

The One-minute Modeller: An Introduction to Simile

Jerome K. Vanclay

*Department of Forestry, Southern Cross University, PO Box 157, Lismore,
NSW 2480, Australia*

ABSTRACT

The Simile programming language provides a powerful and relatively easy to use medium for developing models and simulating the behaviour of forestry systems. This is a highly visual approach to modelling, in that the flow diagram is in effect the computer program. This paper provides a simple introduction to use of the Simile programming language for potential users, which has been developed to provide an initial understanding of the programming features and steps in classes and for workshops.

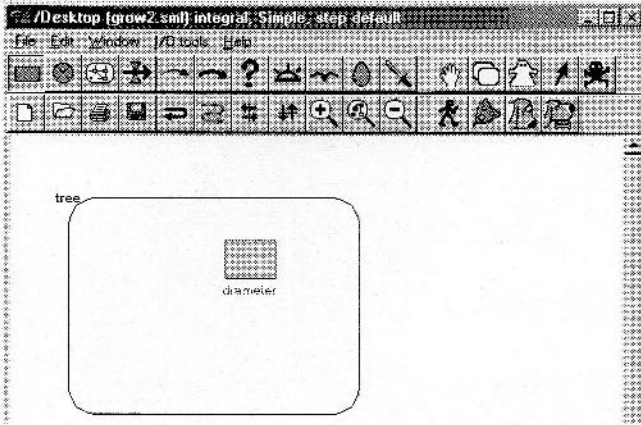
Keywords: forestry modeling; visual simulation; submodels; screen icons; lollipop diagram.

INTRODUCTION

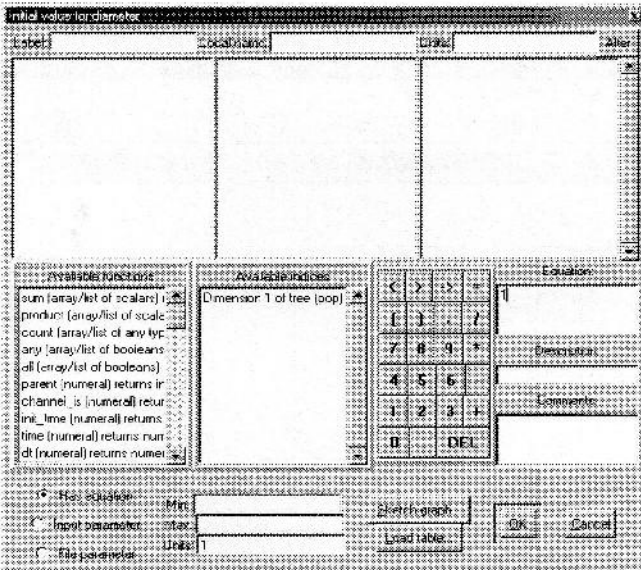
A variety of computer programming languages have been used for developing models to simulate forestry biological processes, stand growth, processing, marketing, human involvement and other 'systems' in forestry. Over time, these model development media are becoming more powerful and easy to use. The Simile programming language, originally known as AME (Agroforestry Modelling Environment), has been developed by researchers at the University of Edinburgh and elsewhere, during the last five years, with a specific focus on forestry modelling. This software is available via the web at <http://www.simulistics.com>. Other languages such as Vensim and Stella offer similar capabilities.

A best-selling book was called 'The One-minute Manager' (Blanchard and Johnson, 1983). In the same spirit, this exposition of Simile can be thought of as 'The One-minute Modeller'. That is, it is a tutorial presentation designed to 'break the ice' for researchers interested in using Simile, by demonstrating some of the main features of the language in an easy-to-understand manner.

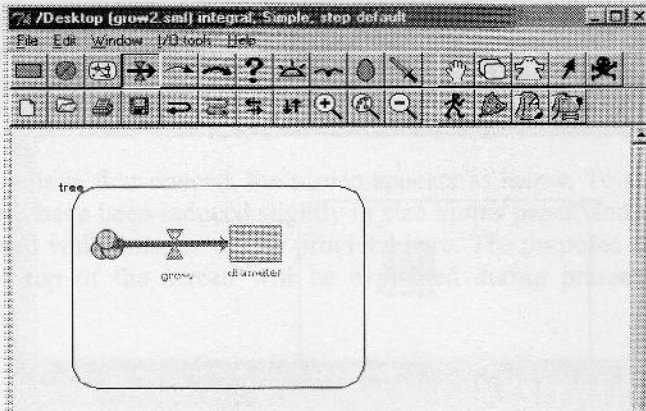
The example used for this exposition concerns growth of a forestry stand. However, the principles apply equally well to other application areas. A number of computer screen images will be presented, to indicate menu options and appearance of the visual model throughout the development steps. While there is



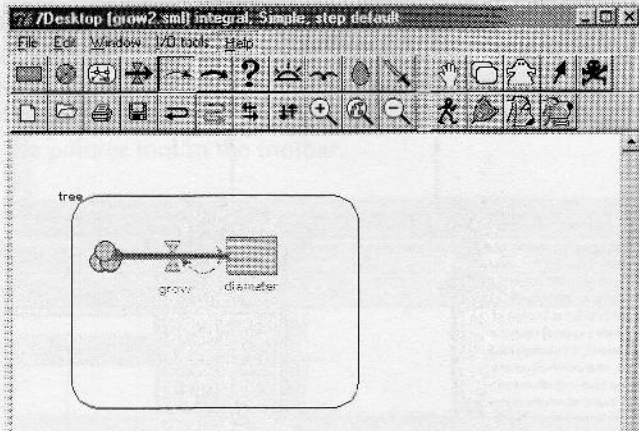
Double-click on the compartment symbol to obtain the following dialogue box. Enter an expression of say 1 in the Equation box. Here the diameter is specified in centimeters



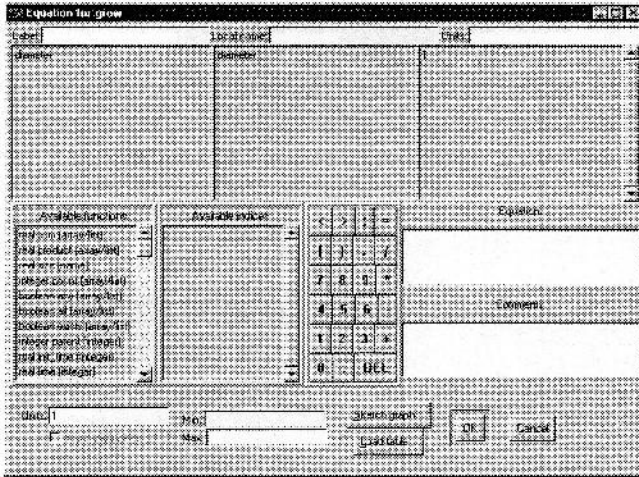
Trees grow, so it is necessary to allow the diameter to increase. This is achieved by creating a flow into the compartment called diameter. The flow tool is the icon showing an arrow with an engineering symbol for a tap or regulator. The flow has been renamed as 'grow'.



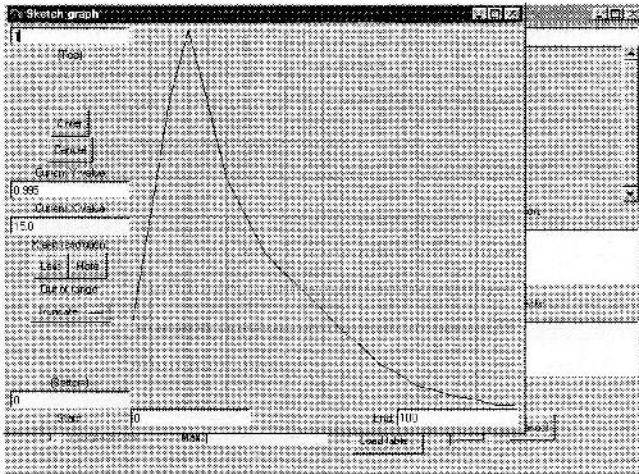
Tree growth may depend on many things, but here only size dependent growth will be modelled. This calls for drawing an influence arrow from 'diameter' to 'grow' in the model diagram. Click the influence tool with a curved thin arrow in the toolbar and draw an influence arrow from the compartment symbol to the flow symbol.



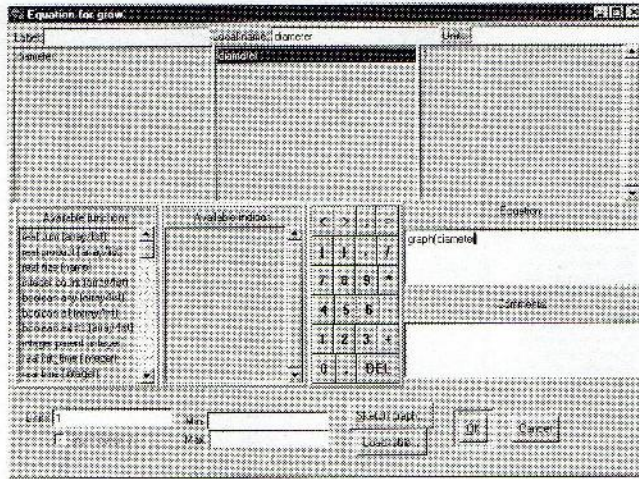
Next, a growth relationship must be supplied. By double-clicking on the flow icon called 'grow', the following dialogue box is obtained:



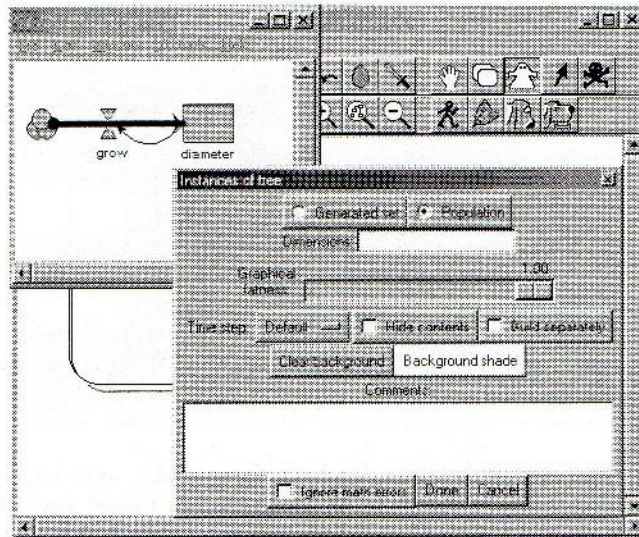
Not everyone is comfortable with equations. To avoid the maths, the relationship may be supplied as a hand-drawn sketch. By clicking on the 'Sketch graph' button, a graph window is obtained. A graph of any shape can be sketched with the mouse on the graph window. The default is to interpolate on 20 line segments, but this can be varied with buttons for "Less" and "More" X-axis resolution.



Notice that start and end values for both the x and y axes can be provided. The relationship can now be accepted by pressing the 'Enter' key. Simile knows that the y-axis represents the growth rate, but it is necessary to specify in the 'Equation' box that the x-axis represents the diameter of a tree. Having done that, press the 'OK' button. A model has been built for the first tree for the moment.

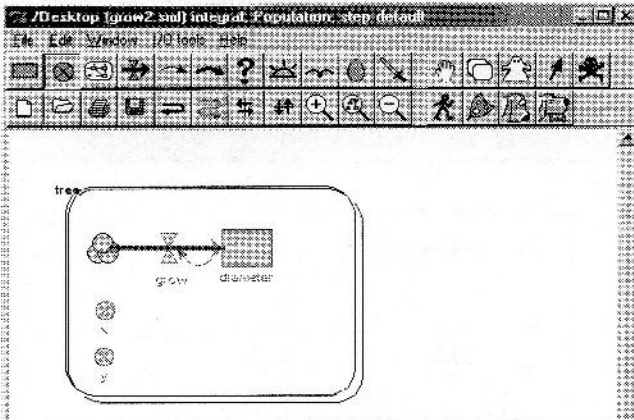


Now, the objective is to model not just one tree, but a forest. To do this, double-click on the boundary of the tree submodel. A new window for the tree submodel appears. In this window, select 'Properties' under 'Edit' on the menu bar. Next, specify the number of instances of the tree submodel in the 'Dimensions' box. If the aim was to model a plantation, a set of say 1000 trees may be generated. However, here the interest is in a natural forest, in which the number of trees varies continually. Thus, just check that the population option has been chosen, with a dot in the 'population' circle.

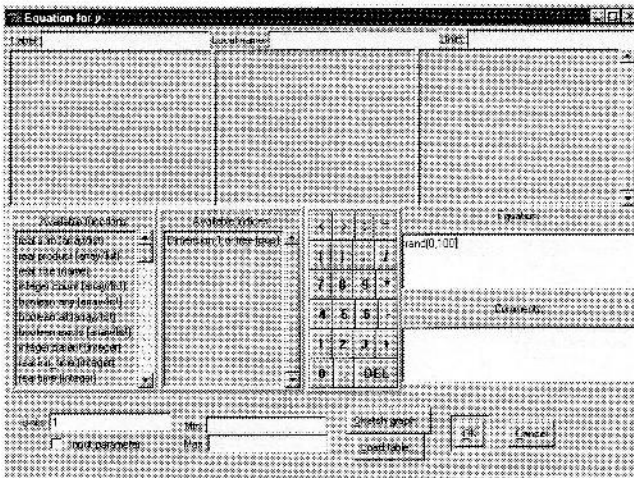


When the 'Done' button is pressed, it will be noticed that the tree submodel is no longer represented as a single instance, but has double lines around the

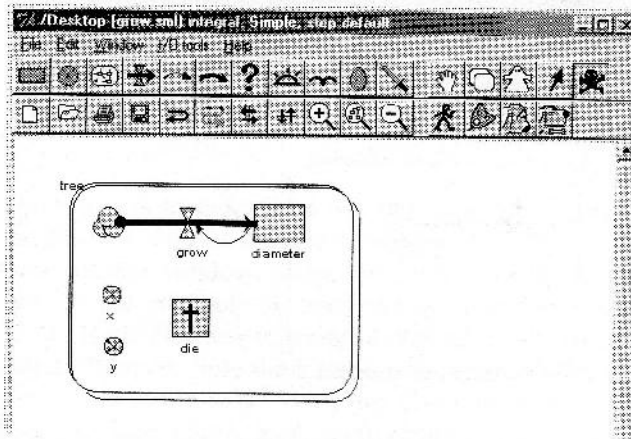
bottom-right and top-left corners like an untidy pack of cards. This is the Simile notation for a population. Now, in order to generate photo-like diagrams of the forest, it is necessary to create some variables to describe the position of the trees in the forest. Click on the variable tool (a circle icon with a cross in it), and then click in the model diagram to deposit a variable symbol. Repeat (another mouse-click in the model diagram) to add additional variables. Two variables have been renamed as x and y, to represent the x- and y-coordinates of trees within the forest.



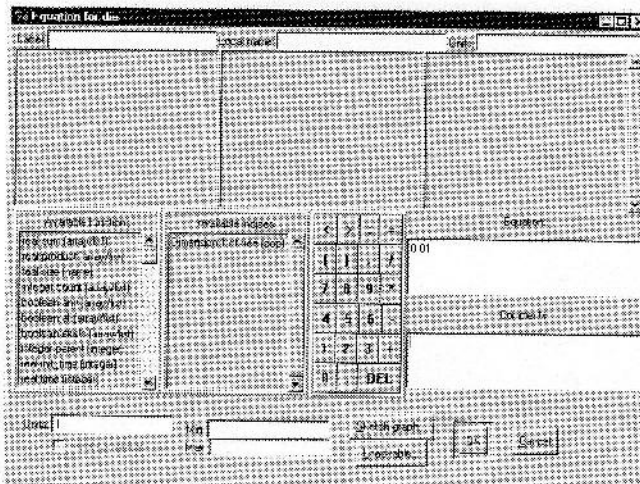
By double-clicking on each variable symbol, the following dialogue box is obtained. It is necessary to specify the initial value for each tree. For simplicity, random numbers between 0 and 100 will be chosen for each tree. Enter the expression `rand_const(0,100)` in the Equation box to generate a random constant. The location, size and growth of trees have now been specified.



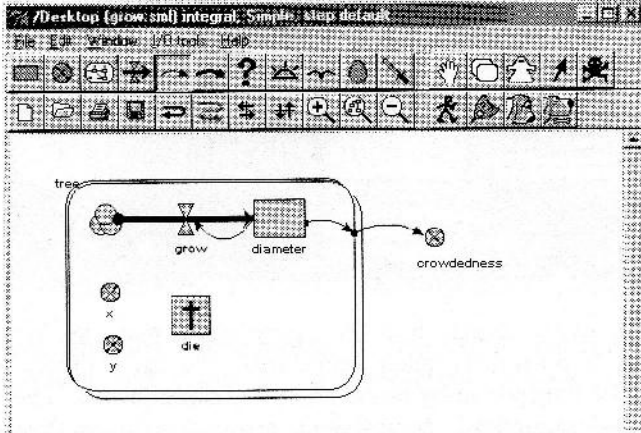
Since these trees are not immortal, it is necessary to specify a mortality function. The mortality tool is used to specify the destruction of instances of a population submodel. Click on the mortality tool (the icon with a sword in the toolbar), and then move the mouse to within the envelope of the population submodel to deposit the mortality symbol in the submodel. Notice that the icon with a cross instead of a sword turns up. The next version of Simile will use a hatchet for both the tool bar and the icon, but for the moment, Simile users just have to accept both the sword and the cross and recognise that they mean the same thing.



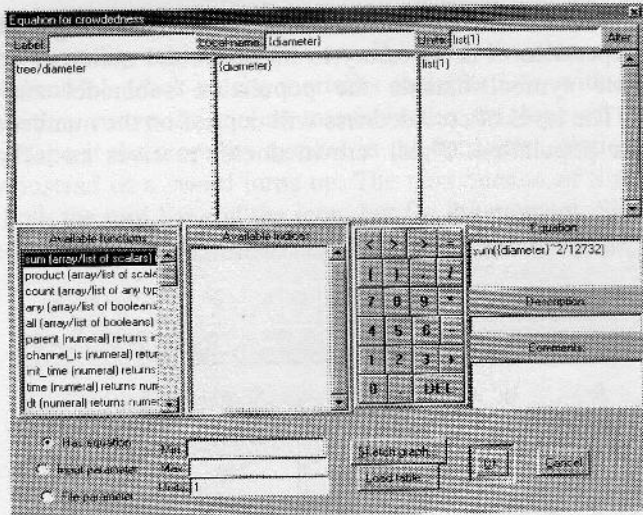
By double-clicking the mortality symbol renamed as 'die', the following equation dialogue window appears. Suppose a 1% mortality rate is assumed, irrespective of size, competition or any other factor. To express this, enter 0.01 in the Equation box.



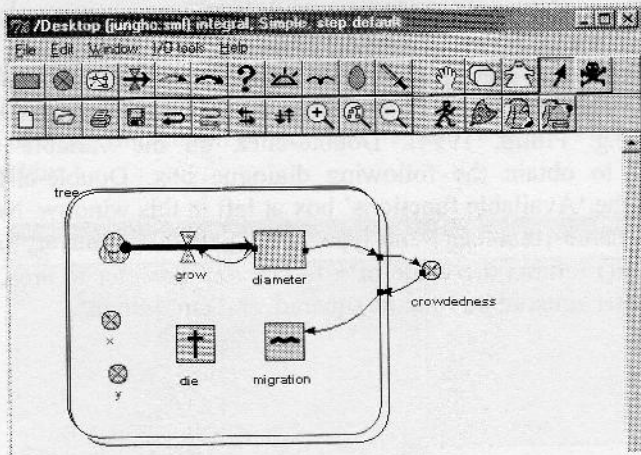
Next, to introduce new trees into our population as seedlings, while making them density-dependent, it is necessary to introduce the notion of crowdedness. Place a variable symbol outside the population submodel and rename it 'crowdedness'. The level of crowdedness will depend on the number and sizes of the trees in our population. Thus, 'crowdedness' receives an influence arrow from 'diameter'.



The forester's concept of stand basal area is adopted here, which is the sum of the sectional areas of the tree stems, because it is easy to compute, and because it is well correlated with many interesting properties of a forest, such as biomass, biodiversity and evapo-transpiration. Stand basal area is simply the sum of the squares of the diameters, converted to cross-sectional area in square metres per hectare (see e.g. Philip, 1994). Double-click on the variable symbol for 'crowdedness' to obtain the following dialogue box. Double-click the sum function from the 'Available functions' box at left in this window. Next, double-click the local name {diameter} and type $^2 * \pi() / 40000$ to convert to basal area. The function $\pi()$ returns the value of π to convert diameter to area, and 40000 converts diameter squared into radius squared, and cm^2 into m^2 .

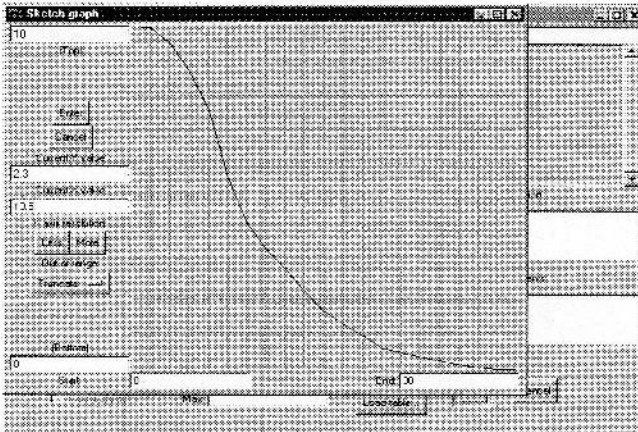


Recruitment of seedlings into the population may be represented as ‘migration’ (the bird symbol) rather than ‘birth’ (the egg symbol), to recognise that seeds may be transported by birds, wind, and other factors. This process has been renamed as ‘migration’. An influence arrow goes from ‘crowdedness’ to ‘migration’.



Double-click on the bird symbol to obtain a dialogue box. Again to avoid equations, click on the ‘Sketch graph’ button. A graph may be sketched illustrating the negative relationship between stand density and the number of seedlings that survive to the end of the first year. The graph predicts numbers of recruits (stems/ha/year) from stand basal area (m^2/ha). It seems reasonable to assume say 10 recruits at low stand density, with an asymptotic decline

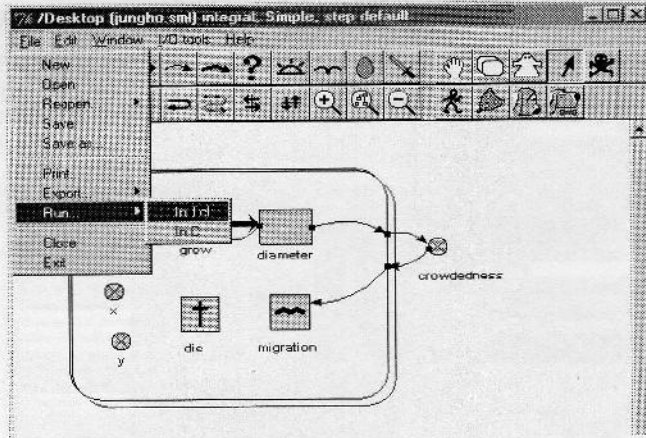
approaching zero as stand basal area reaches 30 m²/ha. The relationship can now be accepted by pressing the 'Enter' key.



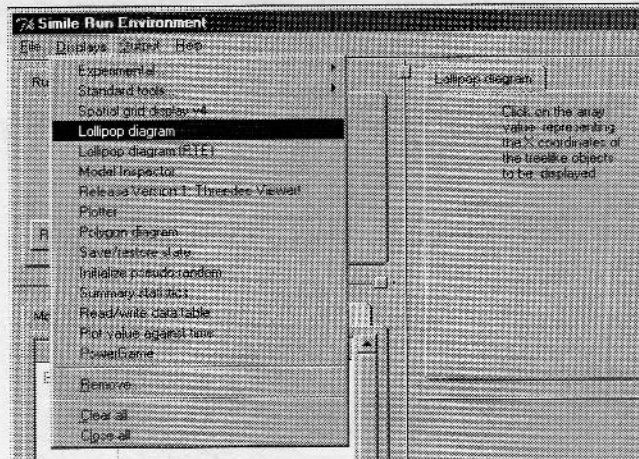
Simile knows that the y-axis represents 'migration', but it is necessary to specify in the 'Equation' box that the x-axis represents 'crowdedness'.

RUNNING THE MODEL

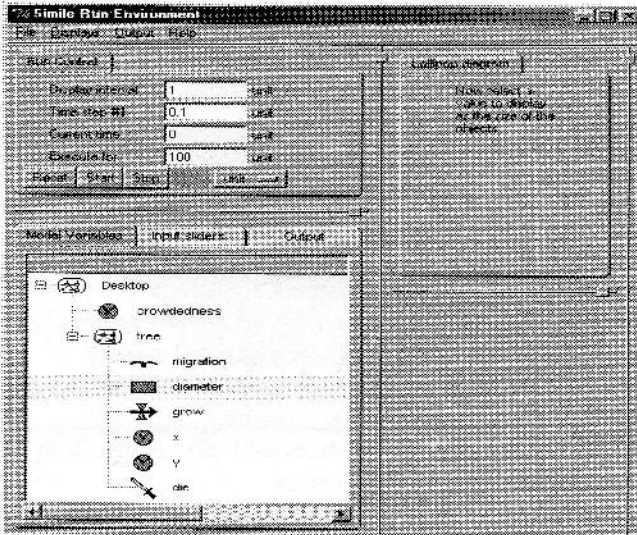
Now that the model has been developed, it is time to run it and inspect the output. In this demonstration, the model will be run with the TCL interpreter, rather than cross-compiling it as C-code allowing run time to be reduced. The two options are available in the Run command under the File menu.



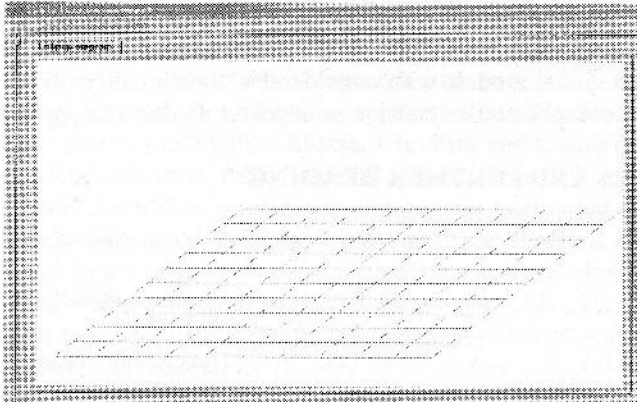
When 'In Tci' under File/Run is selected, the Simile Run Environment window appears. Some displays may be chosen to improve the output display. One of the standard Simile displays will be used here, namely the 'lollipop diagram'.



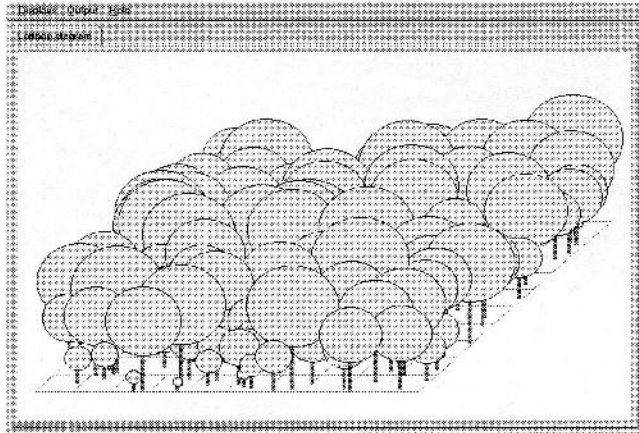
When the lollipop diagram is selected, the x- and y-coordinates must be chosen. Also, the object to be displayed, which is diameter in this forestry example, must be indicated.



Simile will draw a picture of a forest in the lollipop window, with no trees because it hasn't been initialized. Maximise the lollipop window.



When the 'Start' button in the Run control box within the Simile Run Environment window is pressed, trees start to grow. In the mean time, some trees die, and new instances are created. When the 'Stop' button is pressed, Simile will now display a picture of the forest with the size of the trees at a given time. This forest simulation is an impressive example of what can be achieved. It could, however, be extended to have more than one tree species, to make mortality size dependent, to make the growth rate dependent on competition as well as tree size, and so on. But this demonstration has been sufficient to formalize and communicate some ideas about tree growth, and to test whether the ideas are reasonable, in both the logical and empirical sense.



CONCLUDING COMMENTS

Simile is a powerful medium for model development and implementation as a computer program. The example model developed here has demonstrated various features on Simile, including submodels, flows, populations, and birth and death.

This model took just one minute to build, a little less to run, and a little more to document. Hopefully, this example has illustrates some of the capabilities of Simile, and demonstrated that, using this powerful model and program development medium, models with considerable functionality do not necessarily need to be big, complicated, expensive or excessively demanding on data.

REFERENCES AND FURTHER READING

- BLANCHARD, K.H. and S. JOHNSON. 1983. *The One-minute Manager*. Berkeley Group, Berkeley.
- MUETZELFELDT, R.I. and J. TAYLOR. 1997. The suitability of AME for agroforestry modelling. *Agroforestry Forum*. 8(2): 7-9.
- MUETZELFELDT, R.I. and J. TAYLOR. 2001. Developing forest models in the Simile visual modelling environment. Paper to IUFRO conference on *Forest Biometry, Modelling, and Information Science*. June 2001. Greenwich. <http://www.ierm.ed.ac.uk/simile/documents/iufro3.pdf>. Accessed 30 January 2003.
- PHILIP, M.S. 1994. *Measuring Trees and Forests*. CAB International, Wallingford, UK.