Introduction to Modeling with Simile

This is an introduction to using Simile for building and analysing a model. Simile is a visual modeling tool. It allows you to draw a conceptual model diagram, and if you have quantitative information you can then add this into the conceptual model to produce a quantitative model. Simile also allows the dynamics of a quantitative model to be simulated and the results displayed in graphs and tables.

Extra documentation on Simile can be found in the Simile's help files and on the Simile web site http://www.simulistics.com/documents. Links to the relevant web help pages can be found throughout this guide.

The best way to learn about modelling is by practice. So we will look at a simple model which is the Hunter-Gather example from the book "Ecological Modelling: A Common-Sense Approach to Theory and Practice" (Grant and Swannack 2007). We will see how to build and analyse the model with Simile, and then there are some problems which you should try and answer using Simile. At the end I present some additional functionalities of Simile which may be useful for other modelling problems.

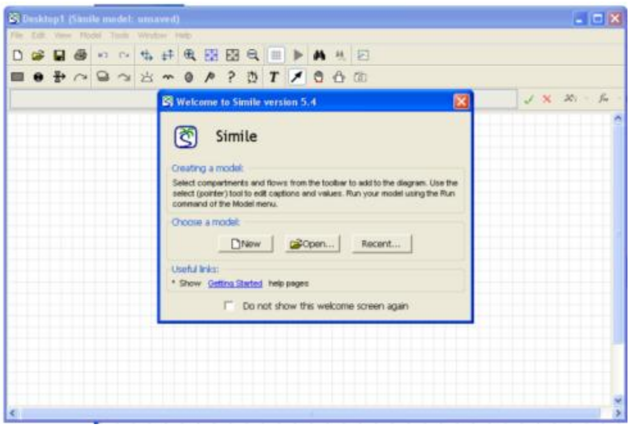
Starting Simile

You start Simile by double-clicking on the desktop icon created during installation, or by clicking on the Simile icon in the Start Menu.

Within a few seconds, the main window will be displayed.

The Main Window

When Simile starts, you see a single window. If you see a welcome page (shown opposite), click on the "New" button


Desktop1 (Simile model: unsaved)
File Edit View Model Tools Window Help
Welcome to Simile version 5.4
Simile
New
Open...
Recent...

Creating a model:

Select compartments and flows from the toolbar to add to the diagram. Use the select (pointer) tool to edit captions and values. Run your model using the Run command of the Model menu.

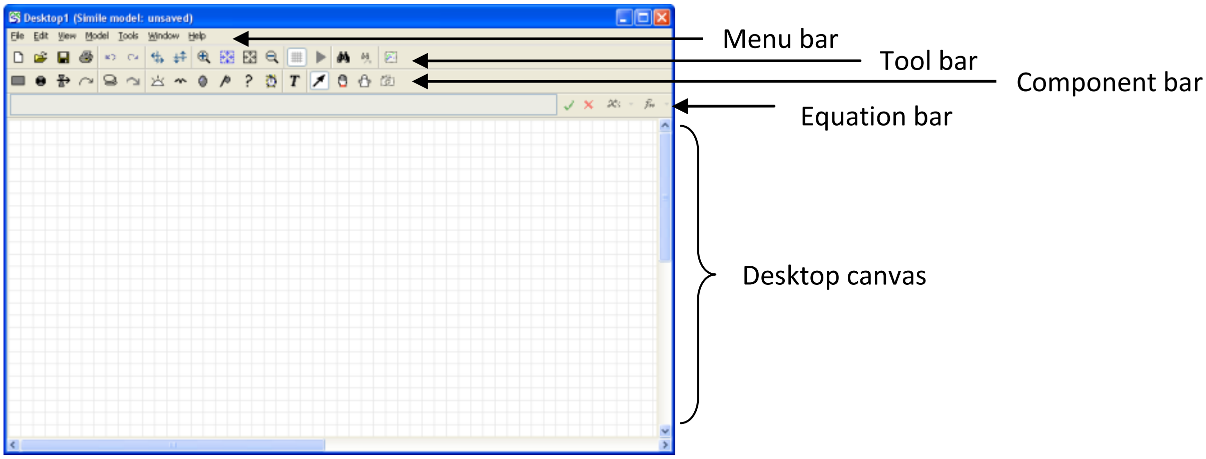
Choose a model:

Useful links:

* Show Getting Started help pages

□ Do not show this welcome screen again

You should now see the Main Simile window, like this…


Desktop1 (Simile model: unsaved)
File Edit View Model Tools Window Help
Menu bar
Tool bar
Component bar
Equation bar
Desktop canvas

The main window contains:

- **The Menu Bar**, containing a standard File menu, some menus specific to Simile, and a Help menu. Hold the mouse over the title for each menu to see the options it contains.

- **The Toolbar**, with standard tools (e.g. for opening and closing files), plus some specific to Simile.

- **The Component Bar**, containing buttons corresponding to each type of model-diagram element: you will use these for building up your model diagram. The right-hand part contains buttons for editing the model diagram. Hold the mouse over each button to get a brief description of its function.

- **The Equation Bar**, for entering equations into the model. At the right-hand end of the equation bar are the tools to accept or reject changes to equations, and to help build equations.

- **The Desktop Canvas**, where you will draw your model diagram. The grid can help you arrange your diagram neatly.

The ecological system: The question and the data

The ecological system we are going to consider is the first example in the book by Grant and Swannack (Grant and Swannack 2007). It is called the Hunter-Gatherer Model.

The Question… We want to know if recent deforestation will have a serious impact on a group of hunter-gathers that live in the forest during the winter. The group usually hunts in the savanna until the end of October and then moves to the forest to spend the winter. Can the group of hunter gathers collect enough food for the winter despite the recent deforestation, or should aid strategies be put in place?

The Data… Severe winter weather, which prevents hunting and gathering activities, typically arrives by the end of November. Thus the group has only one month to harvest the food items they need to survive the winter. There are usually 100 food items in the forest at the beginning of November. For practical purposes, food items are a nonrenewable resource during November since plants no longer produce fruits and both birth and death rates of animal populations are essentially zero. Historically, the group has had no trouble harvesting the 75 food items they need to survive the winter, although the group has told us the rate at which they can harvest food items decreases noticeably as abundance of food items decreases. The group estimates they can harvest about 10% of the available food items per day. As a result of the recent deforestation, we estimate only about 80 food items will remain in the forest when the group arrives this autumn. Although there still appears to be enough food available, we wonder if the group will have enough time to harvest all they need before the severe weather arrives.

The steps taken to model the problem

Two important steps have already been completed.

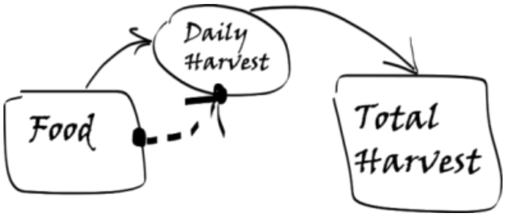
1. We have a good idea of the **question** we want to ask.
2. We have a good idea of the **data** available to us.

We will continue by

3. Developing a **conceptual model** of the system which is suited to the data and the question
4. Developing a **quantitative model** from the conceptual model by explicitly including the data
5. **Evaluating and analyzing** the model's results
6. Posing some problems for you to answer about the model system
7. Looking at some additional Simile skills

Drawing a conceptual model in Simile

We begin by drawing a *conceptual model diagram* for the Hunter-Gatherer model. We are interested in the **Total Harvest** across one season, and this harvest comes from the **Food** in the forest. So there are two components (or state variables) to this system. The **Food** and the **Total Harvest** are linked by the **daily harvest** that removes food from the forest and increases the total harvest. So I've drawn all this in a sketch of the system below.



A note on colours in Simile

- When you select a compartment it will turn blue (a new compartment will automatically be selected).
- When you select a flow it will turn green (a new flow will automatically be selected).
- A compartment or flow will be red if it is not selected *and* it needs some extra information (like a value or equation).
- When a compartment or flow is complete it will turn **black**.
- When a model diagram is all black the model can be run.

There are two state variables: **Food** and **Total Harvest.**
The solid arrows show the "flow" from forest food into total harvest.

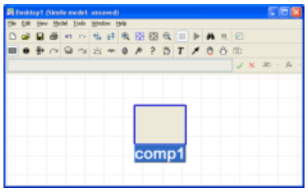
I've added in a dashed arrow, because we were told that the food in the forest affects the daily harvest. The daily harvest isn't constant. Instead, as the food in the forest decreases so does the daily harvest. This dashed arrow reminds me that the daily harvest is influenced by the food in the forest.

I now want to draw this conceptual model in Simile.

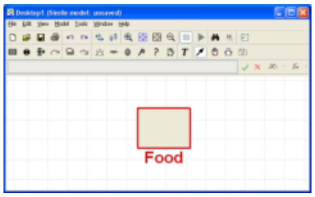
Adding a compartment to a model ■

- Click on the ■ compartment symbol in the toolbar.
- Move the mouse to the middle of the desktop window, and click again.

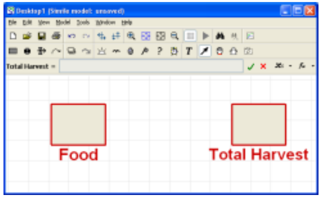
You should see a box labeled **comp1**.



Type in the label **Food**. This box will be the food available in the forest.



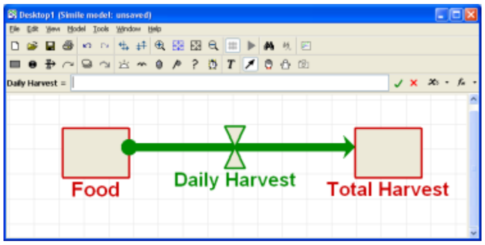
Now repeat this step to create another compartment, and label it **Total Harvest.**



Adding a flow to a model

We want to add in the flow from the Food into the Total Harvest.

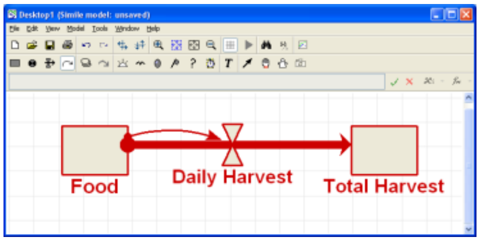
- Click on the flow symbol in the toolbar.
- Move the mouse to the compartment labeled **Food**. Hold the mouse button down and drag the mouse into the centre of the compartment labeled **Total Harvest**. Release the mouse button.
- Type in the label **Daily Harvest**.



Adding in an influence to a model

The food in the forest influences the daily harvest (the dashed arrow in our sketch). In Simile this is represented by an *influence arrow* .

- Click on the influence arrow button in the toolbar.
- Move the mouse to the middle of the compartment labeled **Food**, and drag an influence arrow to the flow labeled **Daily Harvest**.
- Release the mouse button when the flow has turned green.



Adding a variable to a model

The final component is a variable. A variable in Simile is just a number, but we can give it a name. This will be useful when calculating output results, or defining initial parameter values. We do not need this at the moment, but we will later on. Here is how a variable is added.

- Click on the variable symbol in the toolbar.
- Move the mouse to the blank area of the model canvas and click once.
- Type in the label for the variable.

Re-arrange the model diagram

- Click on the pointer button in the toolbar.
- Move the mouse to the middle of the compartment labeled **Food**.
- Drag the mouse, and watch the compartment and its associated arrows re-arrange themselves.
- You can move many of the components in Simile, including the flows, the influence arrows, etc
- The labels of a component can also be moved.

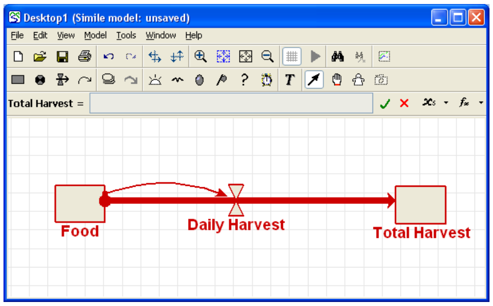
Re-sizing the model diagram

- You can zoom the model diagram to fit the current window by clicking on the button
- You can zoom in to a diagram by clicking on the button
- You can zoom out of a diagram by clicking on the button

Removing model components

- Click on the pointer button in the toolbar.
- Click on the model component you want to remove (it will turn blue).
- Select the **Edit/Delete** option in the menu bar and the selected component will be deleted.

We now have the conceptual model drawn in Simile, which looks something like this...


Desktop1 (Simile model: unsaved)
File Edit View Model Tools Window Help
Total Harvest =
Food
Daily Harvest
Total Harvest

Save the model by selecting **File/Save** in the menu bar. Call the model **forest.sml**

Copying a picture of the conceptual model

The diagram of the model can be copied and pasted into another document by selecting **Edit/Select All** in the menu bar, followed by **Edit/Copy**.

Additional model components

Simile has many other models components, which are described in the help pages. We will discuss some other components later on.

Making a quantitative model in Simile

Up to now we have a conceptual model which describes the flows and influences of the components upon one another. But we have not included any numerical data into the model. We know the amount of food that is initially in the forest, and the rate at which it can be harvested, so we want to add these in and make the model **quantitative**.

The data we have are: Initially there are 100 items of forest food, and nothing has been harvested. Every day 10% of the food in the forest can be harvested. Harvesting occurs during November.

We are told that the daily harvest is 10% of the food in the forest, so on Day 1 the harvest is

$$Day\ 1\ DailyHarvest = 0.1\ \times Initial\ Food$$

The daily harvest on day 1 is calculated from the food available at the start of day. This is why we included an influence arrow in the conceptual model from **Food** to **DailyHarvest**.

Every day the food in the forest is reduced due to harvesting,

$$Day\ 1\ Food = Initial\ Food - Day\ 1\ DailyHarvest$$

and the total harvest is increased

$$Day\ 1\ Total\ Harvest = Initial\ Total\ Harvest + Day\ 1\ DailyHarvest$$

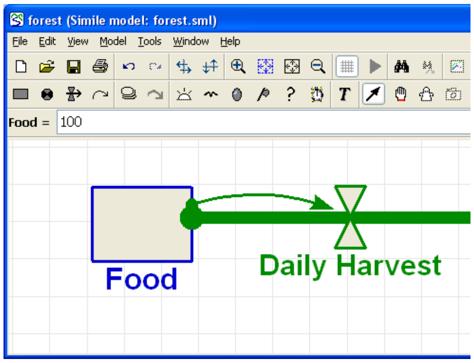
The value of **DailyHarvest** is the rate (per day) at which the forest food is decreasing, and the rate (per day) at which the total harvest is increasing. Simile can use these equations to predict the future Total Harvest and the Food at the end of each day. To do this we have to input this *quantitative information* into the model.

Prepare to set component properties

Click on the ↗ pointer button in the toolbar.

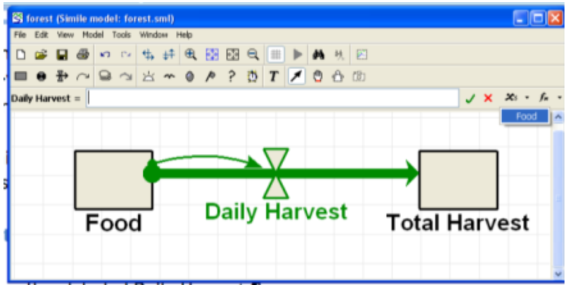
Assign an initial value for a compartment

- Click on the compartment labeled Food.
- Enter the value 100 into the equation bar.
- Click on the ✓ green tick mark or press the Return key.
- Repeat this for the compartment labeled Total Harvest and set its value to 0 in the equation bar.
- As the initial values of the compartments are set they turn black, showing that all the information for this compartment has been specified.



Food = 100

Add the harvest rate equation


forest (Simile model: forest.sml)
File Edit View Model Tools Window Help
Daily Harvest =
Food
Food
Daily Harvest
Total Harvest

- Click on the flow labeled Daily Harvest.
- The variable Food is listed as an influence upon Daily Harvest. These influences are visible by clicking on the x_s button. Selecting an influence from the drop-down list, places the text in the equation bar.
- Enter the expression **0.1*Food** into the equation bar.
- Click on the ✓ green tick mark.
- A full discussion of defining equations in Simile can be found on the Simile help pages.

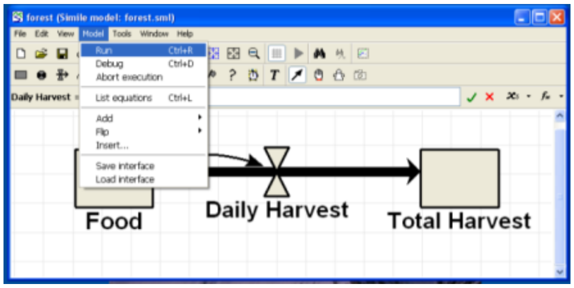
Notice that every component of the model diagram is now black, rather than red as it was before. This indicates that every component has been mathematically specified, and so the model is ready for running. Simile has enough information to work out the flows, and thus to update the amount of food in the forest and the total harvest forward through time.

Seeing the equations and data in a model

You can see all the equations in a model by selecting **Model/List equations** from the menu bar, or by typing Ctrl+L. This will open a window with all the components in the model, their equations and their values. You can compare the equations with those on the previous page.

Running the model

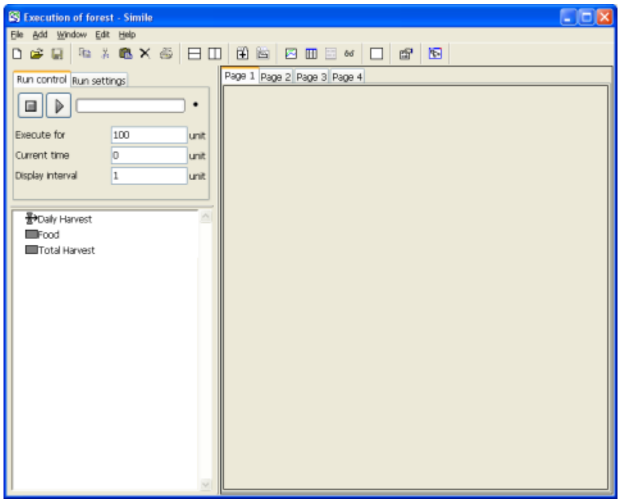
Now all components have been fully set we can run the model to predict the effect of harvesting the forest food.



First we compile the model

- Open the **Model** menu
- Select the **Run** item.

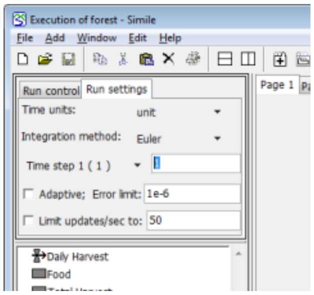
Simile creates a new window: *the execution window*. This contains the controls for running the model; a list of the variables in the model; and an area where any of a variety of tools for showing model results can be displayed.


Execution of forest - Simile
File Add Window Edit Help
Run control Run settings
Page 1 Page 2 Page 3 Page 4
Daily Harvest
Food
Total Harvest

Execute for 100 unit
Current time 0 unit
Display interval 1 unit

Change the time step

Simile supplies a default time step of 0.1 when you first run a model. Usually a time step shorter than the time unit results in greater mathematical accuracy. However, here we are interested in daily time steps, so we want a time-step of 1.

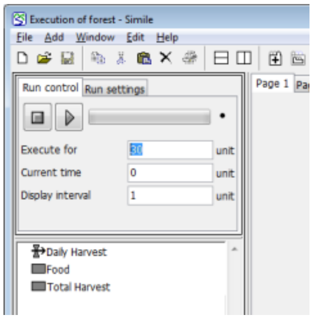


Time units: unit
Integration method: Euler
Time step 1 (1) 1
□ Adaptive; Error limit: 1e-6
□ Limit updates/sec to: 50

- Select the **Run Settings** tab
- Change the value for **Time step #1** from 0.1 to 1.

Change the duration of the simulation run

- Change the value for **Execute for** from 100 to 30.



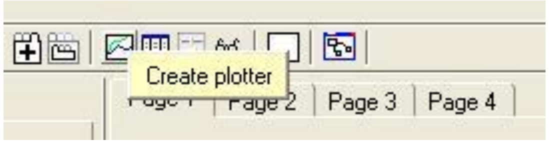
Execute for 30 unit
Current time 0 unit
Display interval 1 unit

This is because we want to run the model for November (30 days).

Select the graph-plot display

Simile has several graphical tools to display results A list of the graphics can be found by clicking on the “Spectacles” button in the toolbar.

- Click on the "Plotter" display tool on the toolbar; or select it from the list in the Add menu.



Choose the variable you wish to have displayed

Note the graph window that appears. This is initially scaled with default values on both axes, but will rescale itself as needed. You can re-size the panel containing the graph by dragging on the little boxes on the horizontal and vertical panel separators.

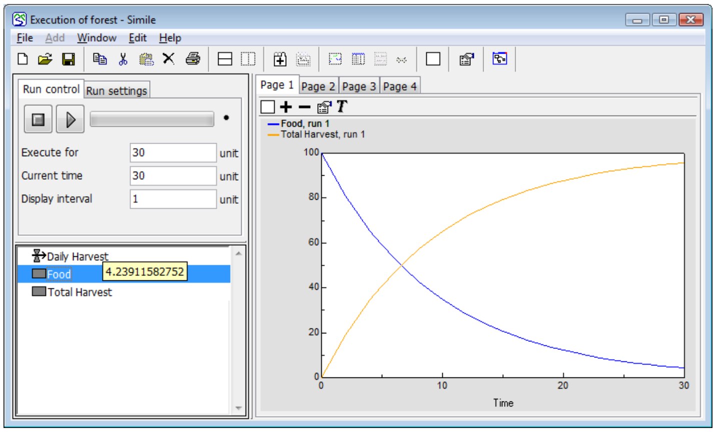
- Click on the + "Add a variable" button on the plotter toolbar
- Click on the compartment labelled **Food** in the list of model variables.
- Click on the + "Add a variable" button on the plotter toolbar
- Click on the compartment labelled **Total Harvest** in the list of model variables.

Start the simulation

- ▷ Click on the "Play" button in the Run control dialogue window.

Note the line that appears on the graph.

By placing the mouse over the variable **Food** in the bottom left window shows the value of food after 30 days. The same can be done for Total Harvest and Daily Harvest.



Execute for 30 unit
Current time 30 unit
Display interval 1 unit

Continue the simulation

- ▷ Click on the "Play" button again.

The simulation carries on for another 30 days.

Resetting the simulation

The simulation can be reset to time=0, and run again.

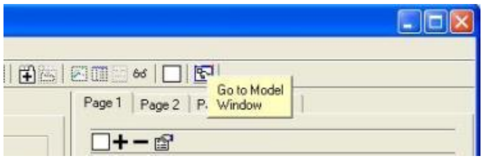
- The ■ stop button will stop the simulation and set the current time to zero
- ▷ Click on the "Play" button to run the model again

Resetting the graphics display

All the graphics can be cleared by clicking the □ clear button in the toolbar. Individual graphics can be cleared by clicking the same icon in each graphics panel.

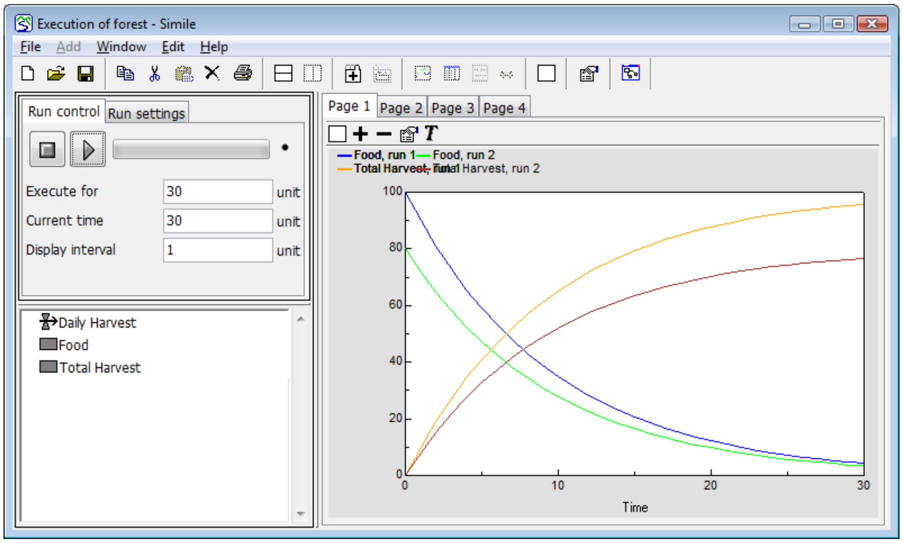
Changing parameter values and rerunning the model

We now want to run a simulation where there is less initial food in the forest due to deforestation.



- Click on the "Go to Model Window" toolbar button. Or select the model window directly.
- Now select the **Food** compartment and set its initial value to 80
- Recompile the model, using **Model/Run** in the menu bar.

- Rerun the model for the first 30 days. The new simulation results are shown as run 2.
- Is the harvest at the end of November greater than the 75 units necessary for the hunter-gather's survival over the winter?
- At what initial food abundance will the hunter-gathers find it difficult to survive the winter?


Execution of forest - Simile

Run control | Run settings

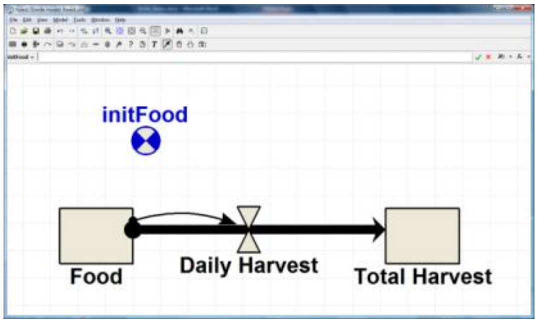
Execute for 30 unit

Current time 30 unit

Display interval 1 unit

Saving the model graphics

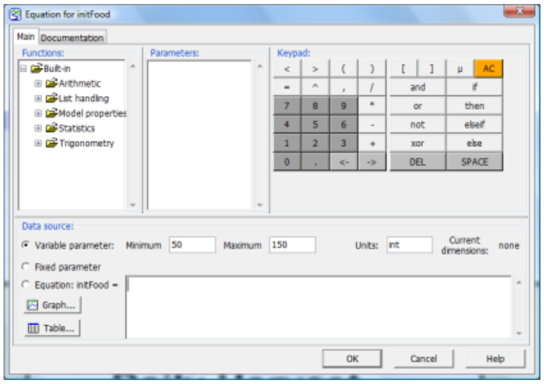
The graph of simulation results can be copied and inserted into a Word document. Select the graph and then select **Edit/Copy** in the menu bar to copy the figure onto the clipboard. The figure can then be pasted into a Word document.



Defining a variable parameter with the equation dialogue window

We want to change the Initial forest food without recompiling the model every time. To do this we will give these two quantities their own variable names.

- Return to the Model Window.
- Create a new variable for the Initial food in the forest and label it **initFood**
- Next we will define this variable as having a range of possible values rather than one fixed value. We will be able to set the value when the model is running.
- Double-click on the **initFood** variable opens up an *equation dialogue window*. This window contains the information that is in the equation bar, and some additional options. It can be used to define many kinds of variables.
- Under data source click the "Variable parameter" option and then type in the minimum and maximum values of the efficiency (we'll take the minimum to be 50 and the maximum 150).


Equation for initFood

Data source:

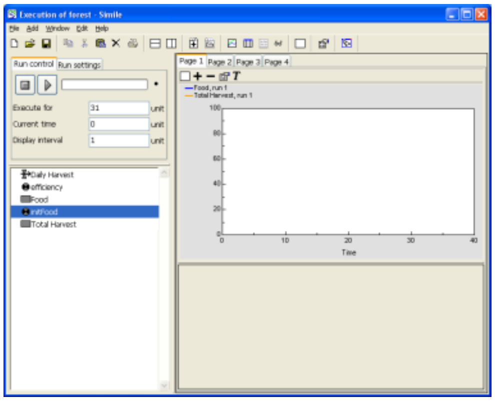
● Variable parameter: Minimum 50 Maximum 150 Units: int Current dimensions: none

○ Fixed parameter

○ Equation: initFood =

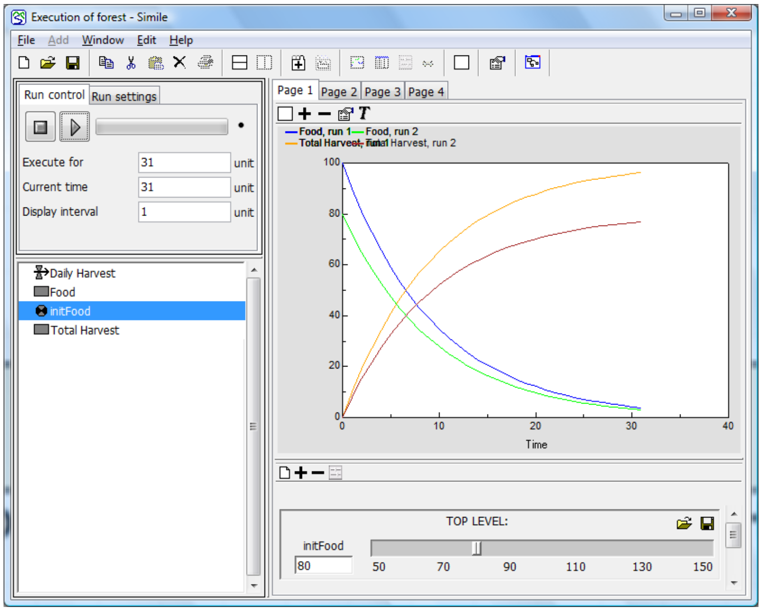
- In the window at the bottom type in 100 as a default value for the initial food, and the press the OK button.
- In the model, initFood now has a line through it to show that it is defined with a range of values.
- Draw influence arrows from initFood to the Food compartment.
- Click on the Food compartment and in the equation bar set its initial value to be **initFood** (as shown).
- The initial food is now defined by a new variable (**initFood**) that has a well defined range.
- The model should be ready to compile using **Model/Run** in the menu bar.

This reopens the *execution window*, with the old simulation results. Now we want to be able to change the values of initFood.



Execute for 31 unit
Current time 0 unit
Display interval 1 unit

- Split the graph window into two by clicking on the "Split pane horizontally" button ⊟ in the tool bar
- Click on the "Create input sliders" button ▤ in the toolbar
- Add the variables initFood and efficiency by clicking on the + "Add a variable" button, and then the variable
- The sliders can be used to set the values for the two variables.
- Clear the graph by selecting the □ "Clear" button
- Reset the simulation with the ■ "Stop" button
- Then the model can be run by clicking on the ▷ "Play" button.
- The sliders can be used to change parameter values without recompiling. The ■ ”Stop” button must be pressed after the slider has been moved to select the initial value.



Execute for 31 unit
Current time 31 unit
Display interval 1 unit
initFood 80

Here are results when initial food is 100, and when initial food is reset to 80.

Problems to answer about the model system

Here are some problems which you can use our Simile modelto answer.

Produce a document which describes the model, the important state variables, and the relationship between variables. Include a diagram with the final structure of the conceptual model.

How long does it take to harvest 75 food items *before* deforestation took place (i.e. when there are 100 food items at the start of November)?

How long does it take to gather 75 food items now that deforestation has happened (i.e. when there are 80 food items at the start of November)?

In your document, explain if the deforestation will threaten the survival of the hunter-gather group (use graphics from the model if you feel it will clarify your answer). Explain the evidence behind your answer, and include a graph of the **daily harvest** for each day in November when initial food abundance is 80. Explain the graph and the important points of the graph for the problem of the hunter-gatherer survival.

What is the daily harvest after day 1 and day 20 of harvesting when initial food abundance is 80?

How low can the initial food abundance in the forest drop, and still allow the hunter-gatherers to collect 75 food items before the end of November?

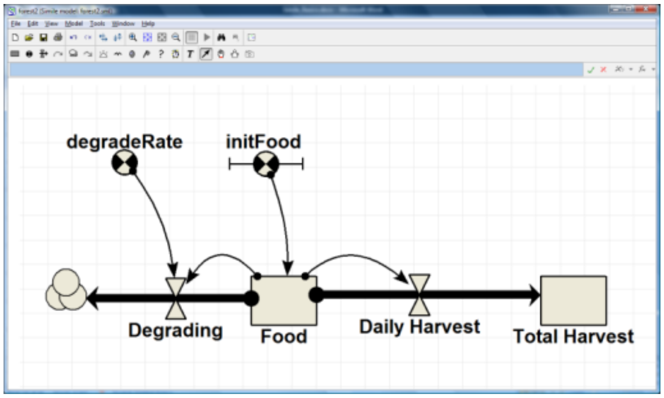
If the harvesting efficiency of the group is decreased, what is the critical threshold of efficiency below which survival over the winter is uncertain? (state the initial food abundance along with the critical foraging efficiency)

Additional Simile Skills

Returning to the Model and Adding a New Compartment

We have an additional piece of information that we want to add to the model. We find out that food in the forest degrades. Every day 15% of the food items degrade and are not fit for harvesting. These items are effectively removed from the harvestable food in the forest. We want to add this information into the model.

- Return to the Model Window by either clicking on the window, or by clicking the button in the execution window.
- Back in the model window we want to add a flow from **Food** which represents its degradation before it can be harvested.
- Click on the flow symbol in the toolbar.
- Move the mouse to the compartment labeled **Food**. Hold the mouse button down and drag the mouse into an empty part of the canvas. Release the mouse button. The flow ends in a "cloud" symbol, meaning that it leaves the system of our model entirely.
- Type in the label **Degrading**.
- Since the Degrading is influenced by the Food, draw an influence arrow from the Food to Degrading.
- We want to set the rate of degrading with a variable. So insert a variable and label it **degradingRate**
- Draw an influence arrow from the **degradingRate** to the flow **Degrading**



The model should look something like this…

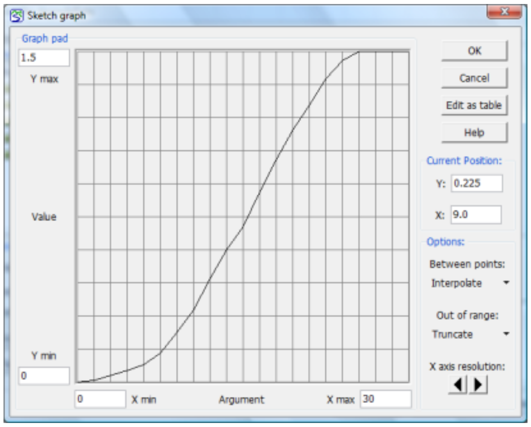
- Set the degrading flow to **degradingRate*Food**
- Set the **degradingRate** to 0.15

Compile this model and run it.

Does this change the conclusions from the original model?

Will the harvest still be large enough for the winter?

Defining a variable from a graph



Y max: 1.5

Y min: 0

X min: 0

X max: 30

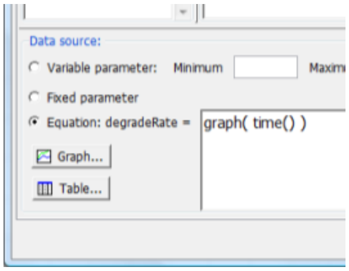
Current Position: Y: 0.225 X: 9.0

Between points: Interpolate

Out of range: Truncate

If the food degrades faster as the month progresses we would want to include this information in the model. To do this we must make the variable **degradeRate** a function of time. One way to do this is to define the **degradeRate** with the graph function.

- Double click on the **degradeRate** variable to open the *equation dialogue window*
- Click on the "Graph…" button in the bottom left of the equation dialogue window.
- The *Sketch graph* window opens.
- Here we can enter the range of the X variable (this will be time, so we set the minimum to zero and the maximum to 30, because 30 days is the end of October). Similarly we enter the range of Y values, which will be the degradingRate. The minimum will be 0 and the maximum 1.5.
- Finally we can draw the graph by clicking with the mouse inside the graph to pull the line into the required form. Above we show the rough form of how the **degradeRate** variable increases with time.
- Finally click the button "OK" to return to the *equation dialogue window* where you will now see "graph()" in the equation box.
- The final step is to tell simile that the X axis of the graph is the time. To do this we enter the time inside the brackets after the word graph (as shown). The time in a simulation is given in Simile by the function **time()**.
- You can now select "OK" and if you've followed the steps correctly the equation should be accepted and you will return to the main window.



● Equation: degradeRate = graph(time())

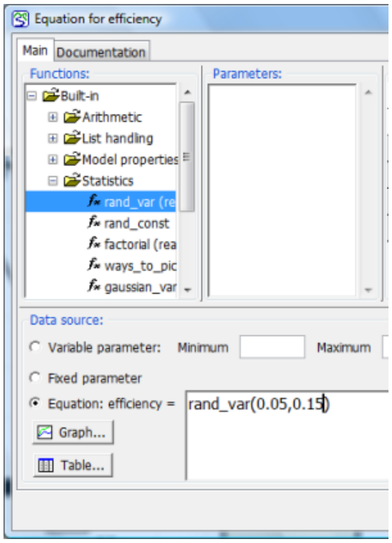
Defining a variable with data from a file

Instead of using a graph, we could have defined the variable **degradeRate** from data in a file. There are two ways of doing this, which are described in the help pages. The first method is with the **table function** (which is below the graph function in the *equation dialogue window*). The second method is to use a **scenario file**.

Random numbers in a simulation

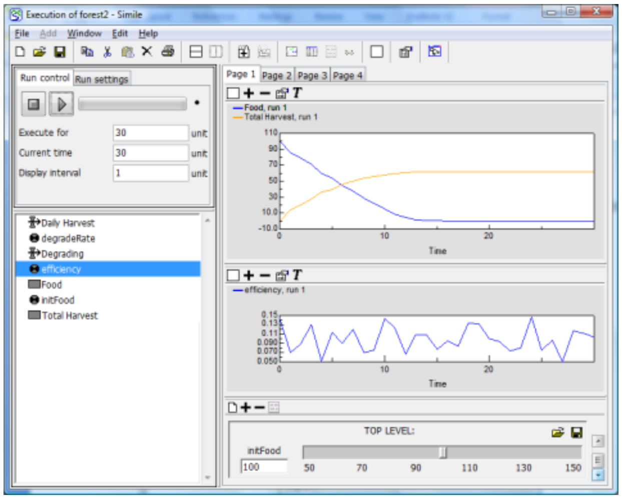
Every variable we have defined up to now has been a well defined number. Simile can also model variables which have random values (e.g. a random number between 0 and 1). For example, the hunting efficiency may not be constant during a simulation (we have set it to 0.1), but may randomly fluctuate about some mean value.

- Create a new variable and label it **efficiency**
- Double click on the **efficiency** variable to open up the *equation dialogue window*



○ Variable parameter: Minimum Maximum
○ Fixed parameter
● Equation: efficiency = rand_var(0.05,0.15)

- In the top left of the window is a section called “Functions:”. Click on the “Statistics” folder and double click on the top entry, which is rand_var(). This function creates a new random number every time step from within a range of possible values.
- rand_var() will have appeared in the box at the bottom of the equation dialogue box. We must now set the range of possible values, which we’ll take as being between 0.05 and 0.15. Enters these values between the brackets on rand_var, so that the full expression looks like “rand_var(0.05, 0.15)”.
- You can then click on the OK button to save this definition.
- Finally we must draw an influence arrow from the variable **efficiency** to the flow **Daily Harvest**, and redefine the Daily Harvest flow to be **efficiency*Food** (instead of 0.1*Food).

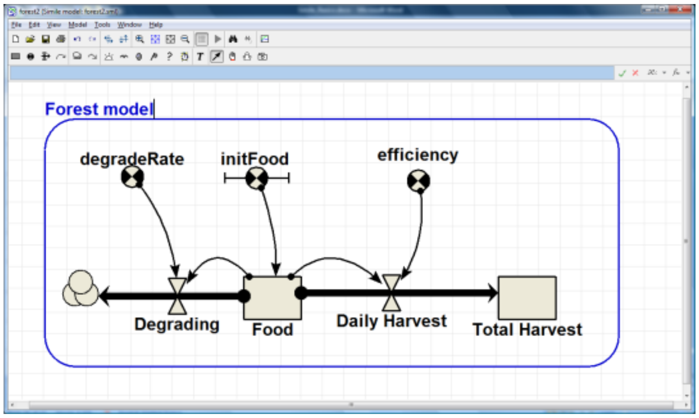


Execute for 30 unit
Current time 30 unit
Display interval 1 unit
initFood 100

The harvesting efficiency is now a random variable which fluctuates daily between 0.05 and 0.15. Running the model and plotting the output gives the following result. We can see that the efficiency is randomly jumping around from one day to the next, but always within our defined range. The effect of this random variation can be seen in the graph of Total Harvest and Food.

Submodels

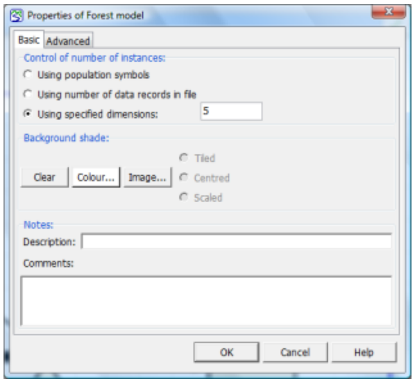
We can enclose all or part of a model within a rectangle, which will define the contents as a submodel. Submodels are useful to make the model diagram easier to read. They are also useful for creating several copies of the same model diagram (e.g. in a spatial population model the same population dynamics could describe populations at different spatial locations, we can use the submodel to create multiple copies of a model, one for each spatial location).



- We want to define our entire model as a submodel. Click on the submodel button in the component bar.
- Use the mouse to draw a box around all the model components, and enter the label as “Forest Model”.

Vectors and matrices

So far all the variables in our model have been single numbers. Simile can define variables which are vectors or matrices (e.g the variable x=[9, 10, 11, 12, 13] is a vector with five elements which happen to be the numbers from 9 to 13). Simile calls a vector a one-dimensional array, and a matrix a two-dimensional array. Arrays are explained in the Simile's help.



Properties of Forest model

Control of number of instances:

○ Using population symbols
○ Using number of data records in file
● Using specified dimensions: 5

Background shade:

○ Tiled
○ Centred
○ Scaled

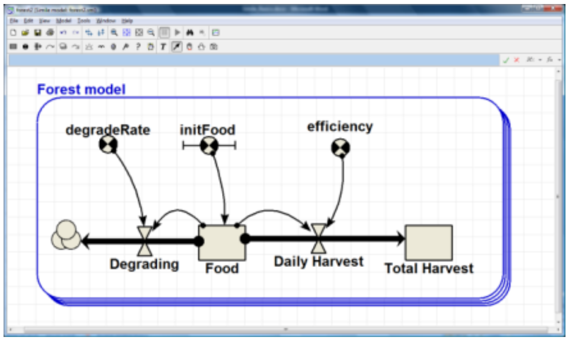
Notes:

Description:

Comments:

We could define 5 copies of the whole model by defining the “Forest Model” submodel into an array. To do this we

- Select the submodel, so that it turns blue.
- Double click in the interior of the submodel to bring up the *Properties window.*
- After the text “Using specified dimensions:” we enter 5.
- Clicking OK will create 5 copies of the submodel.
- This will be shown in the model diagram by the boundary of the submodel having several lines (shown opposite).



Exporting a model as C++ code

Simile is not the only way to simulate these kinds of systems. Many computer programming can be used, but some languages will be easier to learn than others. Some languages will also be more efficient in terms of writing the model, or running the simulation. A common programming language, which is efficient for running a model, is C++. Simile allows a model to be exported into C++ code. To do this select **File/Export/C++ Code** in the menu bar of the main model window and type in the file name where the C++ code is to be saved.

Final Note

This has been a very quick look at Simile. Simile allows you to create and analyze quite sophisticated models, and generate a range of output interfaces. Now to properly understand how to use all these components you're going to have to get stuck in to a modeling project and use Simile.

References

Grant, W. E. and T. M. Swannack (2007). Ecological Modelling: A common-sense approach to theory and practice. New York, Wiley-Blackwell.