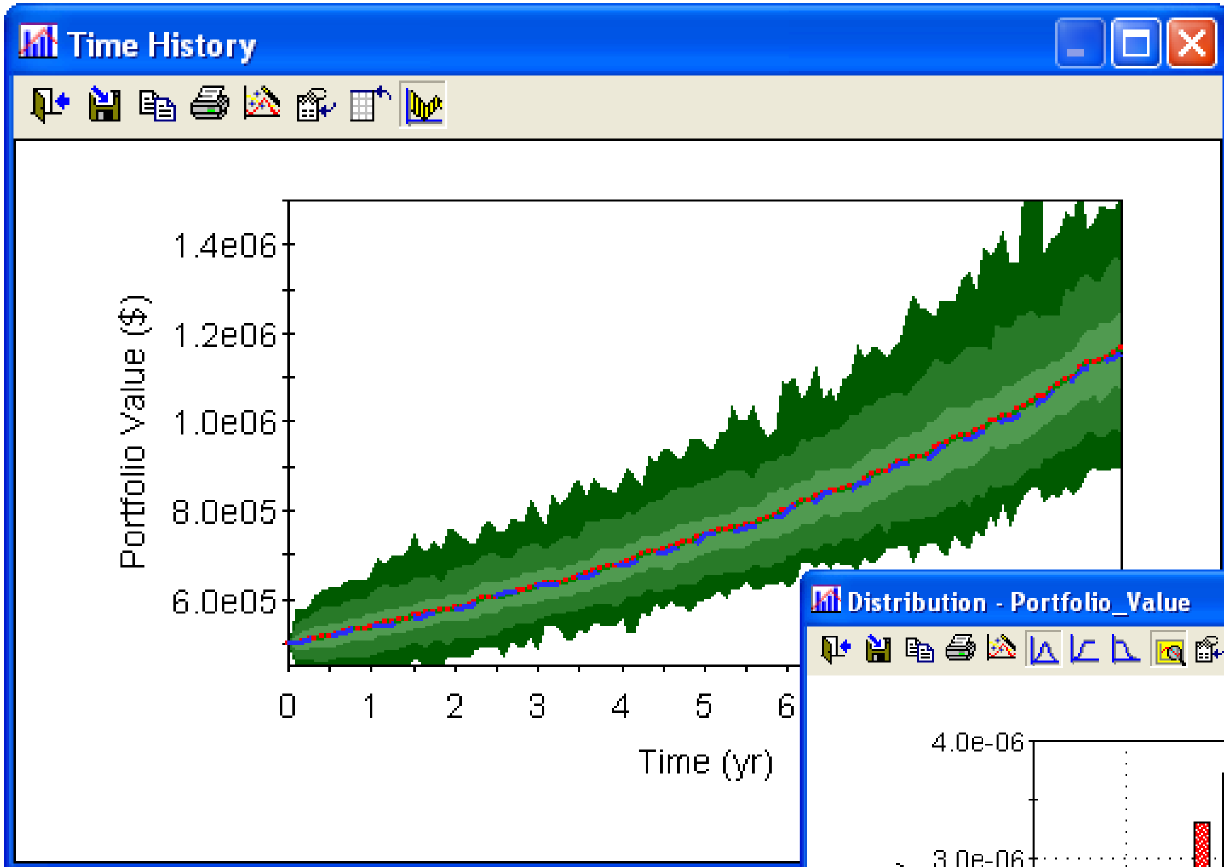


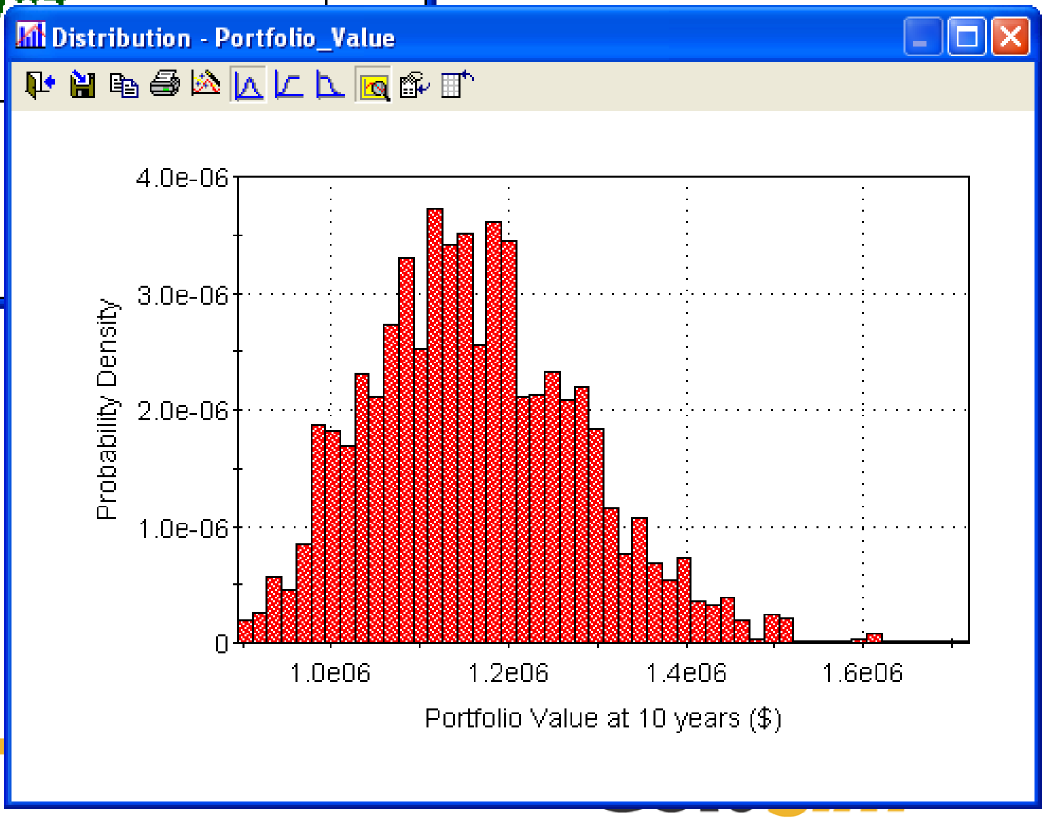
Specify statistical-defined events



Define Monte Carlo simulation options



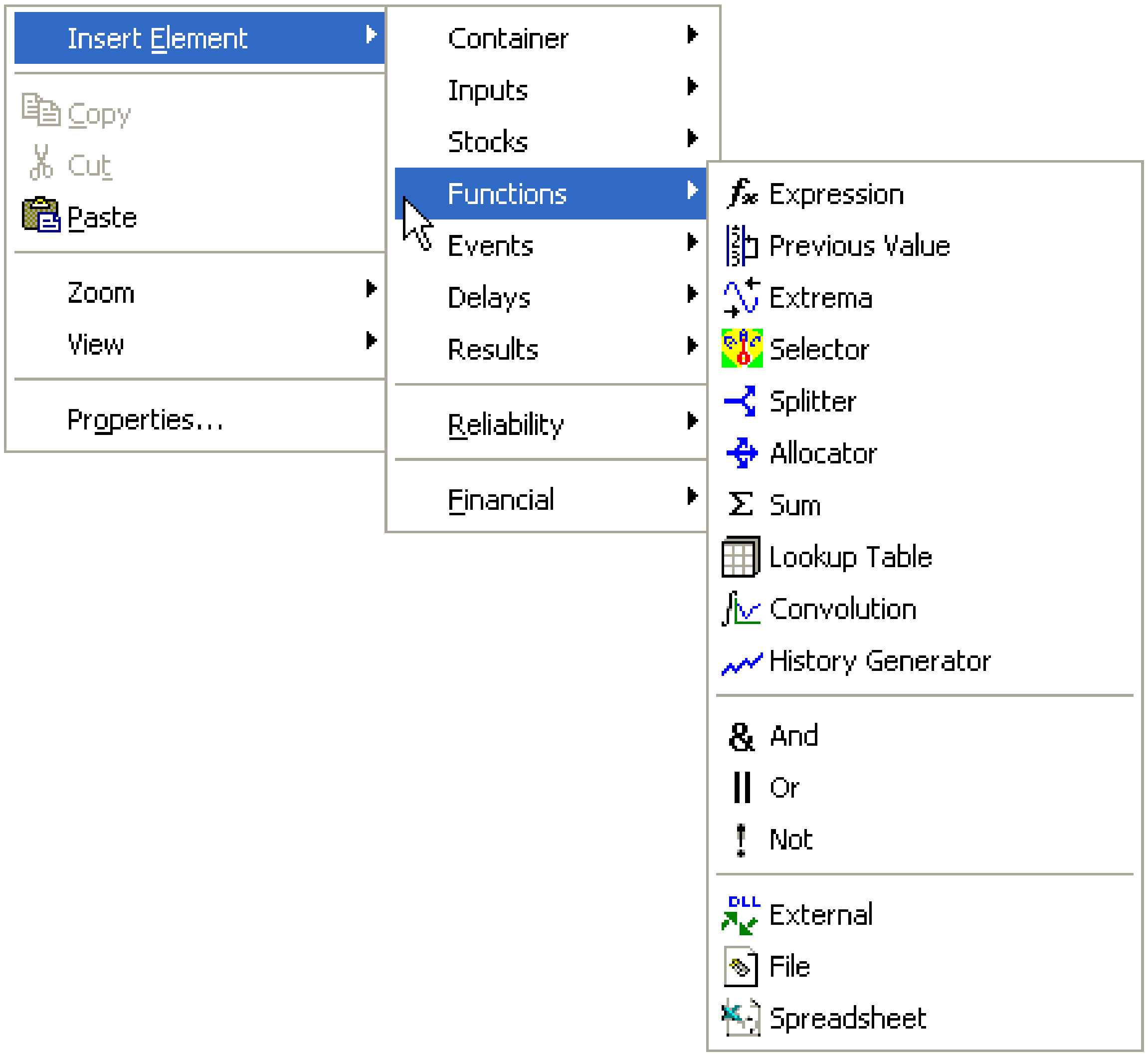

Time History
Portfolio Value ($)
1.4e06
1.2e06
1.0e06
8.0e05
6.0e05
0
1
2
3
4
5
6
Time (yr)

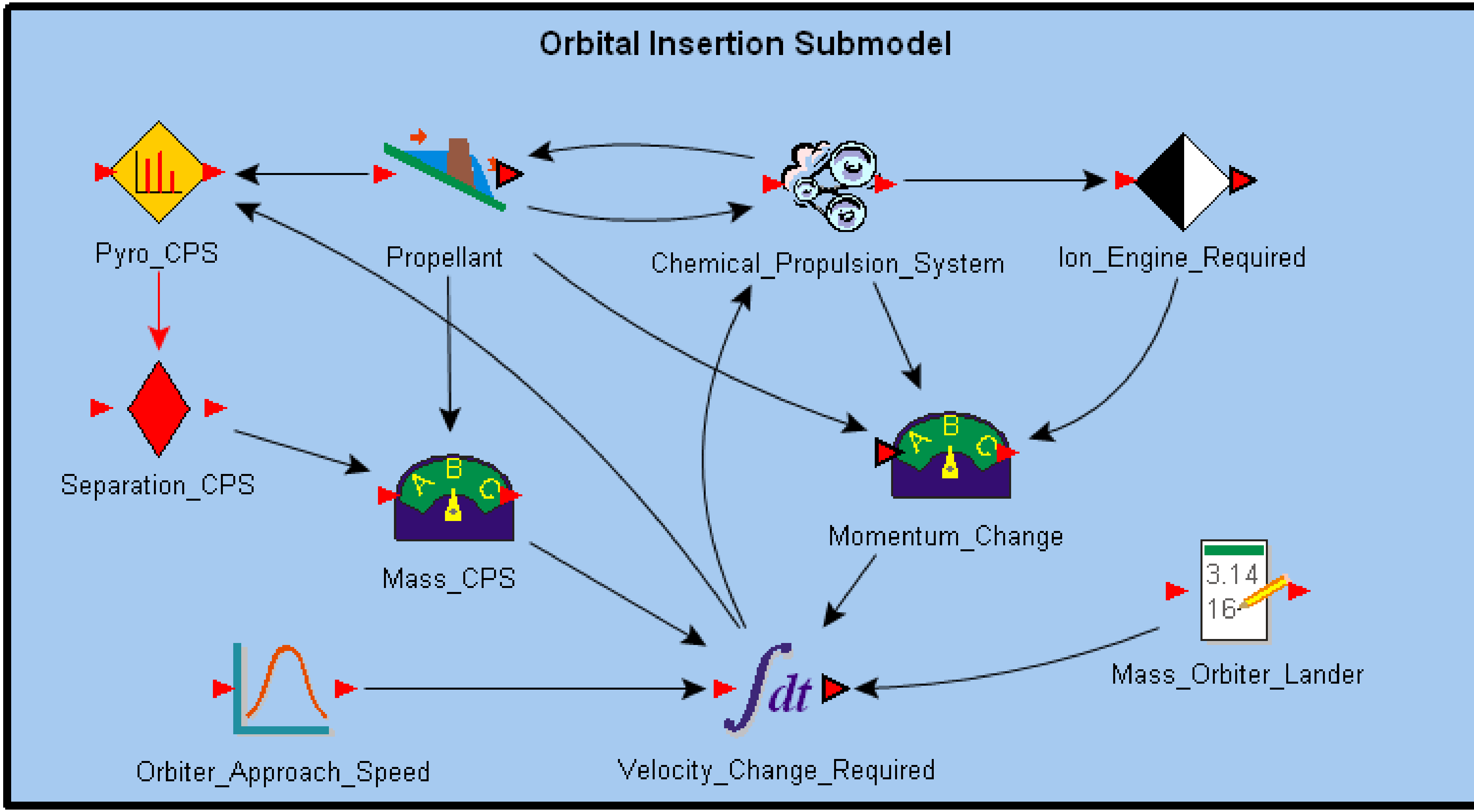

Distribution - Portfolio_Value
Probability Density
4.0e-06
3.0e-06
2.0e-06
1.0e-06
0
1.0e06
1.2e06
1.4e06
1.6e06
Portfolio Value at 10 years ($)

What are the key differences between GoldSim and traditional SD tools?

- GoldSim puts much greater emphasis on probabilistic simulation and producing probabilistic predictions of performance
- **GoldSim provides a much broader range of model objects**
  - Makes the model logic and structure less abstract and more transparent
  - Includes objects to superimpose discrete dynamics on a continuous system

GoldSim provides over 40 element types…

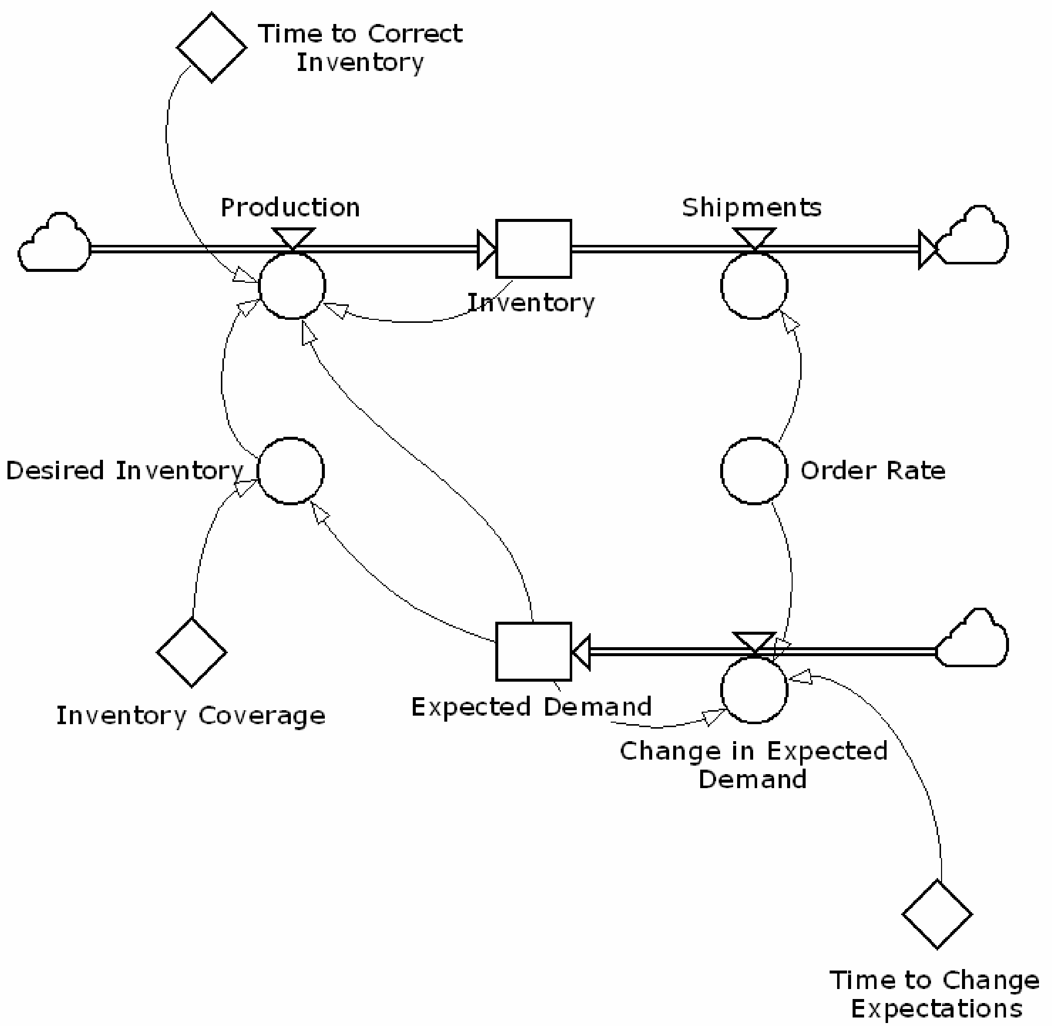



Orbital Insertion Submodel
Pyro_CPS
Propellant
Chemical_Propulsion_System
Ion_Engine_Required
Separation_CPS
Mass_CPS
Momentum_Change
A
B
C
3.14
16
Mass_Orbiter_Lander
dt
Orbiter_Approach_Speed
Velocity_Change_Required

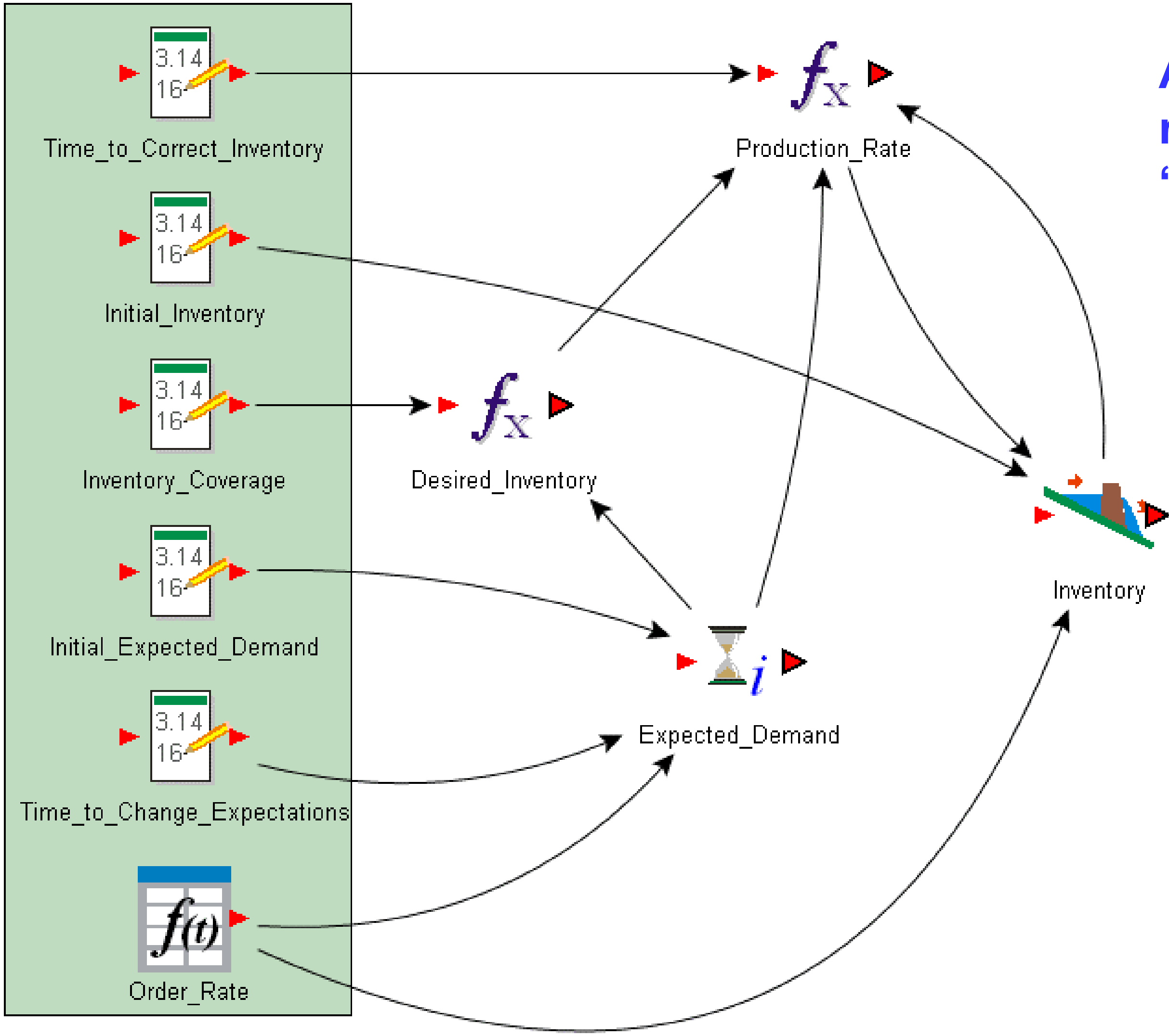
What are the key differences between GoldSim and traditional SD tools?

- GoldSim puts much greater emphasis on probabilistic simulation and producing probabilistic predictions of performance
- **GoldSim provides a much broader range of model objects (elements)**
  - Makes the model logic and structure less abstract and more transparent
  - Includes objects to superimpose discrete dynamics on a continuous system
  - An important implication is that GoldSim does not use the "stock and flow" paradigm

Example: Simple Inventory Model



Example: Simple Inventory Model

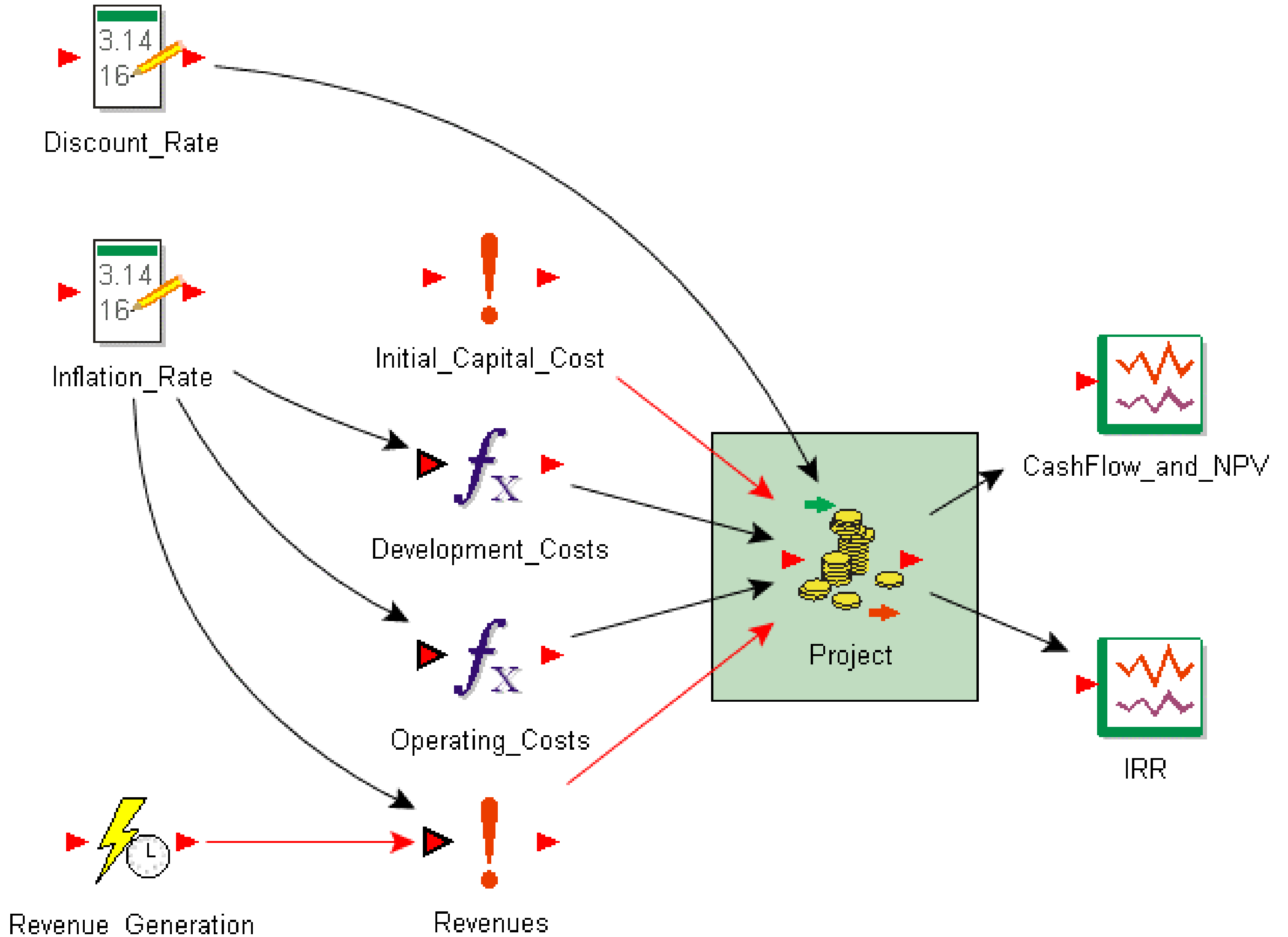


Arrows all represent "influences"

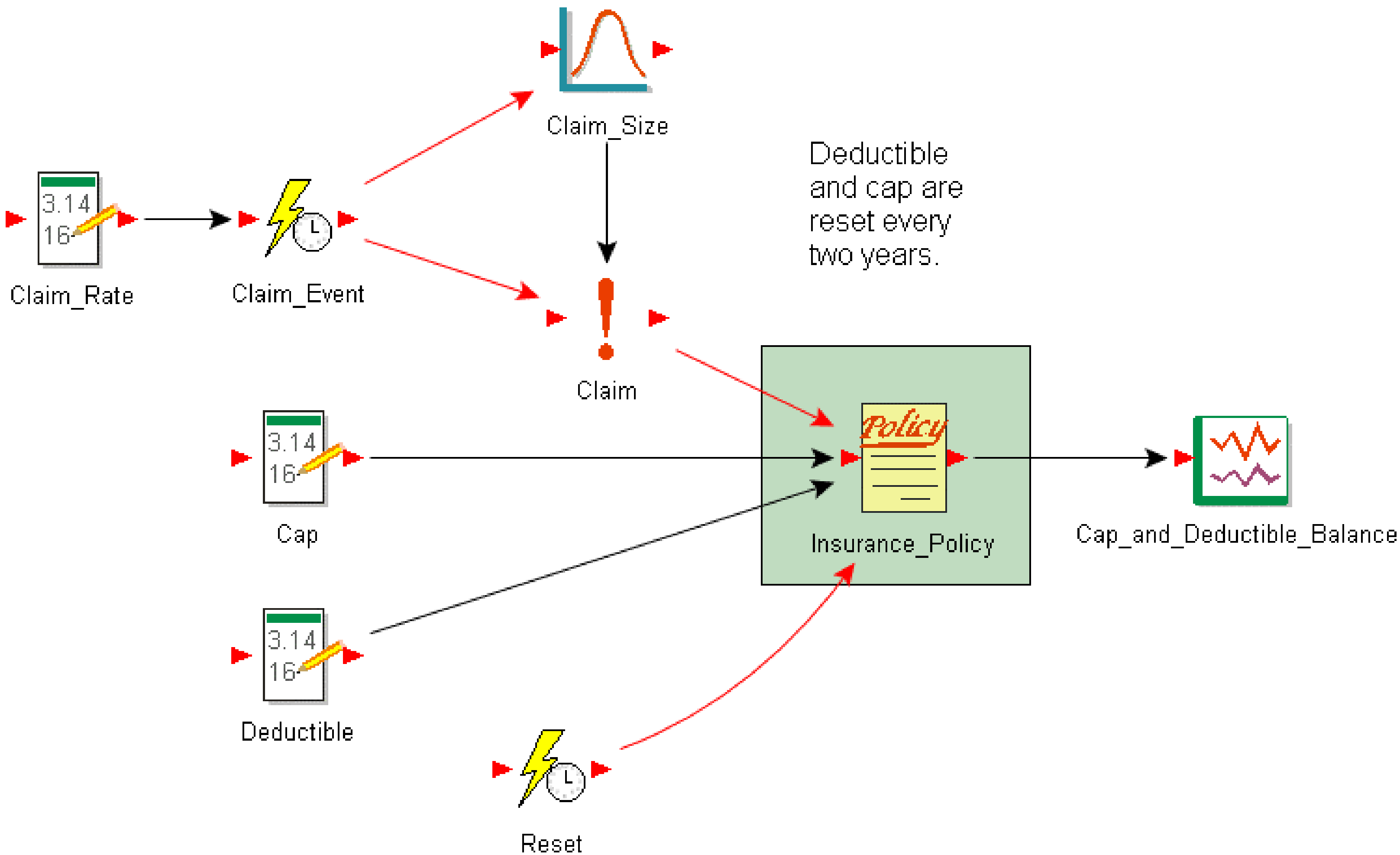
What are the key differences between GoldSim and traditional SD tools?

- GoldSim puts much greater emphasis on probabilistic simulation and producing probabilistic predictions of performance
- GoldSim provides a much broader range of model objects
- **GoldSim was designed to accommodate the addition of specialized extension modules**
  - Modules either address processes that can't be adequately represented using simpler constructs, or add to model transparency

Financial Module

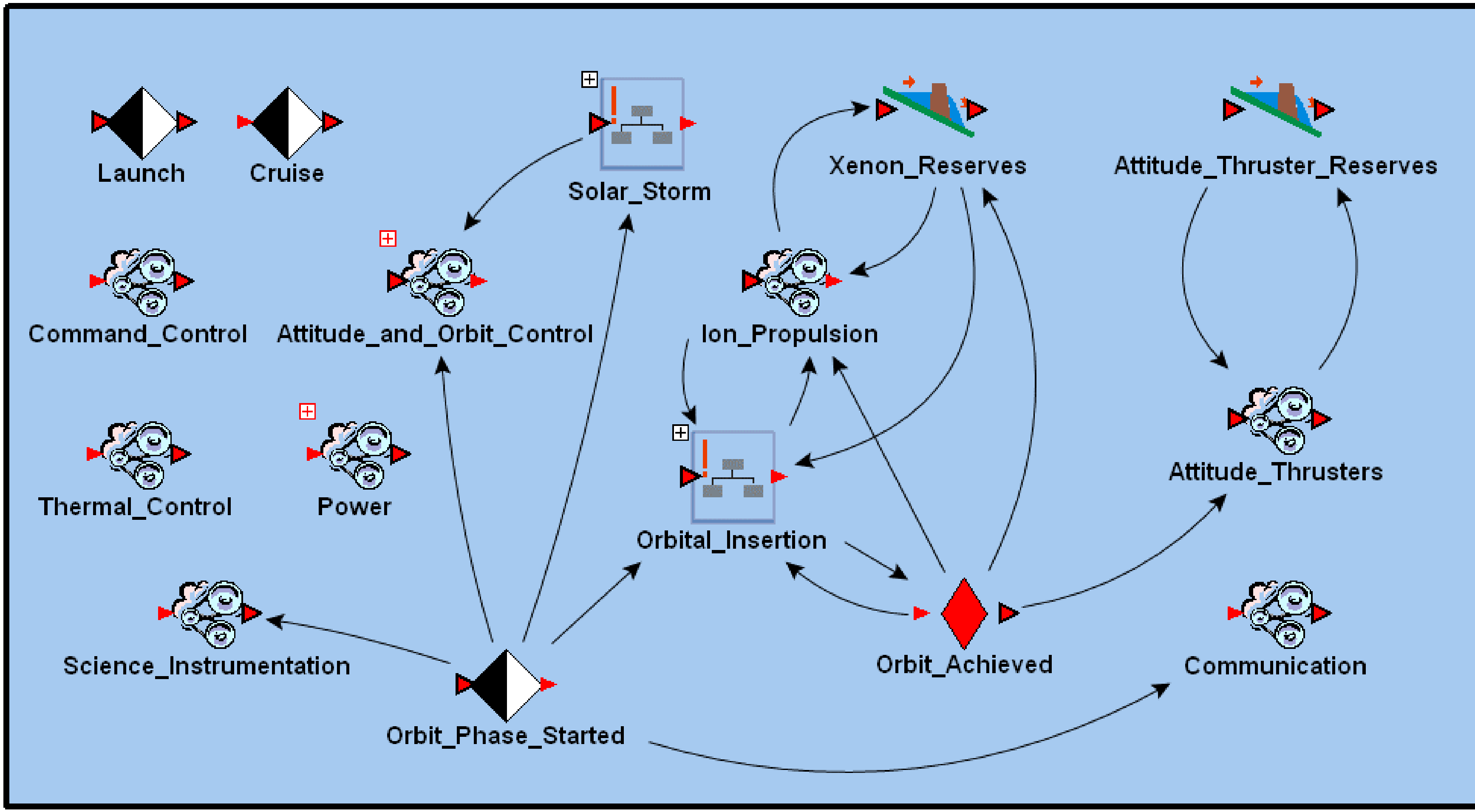


Financial Module



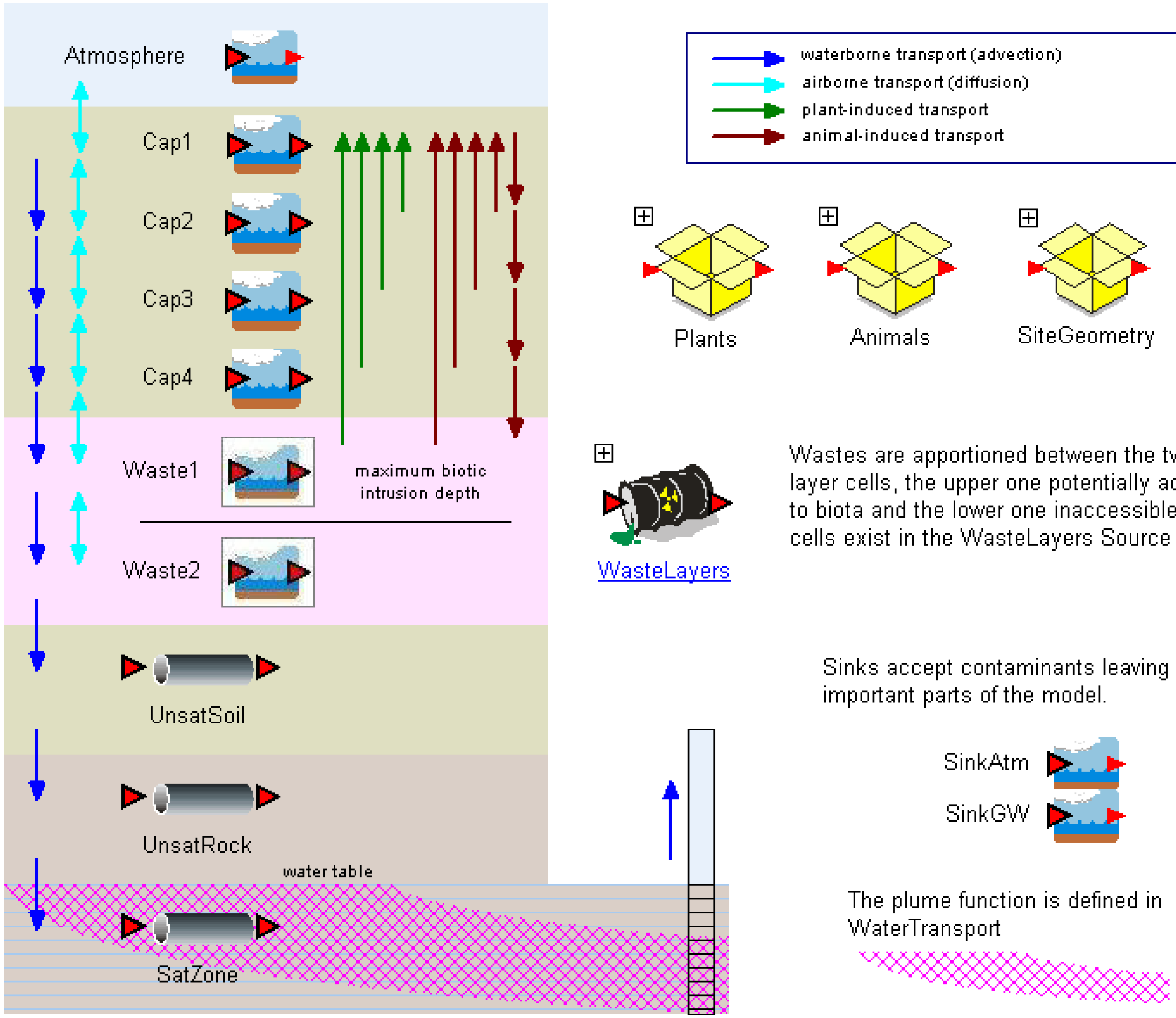
Reliability Module

Risk Analysis for Planetary Orbiter



GoldSim TECHNOLOGY GROUP

Contaminant Transport Module



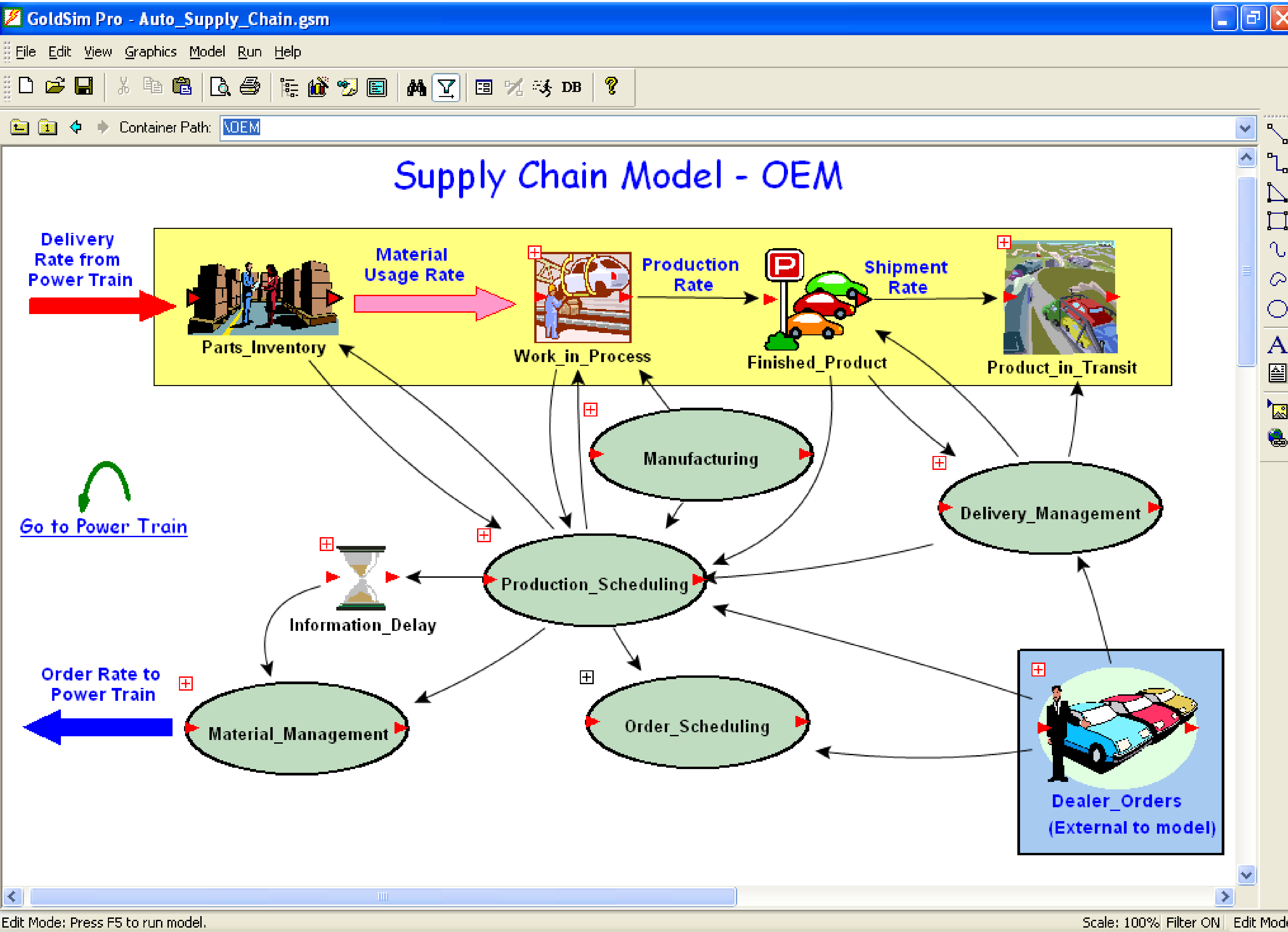
Wastes are apportioned between the two waste layer cells, the upper one potentially accessible to biota and the lower one inaccessible. These cells exist in the WasteLayers Source Container.

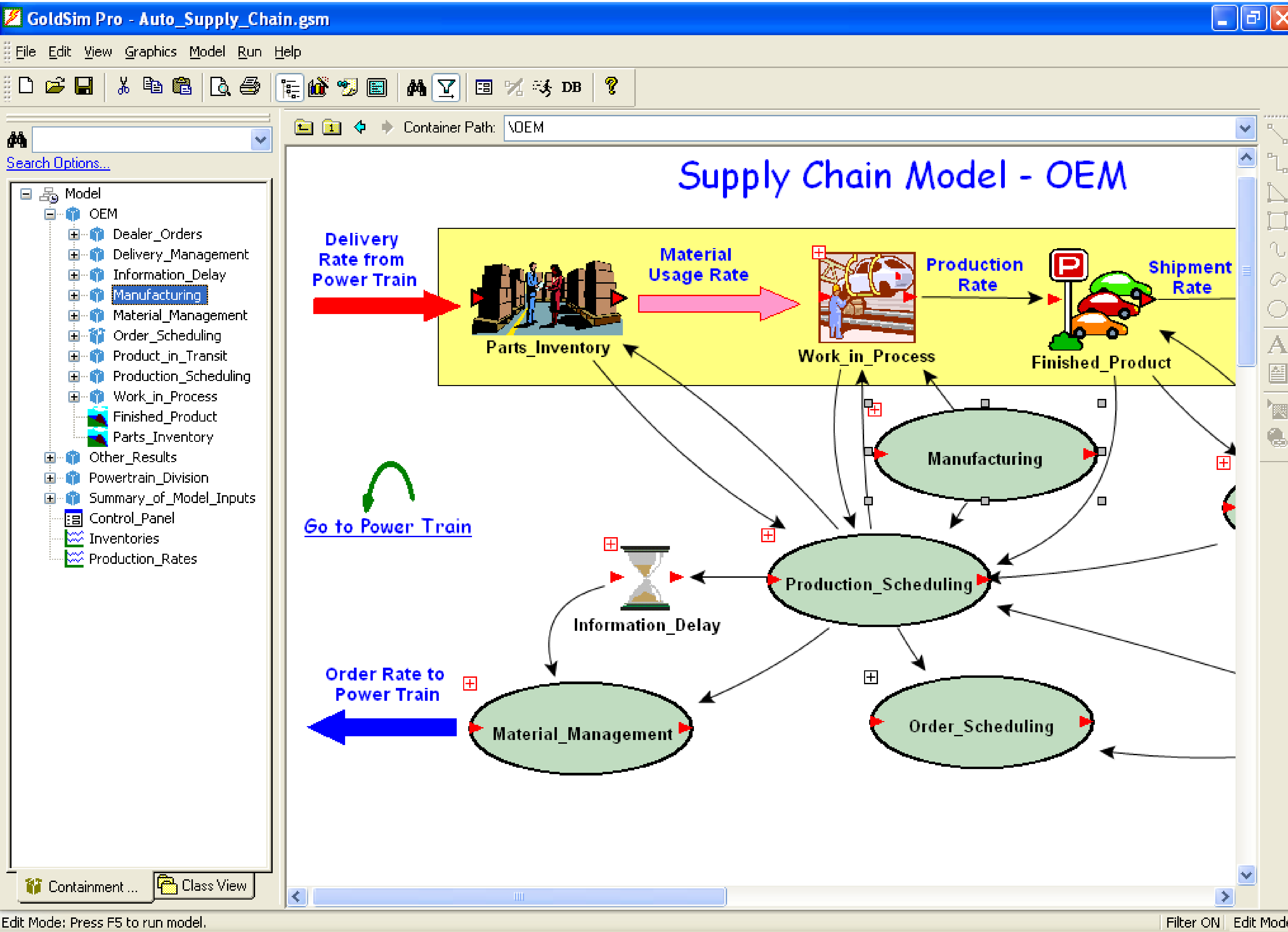
Sinks accept contaminants leaving the important parts of the model.

The plume function is defined in WaterTransport

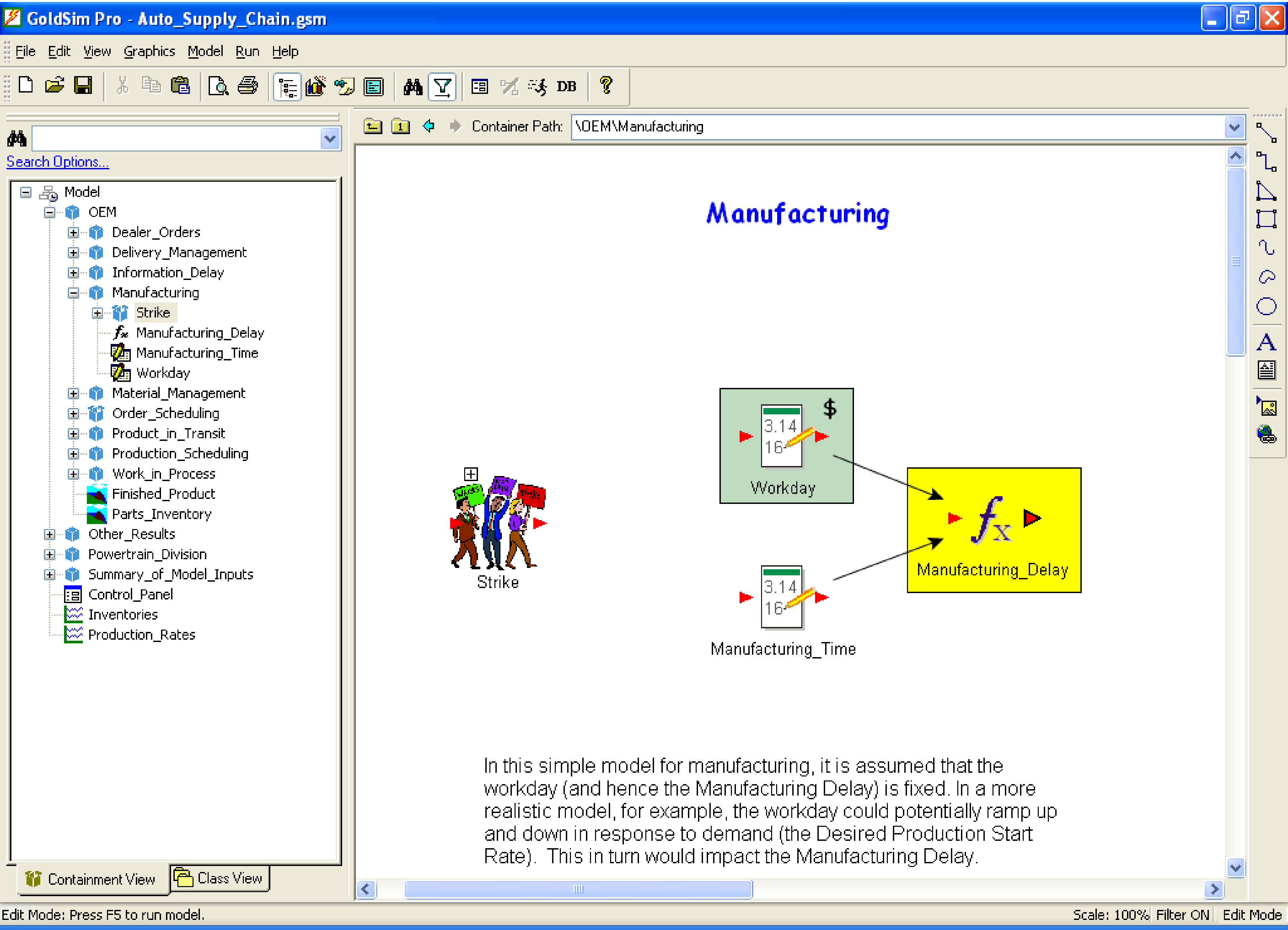
GoldSim models are typically deeply hierarchical

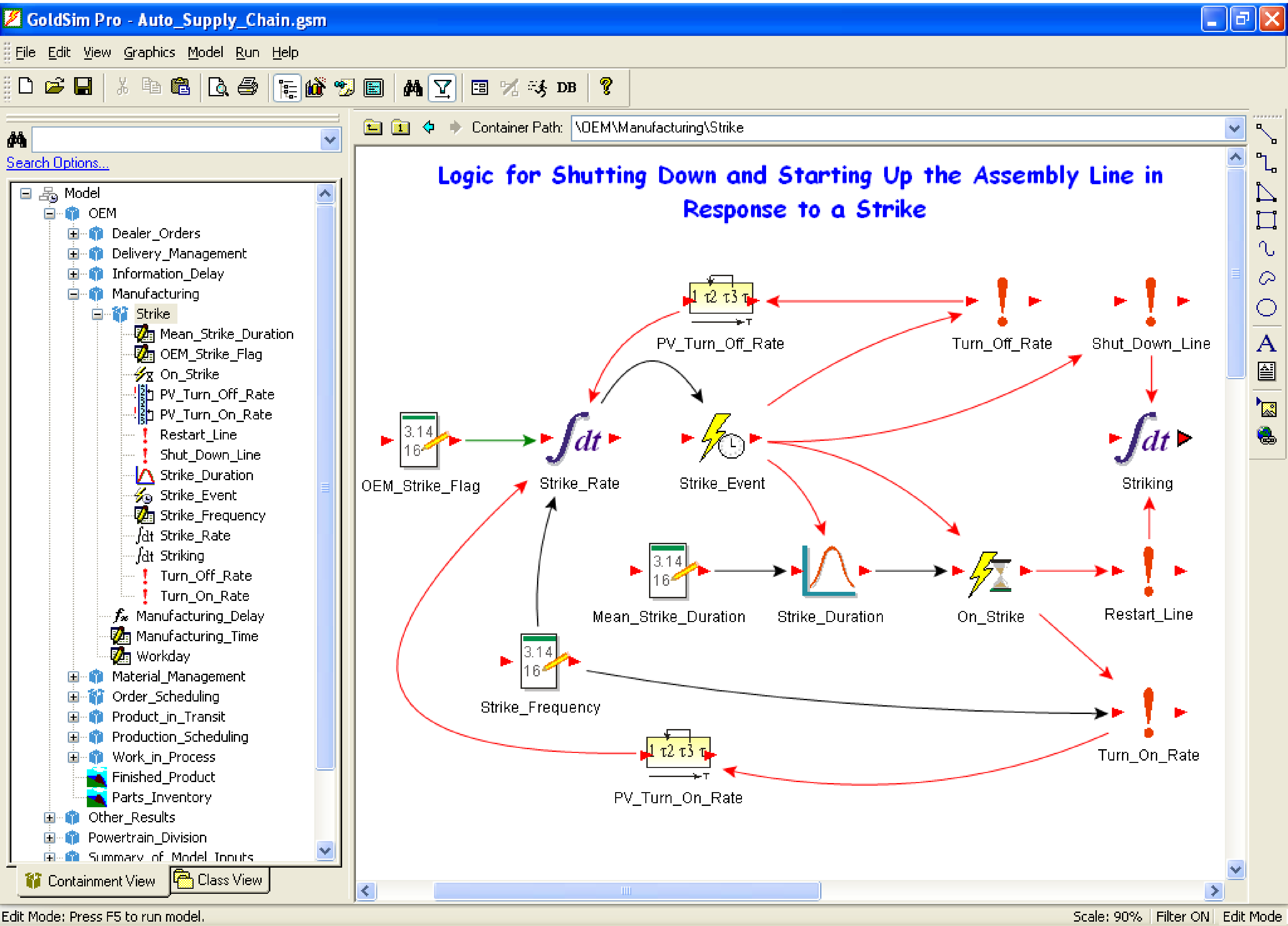
- **Due to the nature of our original user base, GoldSim was designed to support very large models**
  - Largest model to date has 35,000 elements
- **To support this, GoldSim provides:**
  - Unlimited nesting of hierarchical models
  - Local variables


GoldSim Pro - Auto_Supply_Chain.gsm
File Edit View Graphics Model Run Help
Container Path: \OEM
Supply Chain Model - OEM
Delivery Rate from Power Train
Material Usage Rate
Production Rate
Shipment Rate
Parts_Inventory
Work_in_Process
Finished_Product
Product_in_Transit
Manufacturing
Delivery_Management
Go to Power Train
Production_Scheduling
Information_Delay
Order Rate to Power Train
Material_Management
Order_Scheduling
Dealer_Orders
(External to model)
Edit Mode: Press F5 to run model.
Scale: 100% Filter ON Edit Mode



GoldSim Pro - Auto_Supply_Chain.gsm
File Edit View Graphics Model Run Help
Search Options...
Model
OEM
Dealer_Orders
Delivery_Management
Information_Delay
Manufacturing
Material_Management
Order_Scheduling
Product_in_Transit
Production_Scheduling
Work_in_Process
Finished_Product
Parts_Inventory
Other_Results
Powertrain_Division
Summary_of_Model_Inputs
Control_Panel
Inventories
Production_Rates
Containment ...
Class View
Container Path: \OEM
Supply Chain Model - OEM
Delivery Rate from Power Train
Material Usage Rate
Production Rate
Shipment Rate
Parts_Inventory
Work_in_Process
Finished_Product
Manufacturing
Go to Power Train
Production_Scheduling
Information_Delay
Order Rate to Power Train
Material_Management
Order_Scheduling
Edit Mode: Press F5 to run model.
Filter ON
Edit Mode


GoldSim Pro - Auto_Supply_Chain.gsm
File Edit View Graphics Model Run Help
Search Options...
Container Path: \OEM\Manufacturing
Model
OEM
Dealer_Orders
Delivery_Management
Information_Delay
Manufacturing
Strike
Manufacturing_Delay
Manufacturing_Time
Workday
Material_Management
Order_Scheduling
Product_in_Transit
Production_Scheduling
Work_in_Process
Finished_Product
Parts_Inventory
Other_Results
Powertrain_Division
Summary_of_Model_Inputs
Control_Panel
Inventories
Production_Rates
Containment View
Class View
Manufacturing
Workday
Strike
Manufacturing_Delay
Manufacturing_Time
In this simple model for manufacturing, it is assumed that the workday (and hence the Manufacturing Delay) is fixed. In a more realistic model, for example, the workday could potentially ramp up and down in response to demand (the Desired Production Start Rate). This in turn would impact the Manufacturing Delay.
Edit Mode: Press F5 to run model.
Scale: 100% Filter ON Edit Mode


GoldSim Pro - Auto_Supply_Chain.gsm
File Edit View Graphics Model Run Help
Search Options...
Model
OEM
Dealer_Orders
Delivery_Management
Information_Delay
Manufacturing
Strike
Mean_Strike_Duration
OEM_Strike_Flag
On_Strike
PV_Turn_Off_Rate
PV_Turn_On_Rate
Restart_Line
Shut_Down_Line
Strike_Duration
Strike_Event
Strike_Frequency
Strike_Rate
Striking
Turn_Off_Rate
Turn_On_Rate
Manufacturing_Delay
Manufacturing_Time
Workday
Material_Management
Order_Scheduling
Product_in_Transit
Production_Scheduling
Work_in_Process
Finished_Product
Parts_Inventory
Other_Results
Powertrain_Division
Containment View
Class View
Container Path: \OEM\Manufacturing\Strike
Logic for Shutting Down and Starting Up the Assembly Line in Response to a Strike
PV_Turn_Off_Rate
Turn_Off_Rate
Shut_Down_Line
OEM_Strike_Flag
Strike_Rate
Strike_Event
Striking
Mean_Strike_Duration
Strike_Duration
On_Strike
Restart_Line
Strike_Frequency
Turn_On_Rate
PV_Turn_On_Rate
Edit Mode: Press F5 to run model.
Scale: 90%
Filter ON
Edit Mode

GoldSim models are typically deeply hierarchical

- Due to the nature of our original user base, GoldSim was designed to support very large models
  - Largest model to date has 35,000 elements
- To support this, GoldSim provides:
  - Unlimited nesting of hierarchical models
  - Local variables
  - This, by definition, makes feedback loops more difficult to see graphically
    - **We provide other tools to find loops**

Outline

- What is GoldSim and where did it come from?
- A summary of the major differences between GoldSim and traditional SD codes
- **Basic GoldSim Features**
- Overview of Advanced GoldSim Features
- Overview of GoldSim Extension Modules
- Can GoldSim complement traditional SD codes?
- Questions and Discussion

GoldSim Simulation Philosophy

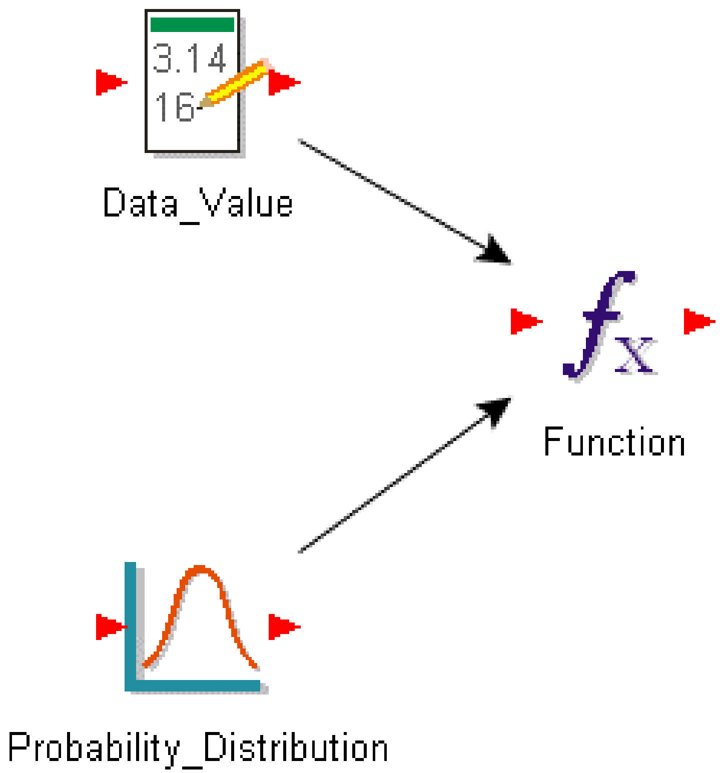
- **Models should be constructed in a "top-down" manner**
  - Capture all key aspects and inter-relationships
  - Only add as much detail as required and justified
  - Keep focused on the "big picture"
- **Must accurately and honestly express our uncertainty in all aspects of the system**
  - parameters
  - processes
  - events
- **A model that cannot be explained and understood is a model that will not be used**
  - No black boxes!

GoldSim Features Reflect this Philosophy

- **Scalable and Extensible**
  - Design allows you to build a simple model, and then add details in a hierarchical manner as warranted
  - Can link to other programs (e.g., spreadsheets, user programs)
  - Designed to facilitate addition of custom modules for specific applications
- **Can represent uncertainty and stochasticity in parameters, processes and events**
- **Specialized objects make models less abstract and more transparent**
- **Powerful navigation, presentation and documentation features allow you to build, maintain and present complex models**

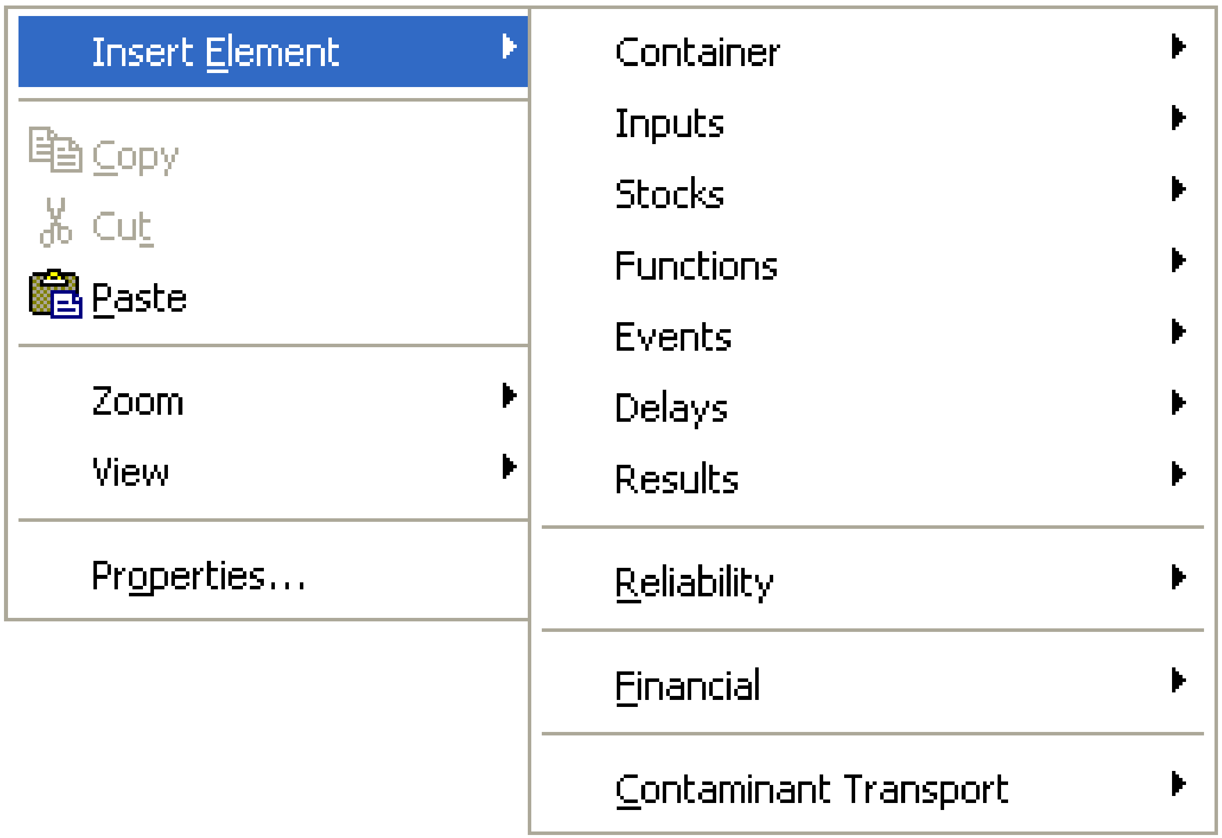
GoldSim TECHNOLOGY GROUP

What Kind of Simulator is GoldSim?



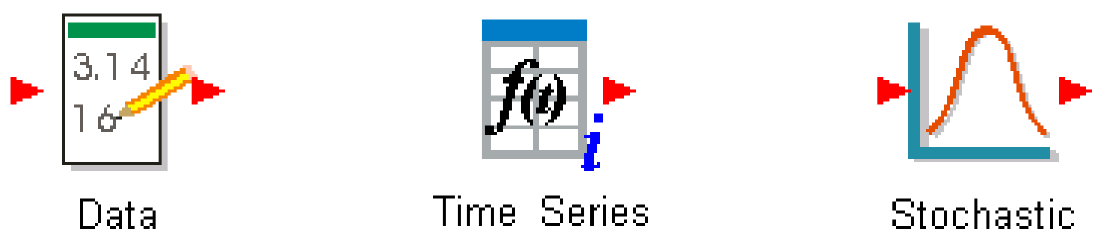
The user creates a model by using GoldSim objects (called elements) to draw a schematic or influence diagram of the system being simulated.

Element Categories



Input Elements

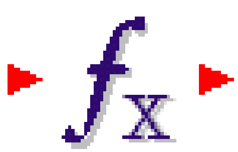
- Some elements provide a mechanism for you to enter Data into a model. You can specify a single scalar datum, or vectors and matrices of data.
- Time Series elements allow you to specify a time series of data
- You can also specify that a particular datum is uncertain, by defining it as a probability distribution (referred to as a Stochastic).



GoldSim
TECHNOLOGY GROUP

Function Elements

- Other elements act as **functions**, which operate on one or more inputs and produce one or more outputs.
- The simplest function element is the **Expression**.
- You define an Expression by simply typing in an equation. Similar to a cell in a spreadsheet, when defining an expression, you can use a wide variety of operators and functions.



Expression

3*sin(a)   min(x,y)   if(b>10,x,y)   bess(a,b)

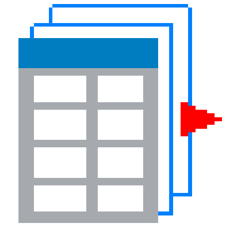
exp(k*time)   log(x/y)   (x-y) * (z+v)

Dimensions and Units in GoldSim

- **GoldSim is dimensionally aware.**
- **GoldSim elements are all strongly typed (unit, scalar/vector/matrix, value/condition).**
- **GoldSim has an extensive database of units and conversion factors. You can enter data and display results in any units. You can even define your own custom units.**
- **When elements are linked, GoldSim enforces dimensional consistency and carries out all unit conversions internally.**

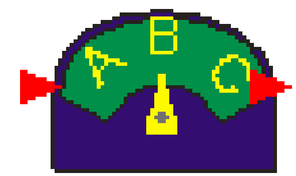
Examples of Other Function Elements

- Another example of a function element is the **Look-Up Table**. In this element, the output is computed by interpolating between the values of a user-defined table (1D, 2D, or 3D).



Table

- Other function elements have more complex behavior. For example, a **Selector** allows you to specify nested "if, then" logic in your model (it acts like a switch with multiple settings).



Selector

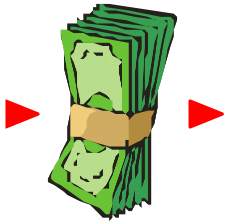
- The **Extrema** element computes the highest (peak) or lowest (valley) value achieved by its input.

Extrema

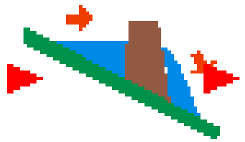
Dynamic Elements (Stocks)

- **Dynamic element outputs are determined by the previous values of their inputs.**

- **Two types of dynamic elements are the Fund and the Reservoir. In their simplest form, these elements require an initial value, rates of additions (deposits) and withdrawals, and output a current value:**



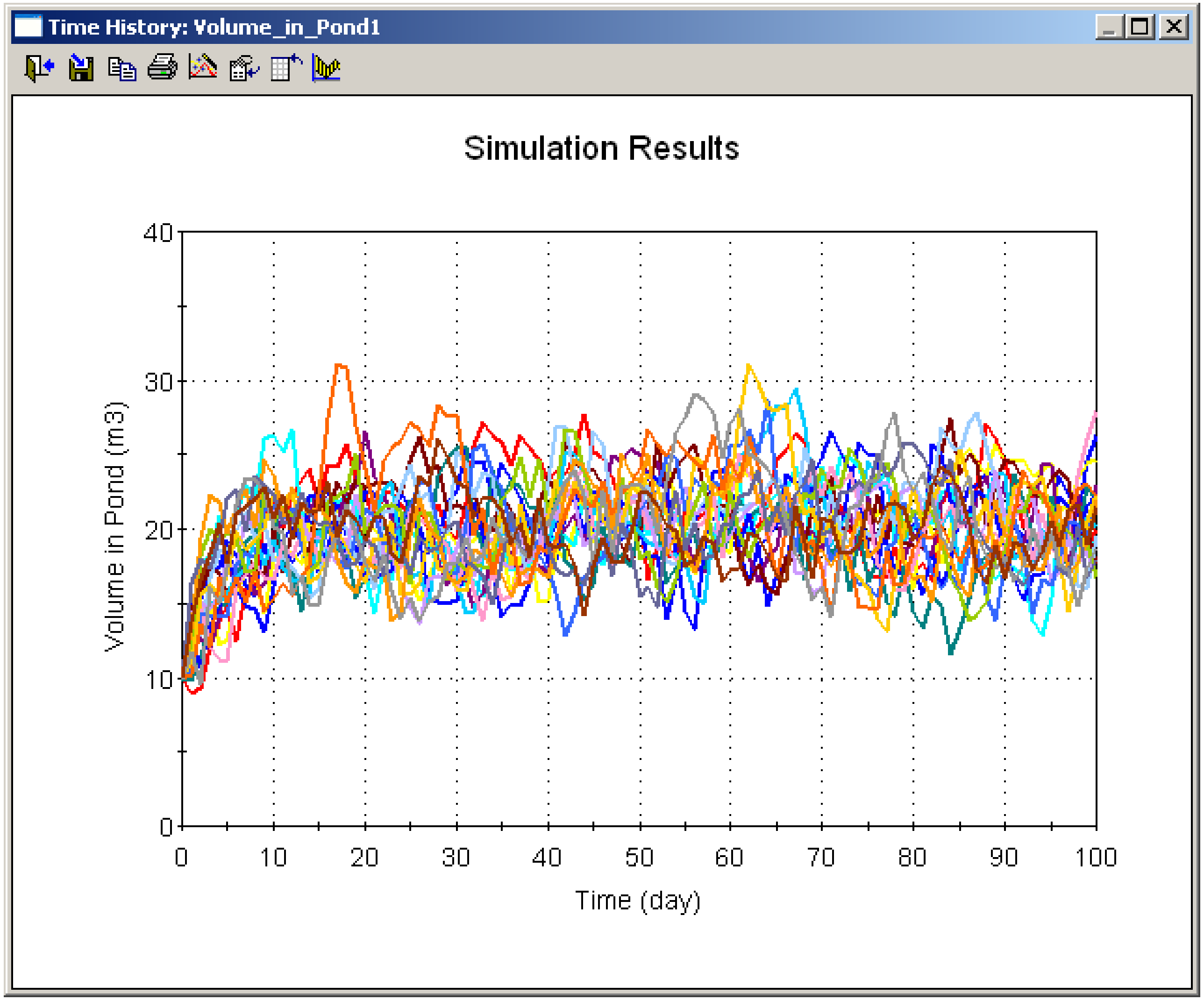
Fund



Reservoir

$$\text{Current Value} = \text{Initial Value} + \int \text{Rate of Change}$$

Probabilistic Simulation



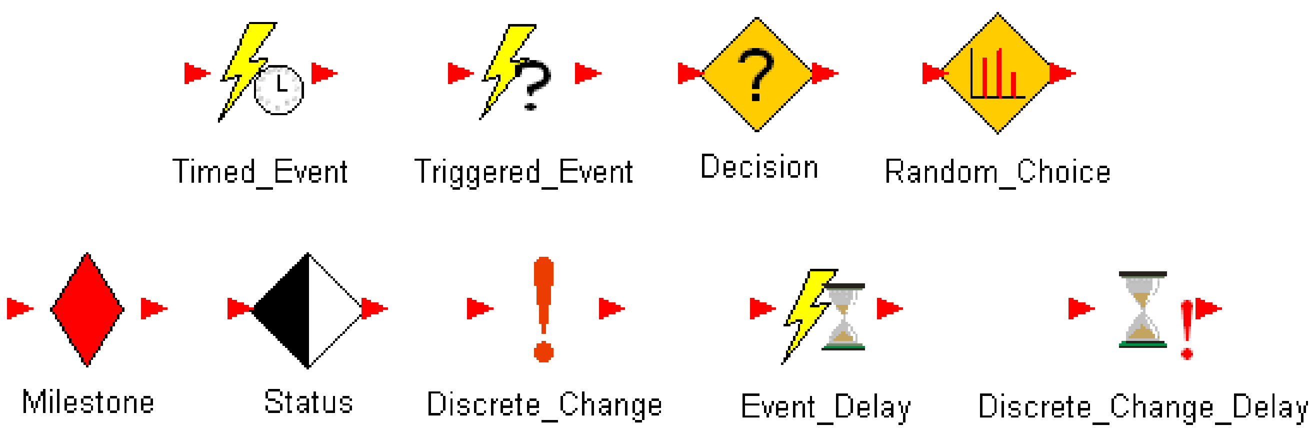
To do predictive modeling, we need to represent:

- Uncertain parameters
- Stochastic variables

Use Monte Carlo simulation

Simulating the Occurrence and Consequences of Discrete Events

- **In some kinds of systems, processes occur which are discrete, rather than continuous**
  - accidents, system failures, financial transactions
- **GoldSim provides a powerful collection of elements for representing the occurrence and consequences of discrete events.**

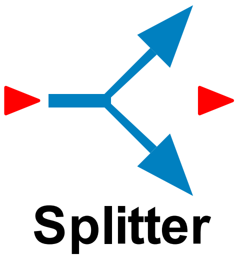


Creating Subsystems

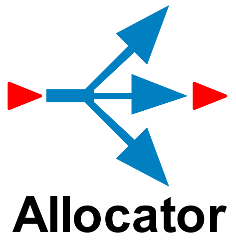
- Complex models may have many thousands of elements. In order to organize and view such a model, it is useful (in fact, essential) to group the elements into subsystems.
- Subsystems are created by placing elements into Containers. A container is analogous to a “folder” or a “box”.
- Containers can be placed into other containers, and any level of containment can be specified.
- Containers can be locked.
- Containers can be easily reused.
- Containers have many other features for advanced users.

Some Additional Elements Useful for Material Management Simulations

- **The Splitter element splits an incoming flow into multiple outputs based on specified fractions.**

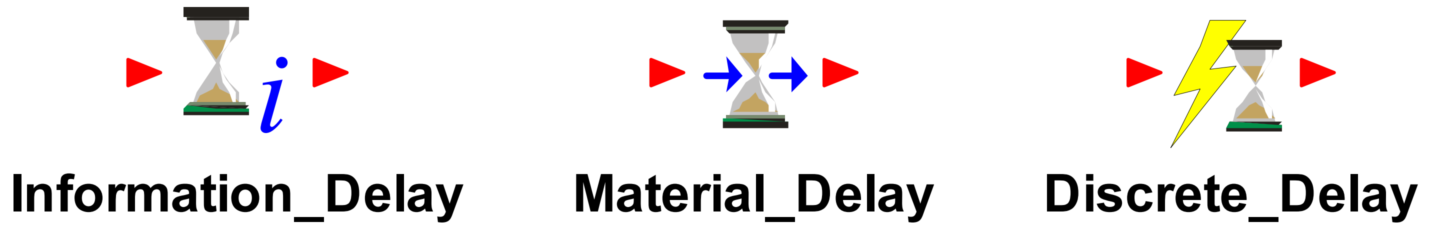


- **The Allocator element allocates an incoming flow into multiple outputs given specified demands and priorities**



Delay Elements

- The output of a Delay element lags its input.
- Delay elements can be used to model processes such as the movement of water through soil, the movement of parts on a conveyor, and the transfer of information from one person to another.



Conditions and Triggers

- **Some elements can be defined as conditions (True/False) rather than values**
- **Use of conditions can make the model logic much clearer**

- **"Events" are of two types:**
  - "Scheduled" or "Timed" events
    - **e.g., once a week regularly, once a year randomly**
  - Conditional events
    - **e.g., whenever X becomes greater than Y**

Outline

- What is GoldSim and where did it come from?
- A summary of the major differences between GoldSim and traditional SD codes
- Basic GoldSim Features
- **Overview of Advanced GoldSim Features**
- Overview of GoldSim Extension Modules
- Can GoldSim complement traditional SD codes?
- Questions and Discussion

Advanced Properties of Containers

- **You can localize Containers in your model so that variable names can be repeated without causing conflicts**
- **You can clone a Container (or individual elements in a Container). Clones all behave identically**
  - Allows you to rapidly build and maintain models consisting of parallel systems that are governed by the same equations but require different inputs
- **You can make Containers conditional. This allows you to make a Container and all of its contents inactive unless specific events occur and/or conditions are met (useful for simulating tasks and projects)**
- **Can have their own timestep**
- **Can be iterative (looping)**

Additional Features for Managing Complexity in GoldSim

- **Powerful search capabilities**
  - Find an element
  - who affects who?

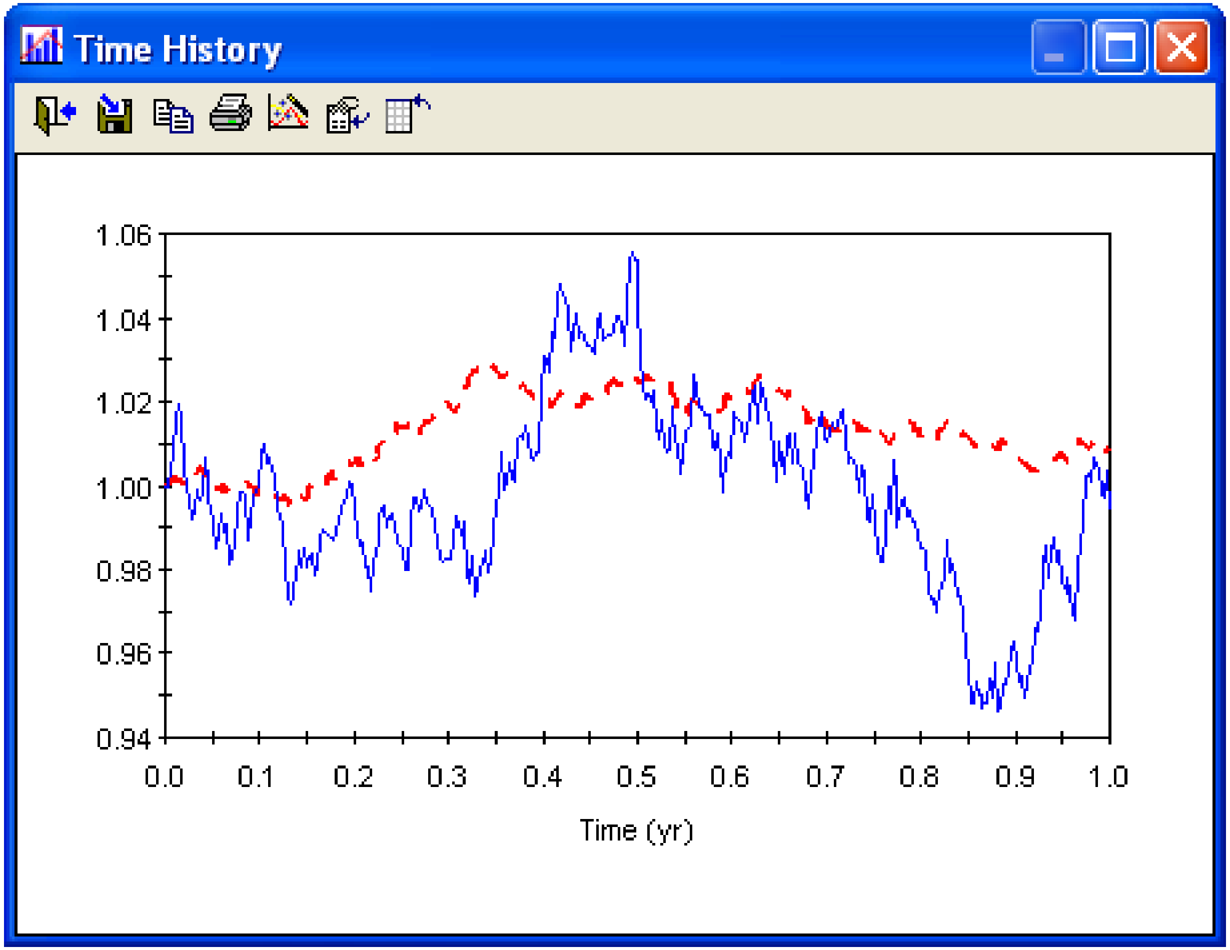
- **The ability to record versions (revisions) of a particular model file, so that you can identify the differences between the various versions of the file as the model is iteratively modified.**

Convolution Element

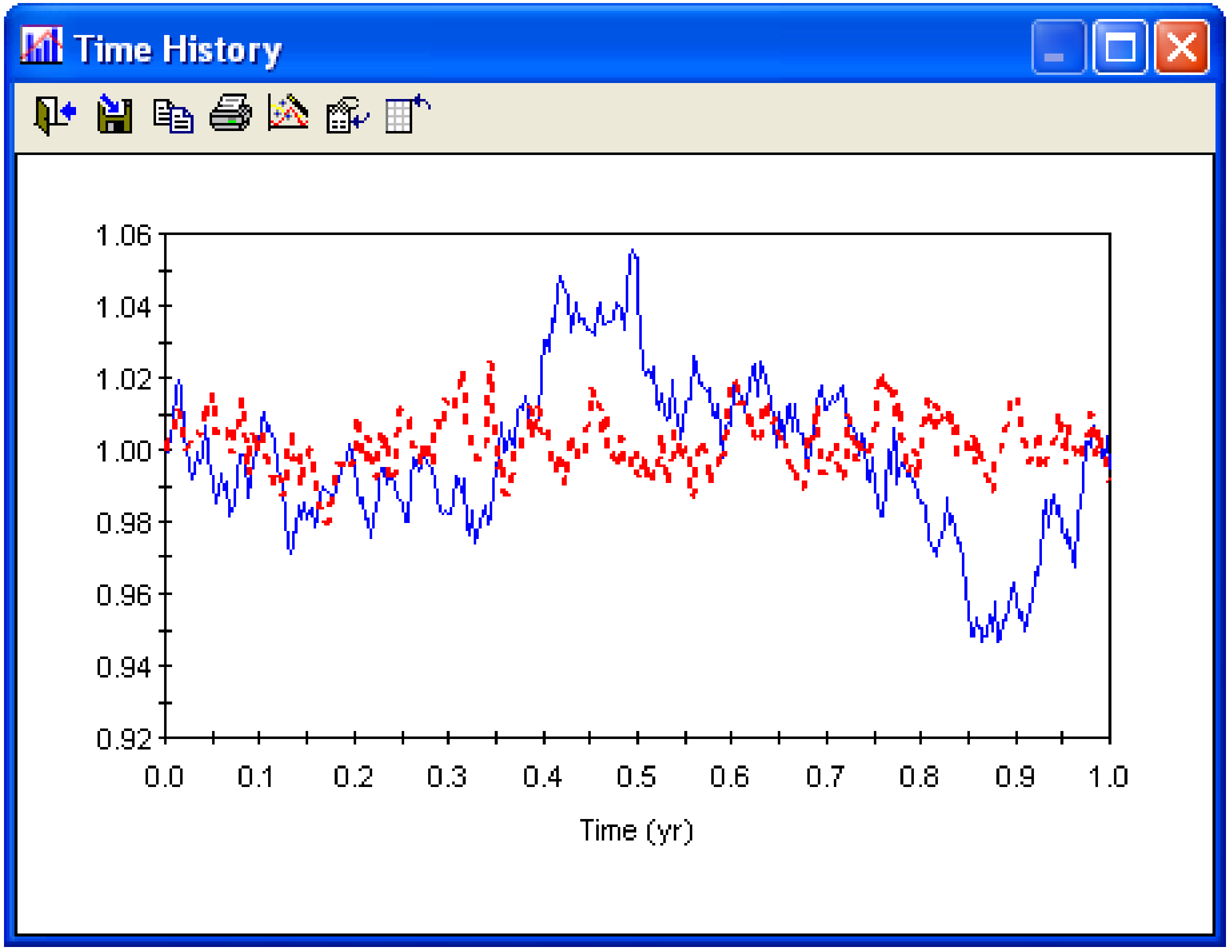
- **Solves convolution integrals**
- **Inputs:**
  - An input function (which can be time-variable)
  - A transfer function (impulse response function)
- **Effectively allows you to create custom transfer or delay functions**

History Generator Element

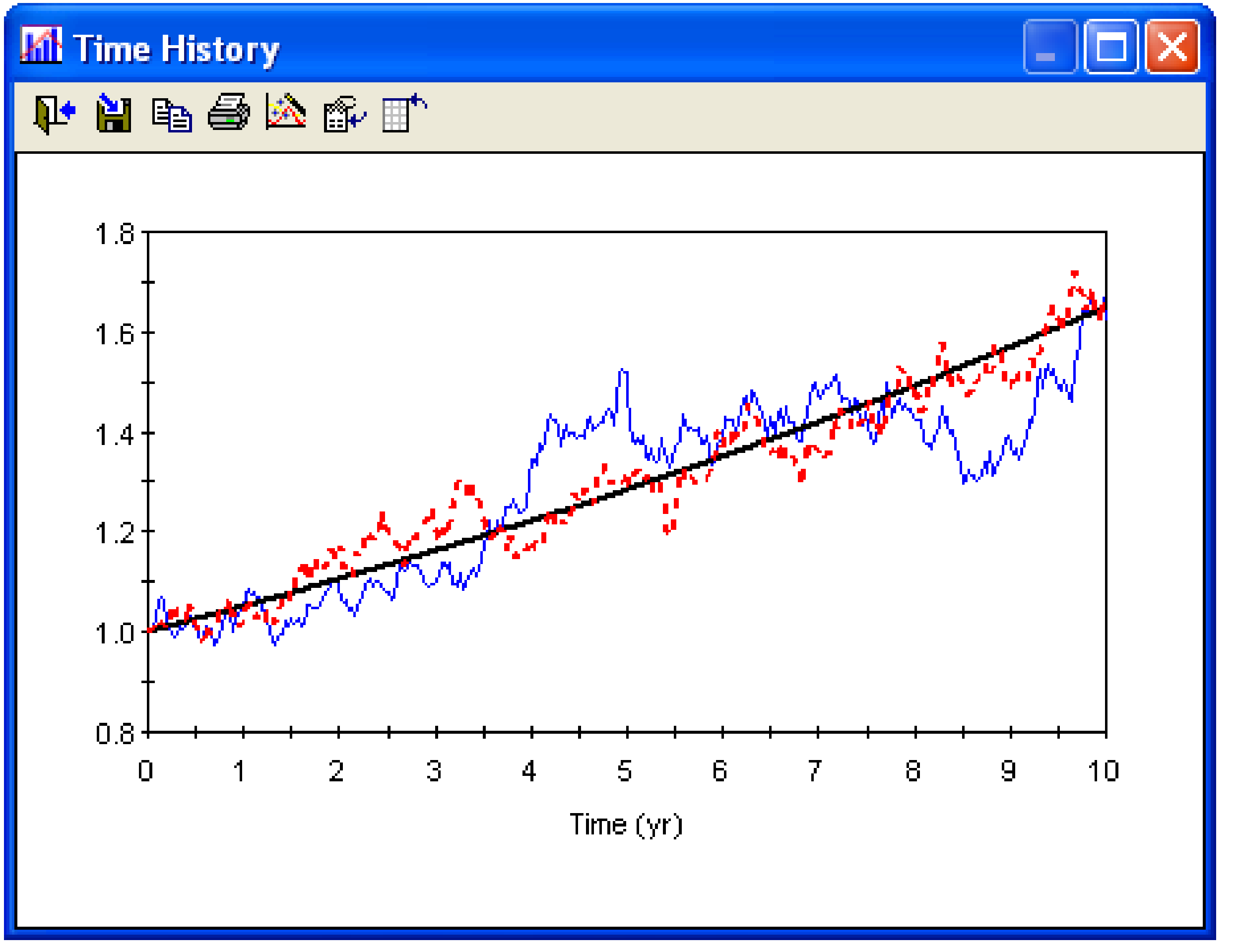
- **A powerful element that generates stochastic time series given some statistical inputs (e.g., growth rate, volatility)**
  - Can simulate a variety of history types
  - Can simulate geometric growth, random walks, and random movement around a target
  - Can simulate correlated arrays of stochastic variables (by specifying a correlation matrix)



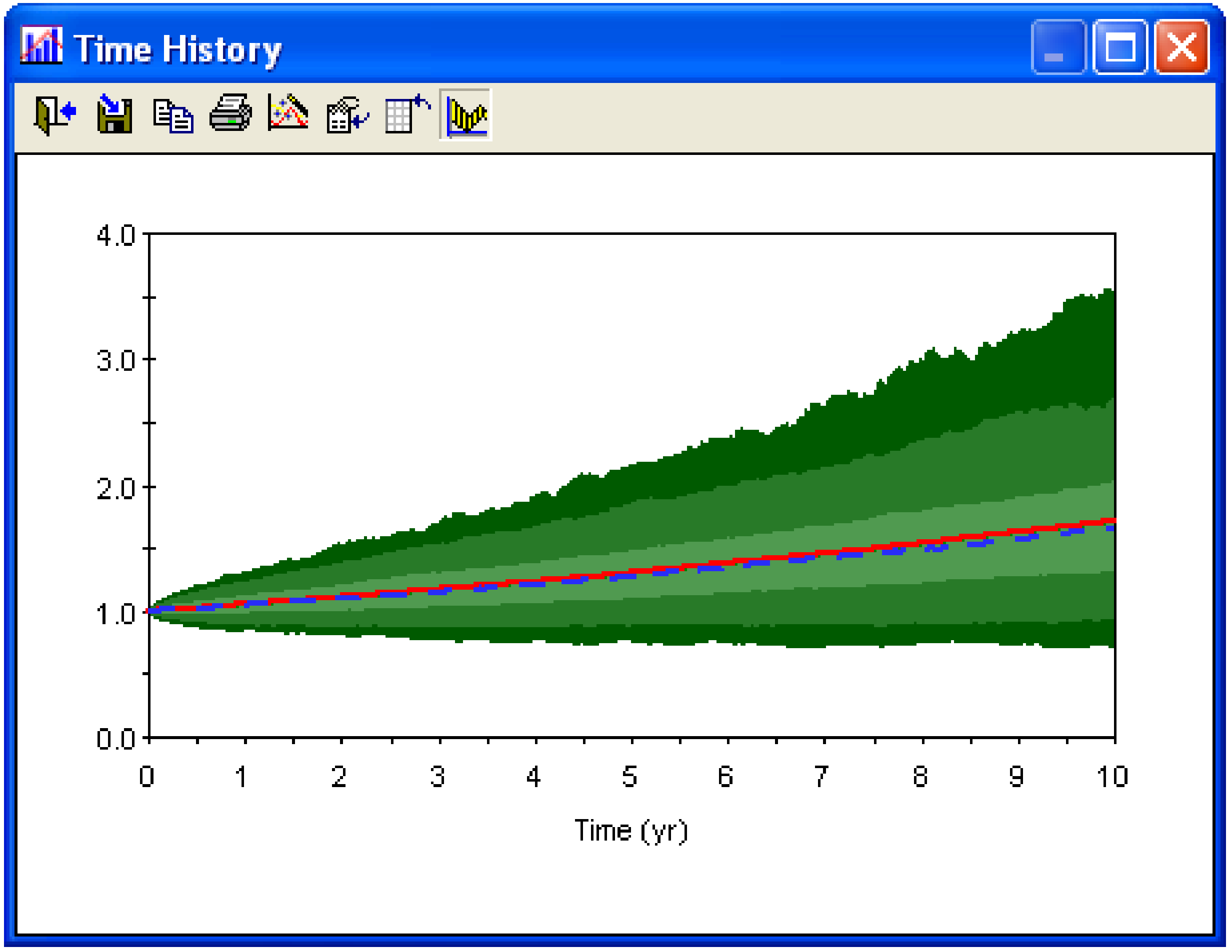
Random Walks: High and Low Volatility



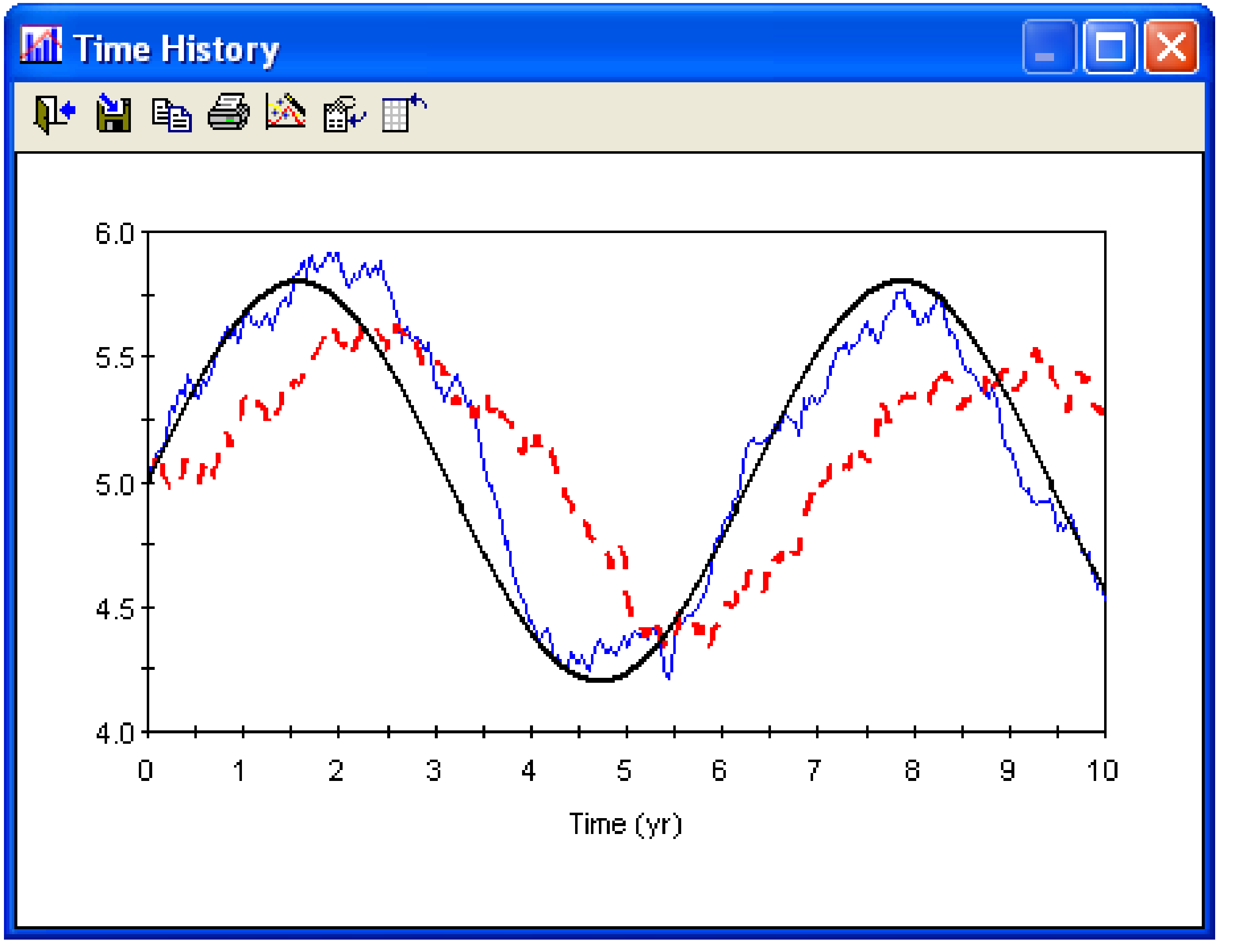
Random Walks: With and Without Reversion to a Constant Target



Geometric Growth: High and Low Volatility



Volatile Geometric Growth: Multiple Realizations



Random Walk That Tracks a Dynamic Target: With and Without Time Lag

Working with Arrays

GoldSim provides almost 40 functions for manipulating arrays

Vector Constructor	vector(,)
Vector Item Access	GetItem(v,r)
Vector Row Access	GetRow(v,r)
Vector Length	GetRowCount()
Vector Sum	sumv()
Vector Product	prodv()
Vector Minimum	minv()
Vector Maximum	maxv()
Vector Mean	meanv()
Vector Standard Deviation	sdv()
Vector Minimum Ordinal	rowmin()
Vector Maximum Ordinal	rowmax()
Vector Dot Product	dot(,)
V * VT = M	vvmatrix(,)
Matrix Constructor	matrix(,)
Matrix Item Access	GetItem(m,r,c)
Matrix Row Access	GetRow(m,r)
Matrix Column Access	GetColumn(m,c)
Matrix Length	GetRowCount()
Matrix Width	GetColumnCount()
Matrix: sum of items in each row	sumr()
Matrix: product of items in each row	prodr()
Matrix: mean of items in each row	meanr()
Matrix: std. dev. of items in each row	sdr()
Matrix: smallest item in each row	minr()
Matrix: greatest item in each row	maxr()
Matrix: sum of items in each column	sumc()
Matrix: product of items in each column	prodc()
Matrix: mean of items in each column	meanc()
Matrix: std. dev. of items in each column	sdc()
Matrix: smallest item in each column	minc()
Matrix: greatest item in each column	maxc()
Matrix Transpose	trans()
Matrix Inverse	inv()
Array Minimum (term-by-term)	min(,)
Array Maximum (term-by-term)	max(,)
Array (Linear Algebra) Multiplication	mult(,)

Linking Spreadsheets and Databases to GoldSim

- **You can dynamically link GoldSim to a spreadsheet**
  - Import time series, lookup tables, or scalar or array data from a spreadsheet
  - Export time series and other results to spreadsheet
  - Spreadsheet can act as a sub-routine
    - **Can link to VBA applications**
- **You can import information from any ODBC compliant database directly into GoldSim prior to a simulation**
  - GoldSim records when the database was uploaded
  - Facilitates QA/QC of model data

GoldSim TECHNOLOGY GROUP

Dynamically Linking External Programs to GoldSim

- **If GoldSim's built-in elements are not capable of adequately representing a particular aspect of your model, you can dynamically link an external program to GoldSim.**
- **It behaves identically to an Expression element, but instead of using an equation, GoldSim dynamically calls and runs the external program.**
- **This allows complex external programs to be linked directly into the probabilistic, graphical GoldSim framework.**

Sensitivity Analysis

- **Computes statistical measures**
  - Based on multiple Monte Carlo simulations (all variables changed simultaneously)
- **Graphical sensitivity analyses (tornado charts, x-y charts)**
  - Variables of interest are changed while holding all other variables constant