Ecological modelling with Simile

Lecture 1

Part A: Introduction to ecological modelling
Part B: Introduction to Simile
Part C: System Dynamics in Simile

Robert Muetzelfeldt
Jasper Taylor
Jonathan Massheder

www.simulistics.com

Aims of course

- How to model (using Simile as the modelling platform)
- How to use Simile to model (for those with experience in modelling)
- Raise awareness of possibilities
- Role of modelling in the research community

Part A
Introduction to ecological modelling

Core concepts

- Purpose
- Idealisation
- Design
- Syntax
- Semantics
- Modelling paradigms

Purpose

- “a model for...”, not “a model of...”
- Prediction
  Management
  Testing understanding
- Problems when purpose is <u>not</u> clearly defined: e.g. IBP models

Idealisation

- The simplification needed to satisfy our purpose
- No need to apologise for an appropriately-simplified model

Design

- cf architectural design
  map making
- Design criteria:
  accuracy, use of data, cost of development, ease of use, simulation speed, understandability
- Constrained by available building blocks

Syntax

- The elements of the design language: vocabulary
- How they can be put together: grammar

Semantics

- What do the building blocks 'mean'?
- What constitutes good model design?
- How does the model relate to real-world objects and relationships?

Modelling paradigms

- Paradigm: "a conceptual framework within which scientific theories are developed"
- A "school of thought" within which the modeller operates
- Examples:
  System Dynamics — differential equation
  object-based — multi-agent
  statistical — probabilistic
  rule-based — cellular automaton
  linear programming — discrete-event

Compartment-flow (System Dynamics) is a good paradigm for ecological modelling because...

- it builds on existing concepts
- it's diagrammatic
- it's in widespread use
- it encourages a layered approach (conceptual structure before mathematical detail)
- it's applicable to a wide range of ecological and environmental problems
- it's a suitable basis for computer modelling software

Object-oriented modelling is a good paradigm for ecological modelling because:

- there's a close correspondence between the objects in the real world and the objects in the model.
- it reflects the idea that many individuals follow the same rules.
- it enables us to talk about the hierarchical composition of some ecological system.
- it enables us to describe relationships between things (shading, closeness, ownership).
- it enables processes of creation and destruction of things to be represented in a natural way.

Part B
Introduction to Simile

Background to Simile

- DFID/FRP Agroforestry Modelling Project
- Undergrad/MSc teaching
- FLORES
- ModMED
- Commercialisation

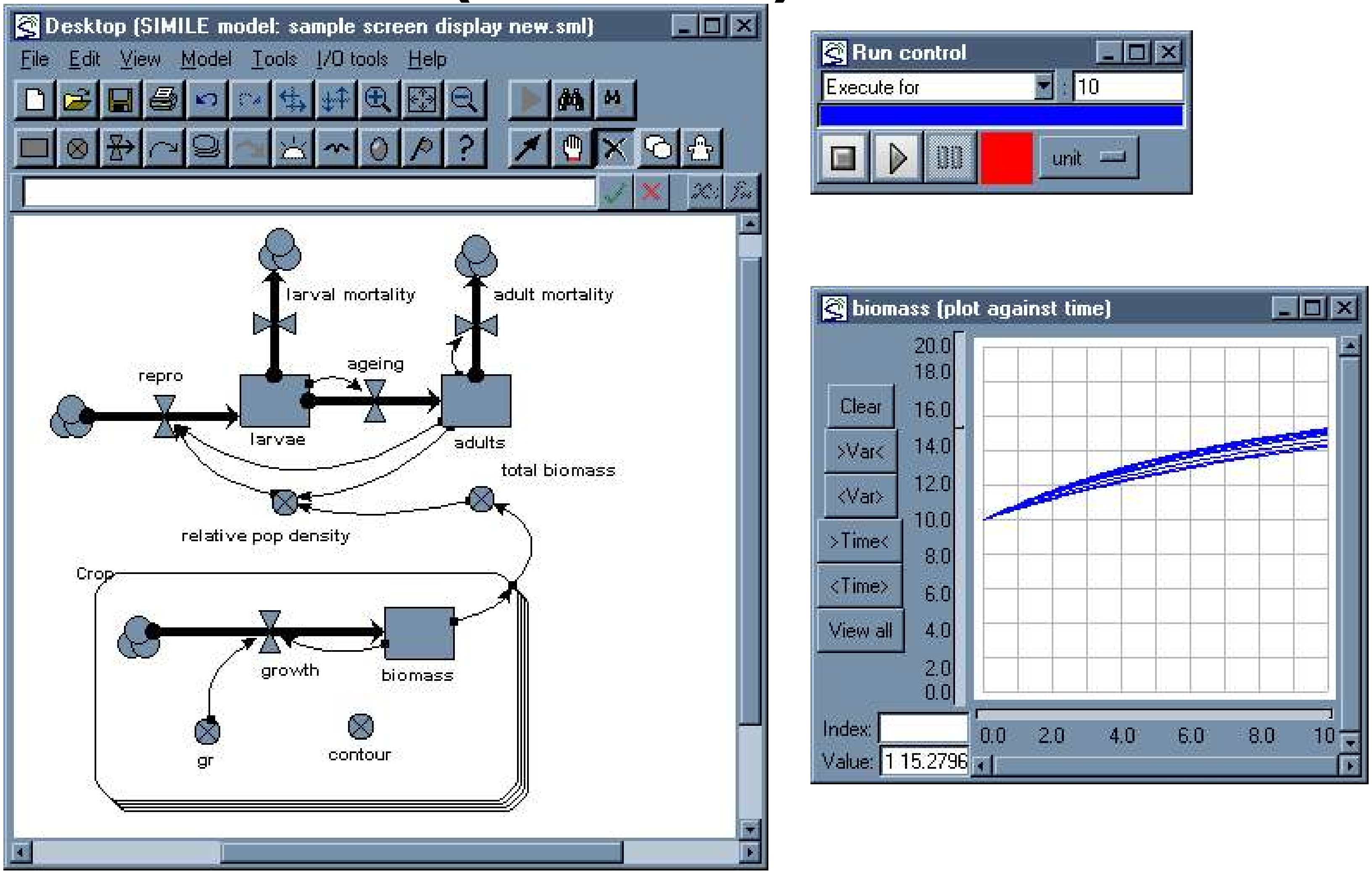
Simile: key features

- Combines System Dynamics and object-based modelling approaches
- Intuitive graphical user interface
- Highly-efficient simulations (in C++)
- Supports modular modelling
- Customisable input/output tools
- Open format (Prolog/XML) for saved models

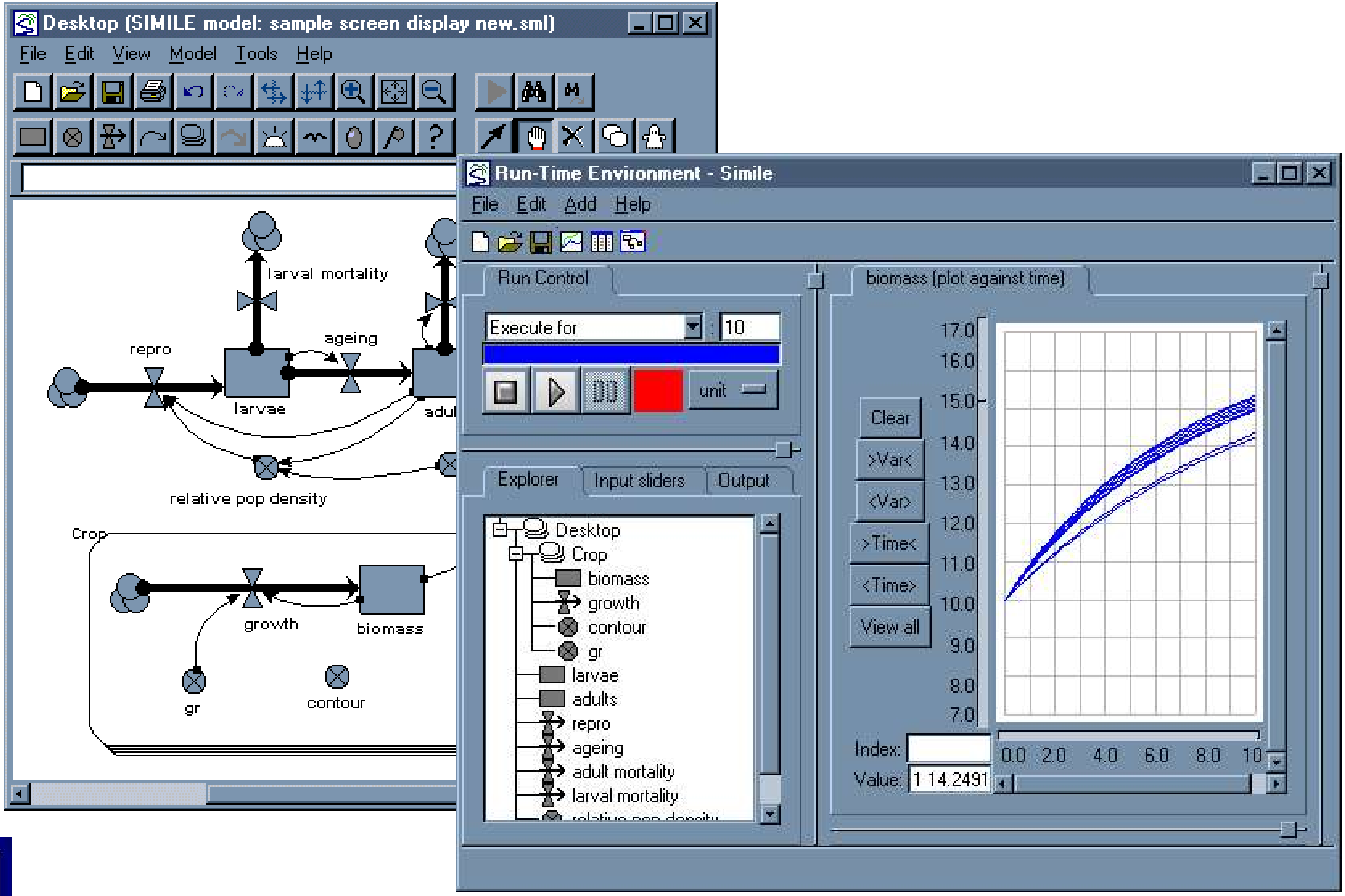
Modelling concepts supported by Simile

- System Dynamics
- Differential/difference equations
- Age/size/sex/species classes
- Objects: multiple, create/destroy, associations
- Spatial: layers, grid, polygons etc
- Modularity

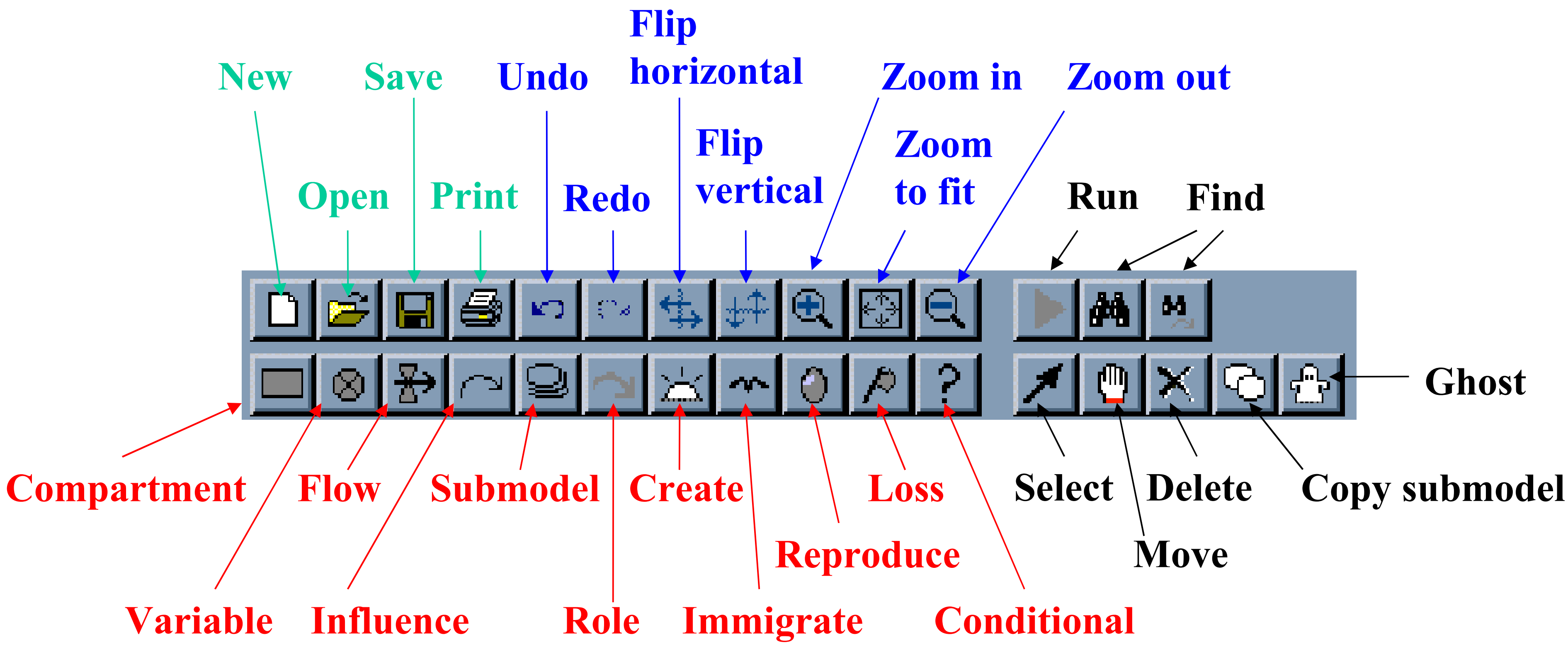
Sample screen display (classic)



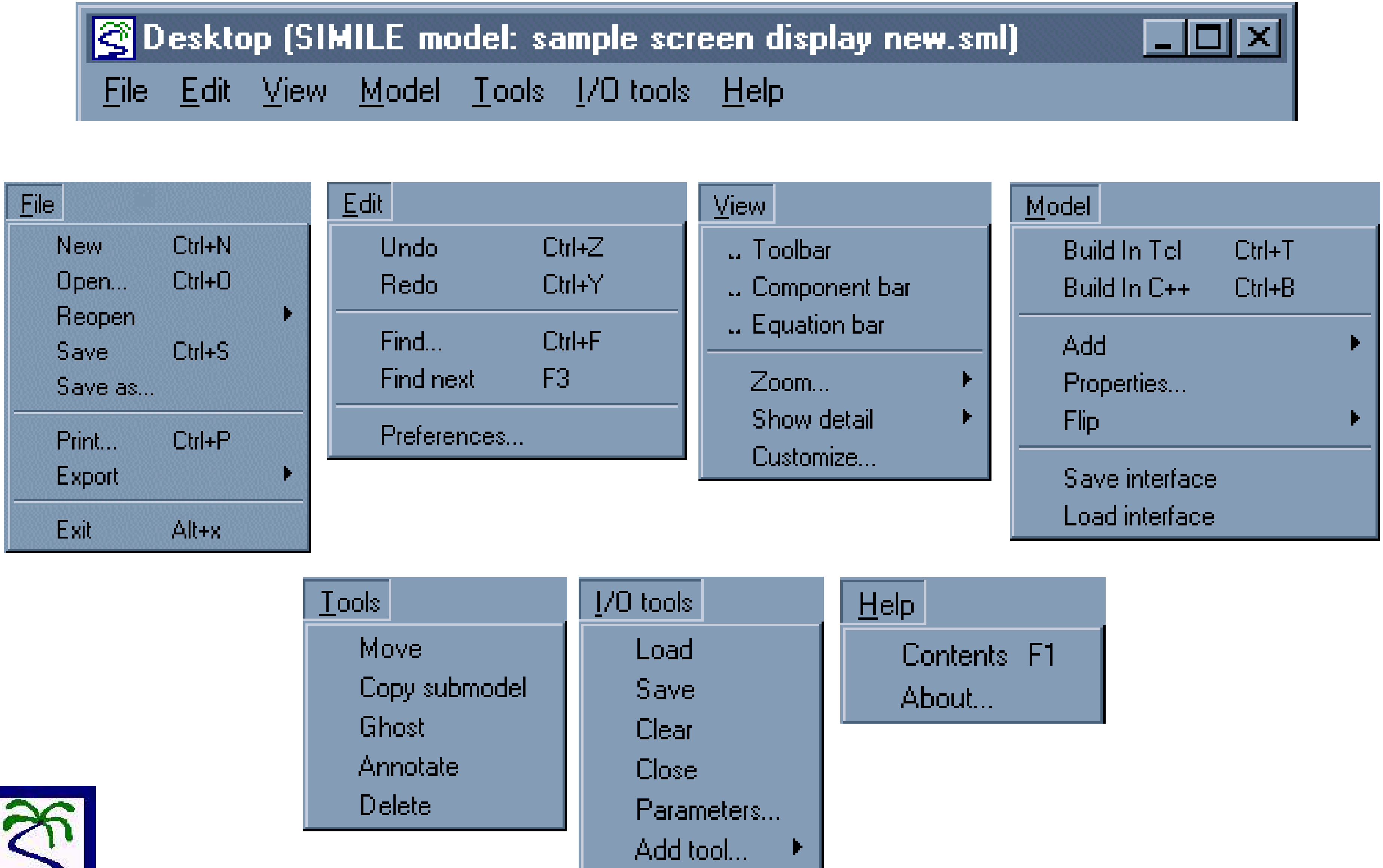
Sample screen display (new)



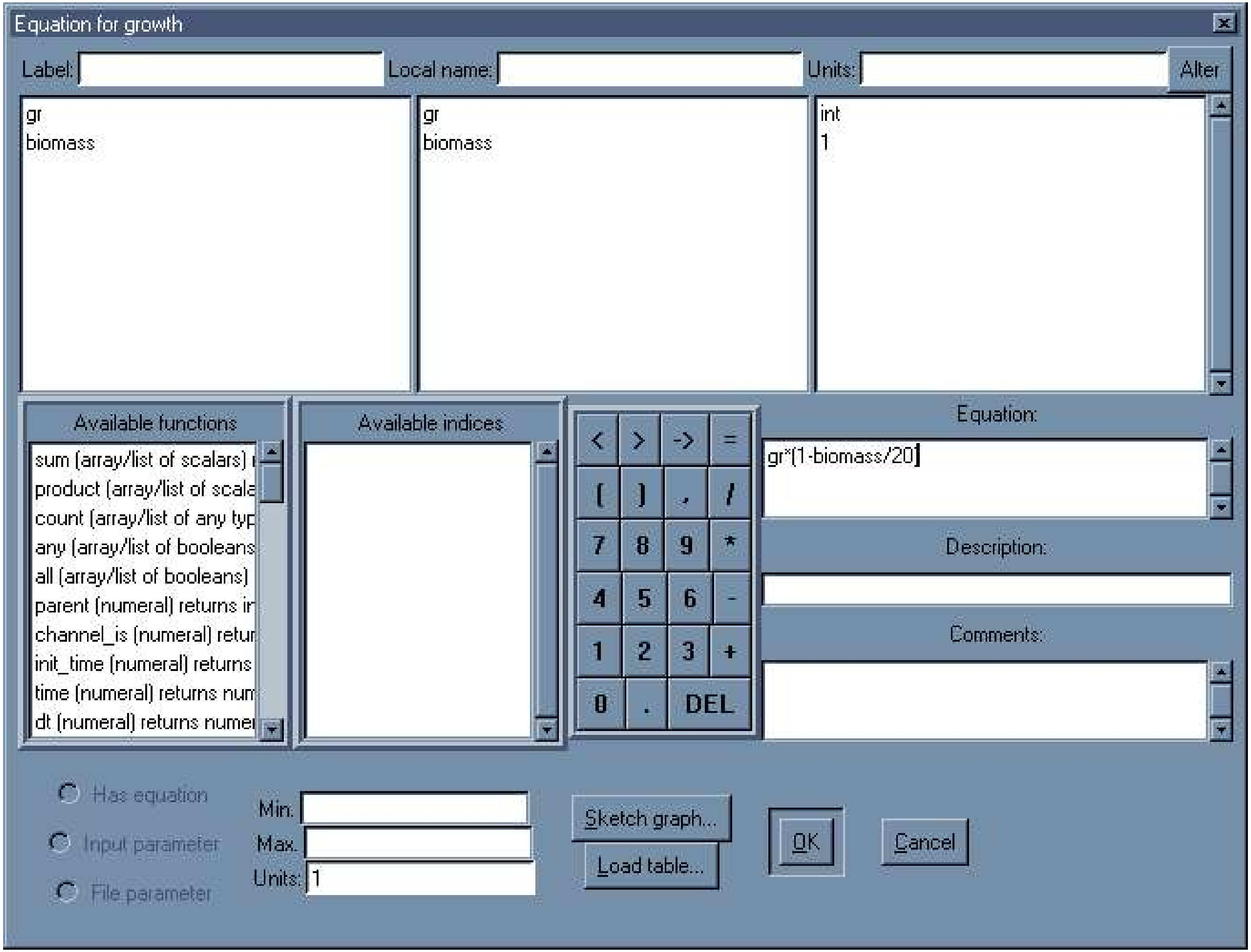
The Toolbar


New
Open
Save
Print
Undo
Redo
Flip horizontal
Flip vertical
Zoom in
Zoom to fit
Zoom out
Run
Find
Ghost
Compartment
Flow
Submodel
Create
Loss
Reproduce
Variable
Influence
Role
Immigrate
Conditional
Select
Delete
Copy submodel
Move

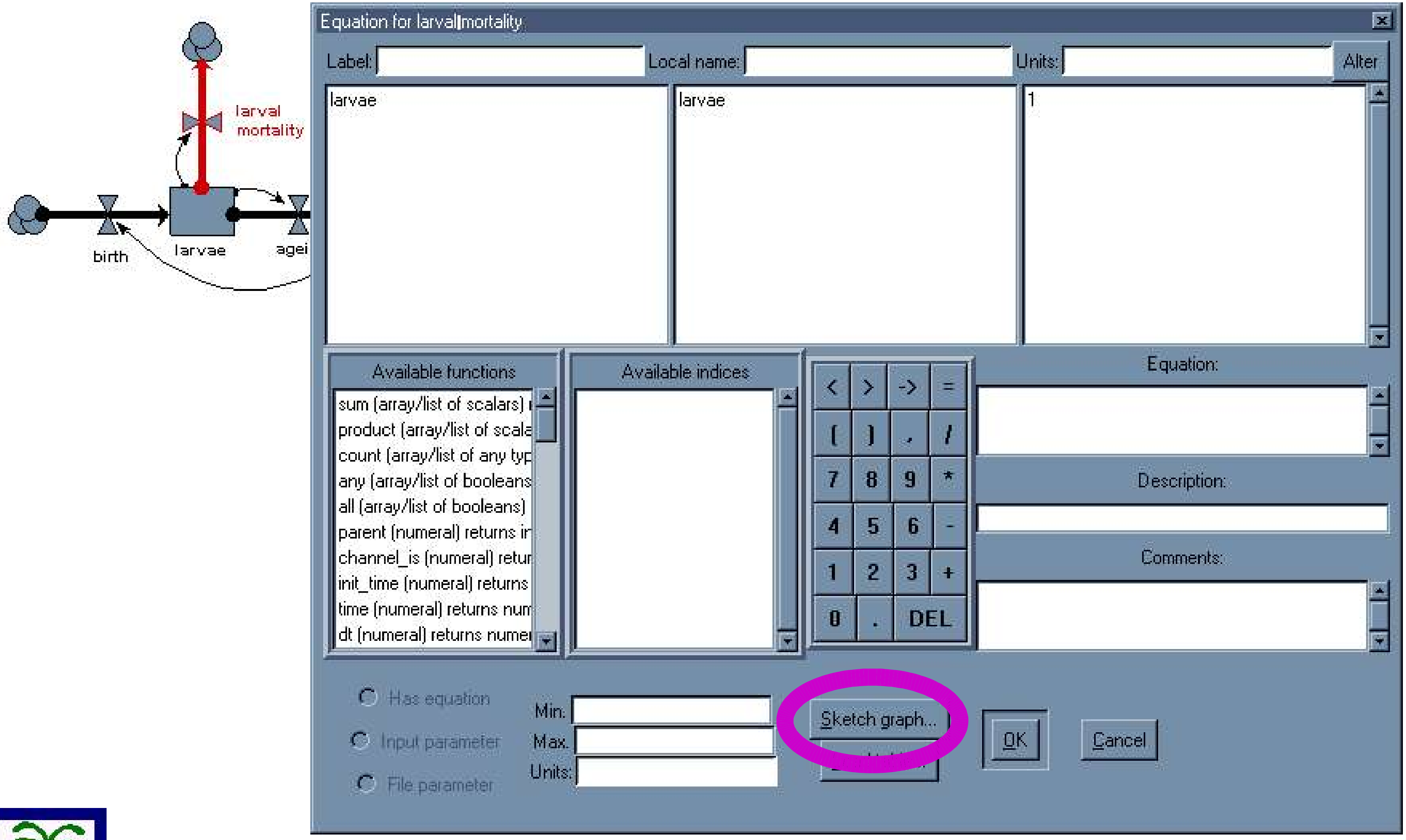
The Desktop


Desktop (SIMILE model: sample screen display new.sml)
File Edit View Model Tools I/O tools Help
File
New Ctrl+N
Open... Ctrl+O
Reopen
Save Ctrl+S
Save as...
Print... Ctrl+P
Export
Exit Alt+x
Edit
Undo Ctrl+Z
Redo Ctrl+Y
Find... Ctrl+F
Find next F3
Preferences...
View
Toolbar
Component bar
Equation bar
Zoom...
Show detail
Customize...
Model
Build In Tcl Ctrl+T
Build In C++ Ctrl+B
Add
Properties...
Flip
Save interface
Load interface
Tools
Move
Copy submodel
Ghost
Annotate
Delete
I/O tools
Load
Save
Clear
Close
Parameters...
Add tool...
Help
Contents F1
About...

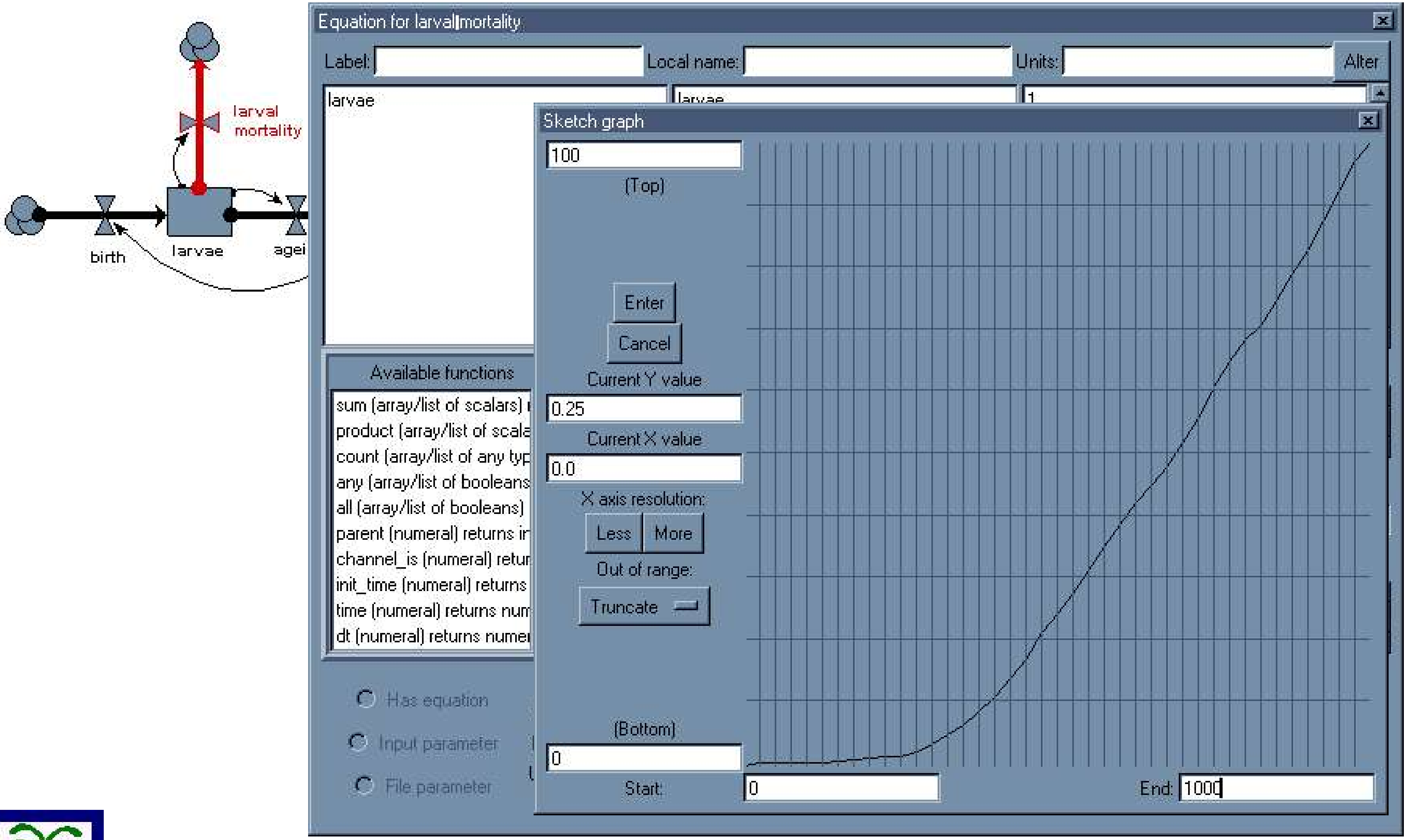
The Equation Dialogue



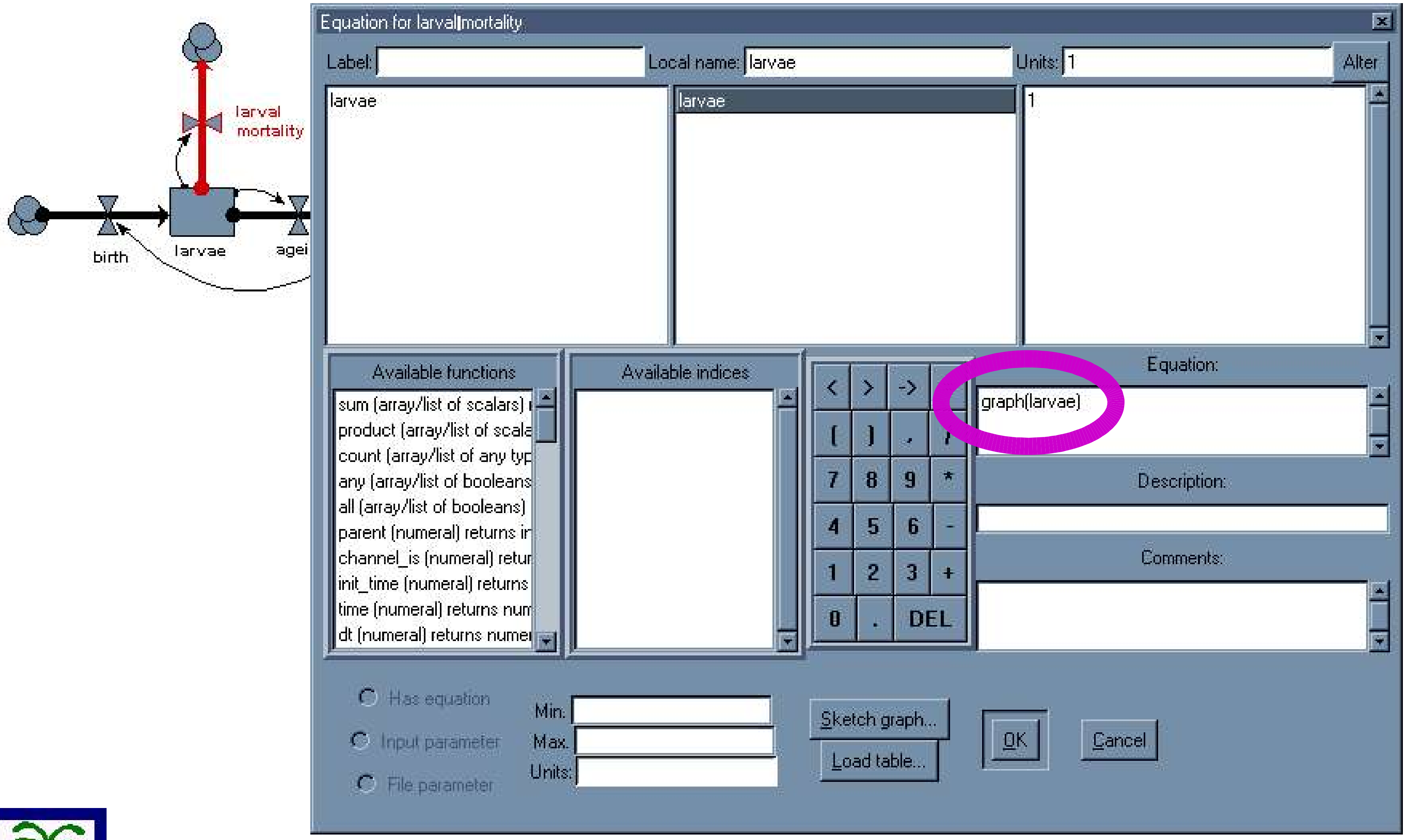
The sketch-graph window (1)



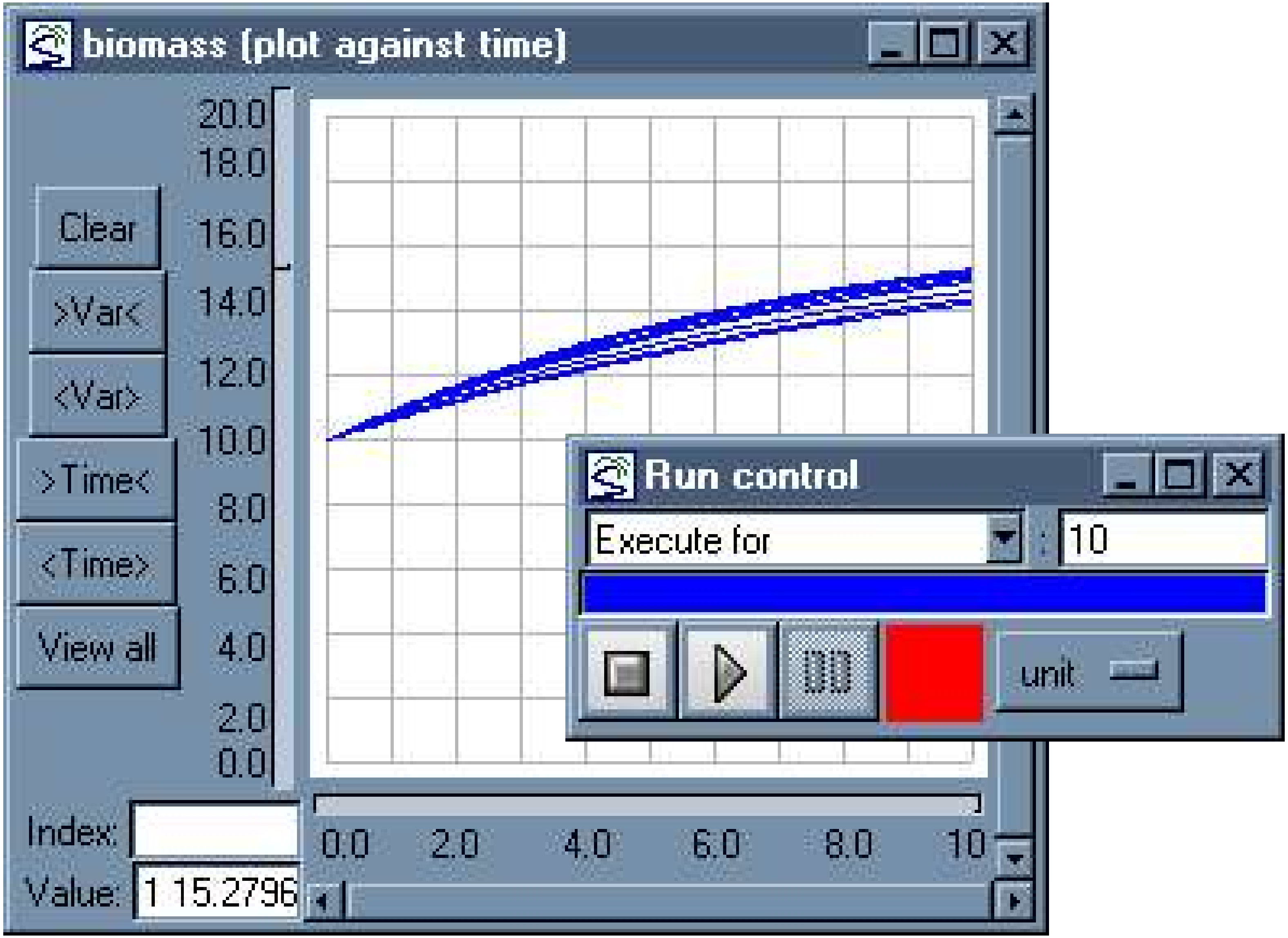
The sketch-graph window (2)



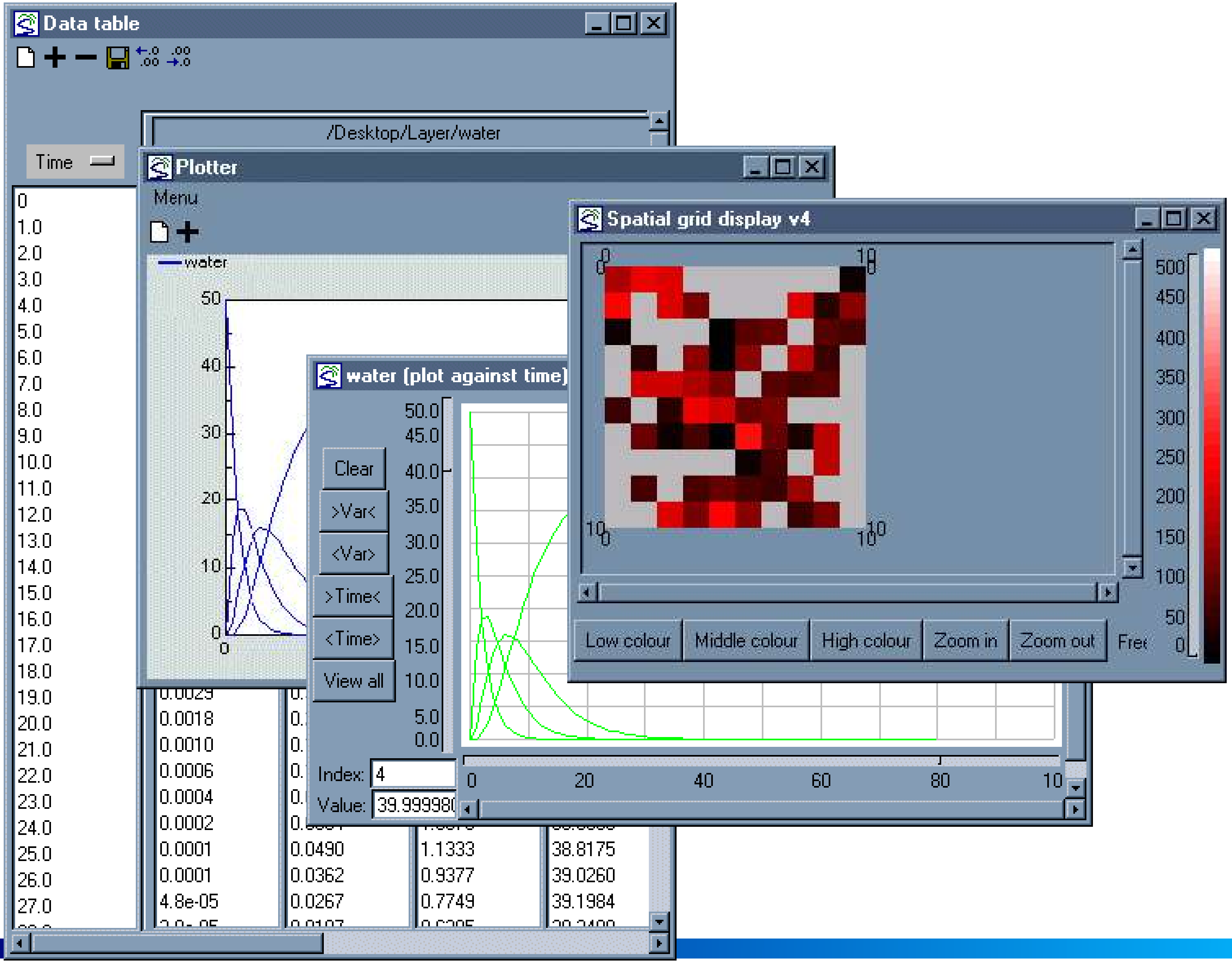
The sketch-graph window (3)



Run Control and Helper windows



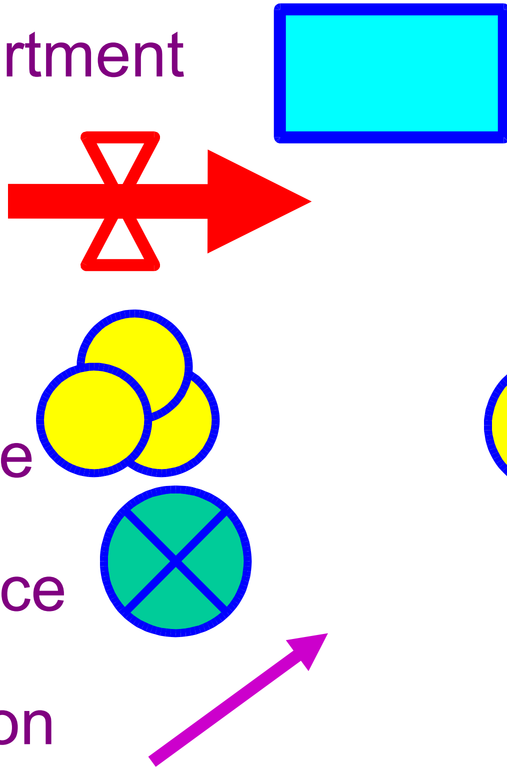
Some display helpers

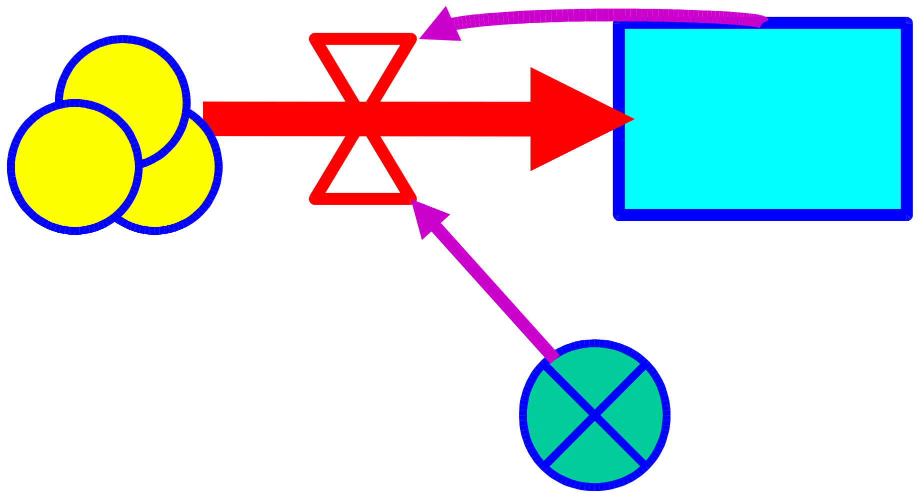


Part C
System Dynamics in Simile

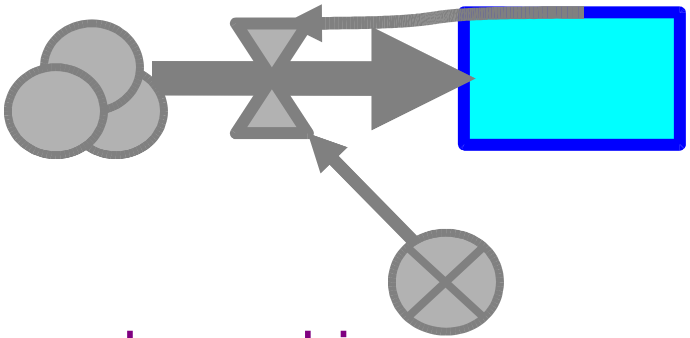
System Dynamics symbols

- compartment
- flow
- cloud
- variable
- influence
- equation



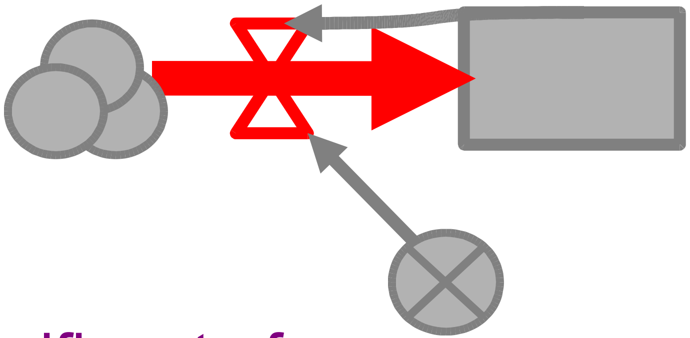


Compartment



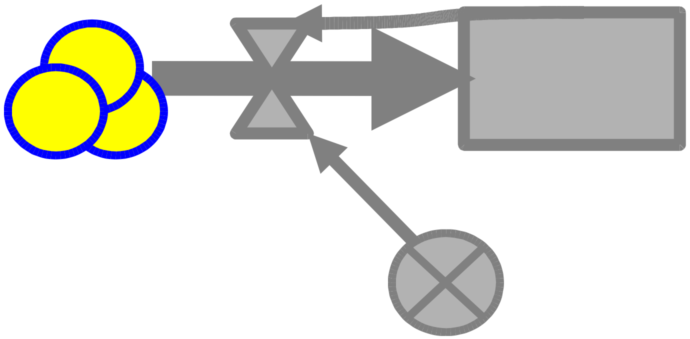
- amount of some substance
- ideally, label as object:substance e.g. sheep_biomass
- typical units: kg, kg m-2, numbers, numbers ha-1
- exceptionally, non-substance things like height, position
- also called 'state variable’, stock, level
- unlike real compartments, can go negative
- unlike real compartments, has infinite capacity
- cannot receive an influence arrow - changes only from flows
- rate of change is the net effect of all inflows minus all outflows
- two connected compartments must have same substance, same units
- can only contain one substance

Flow



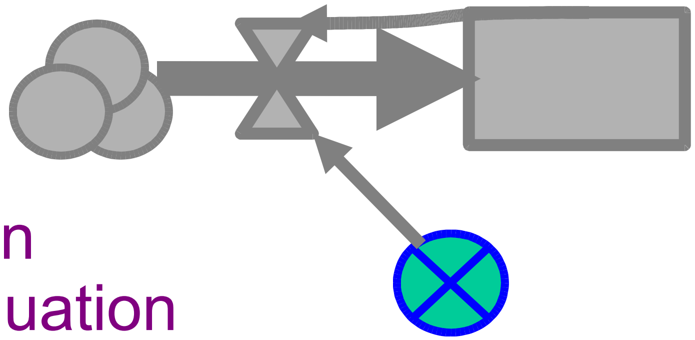
- usually corresponds to a process
- represents an absolute rate (cf specific rate for some parameters)
- units are compartment units/time
- corresponds to the additive terms in a differential-equation model
- can be negative (flow in reverse direction)
- may be several flows between two compartments, in either direction
- preferable to have one flow for each separately-analysable process (e.g. types of mortality)

Cloud



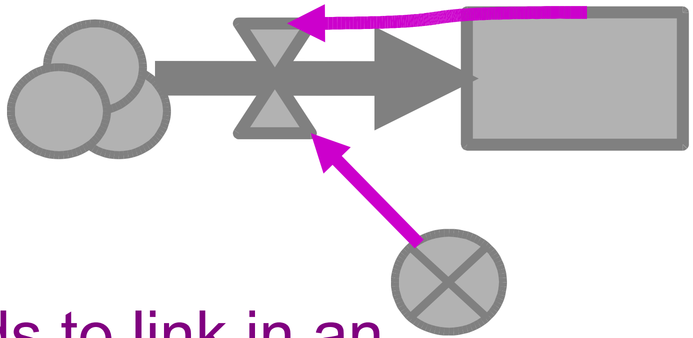
- used exactly like a compartment (at start or end of a flow)
- its value is irrelevant, unknown, unspecified (so can't be set or plotted)
- corresponds to 'the outside world'
- cannot receive or be the source of an influence arrow

Variable



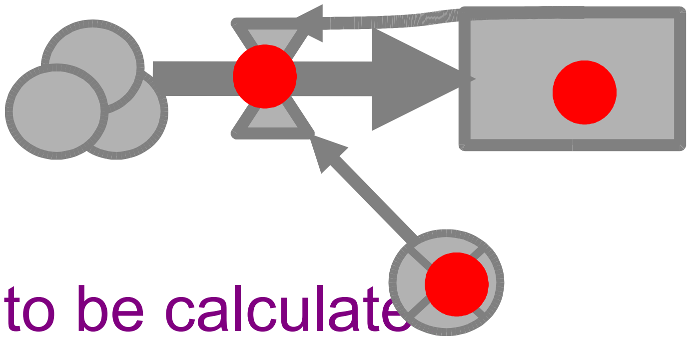
- parameter:
  - 'constant' during simulation run
  - typically, a coefficient in an equation
  - never needed: can use numeric value in equation instead
- intermediate variable
  - both source and recipient of influence arrows
- output variable
  - used only for reporting on model behaviour
- exogenous variable
  - an 'external variable': just a function of time
  - influences the model, but is not influenced by it
  - typically, climatic factors

Influence



- often (but not always) corresponds to link in an 'influence diagram'
- captures the idea that something affects something else
- formally, represents the fact that one term is used in the calculation of another (i.e. appears on the right-hand-side of the function used to calculate the other)
- can start from a compartment, flow or variable
- can go to a flow or a variable (not to a compartment, except to for initialisation)

Equation

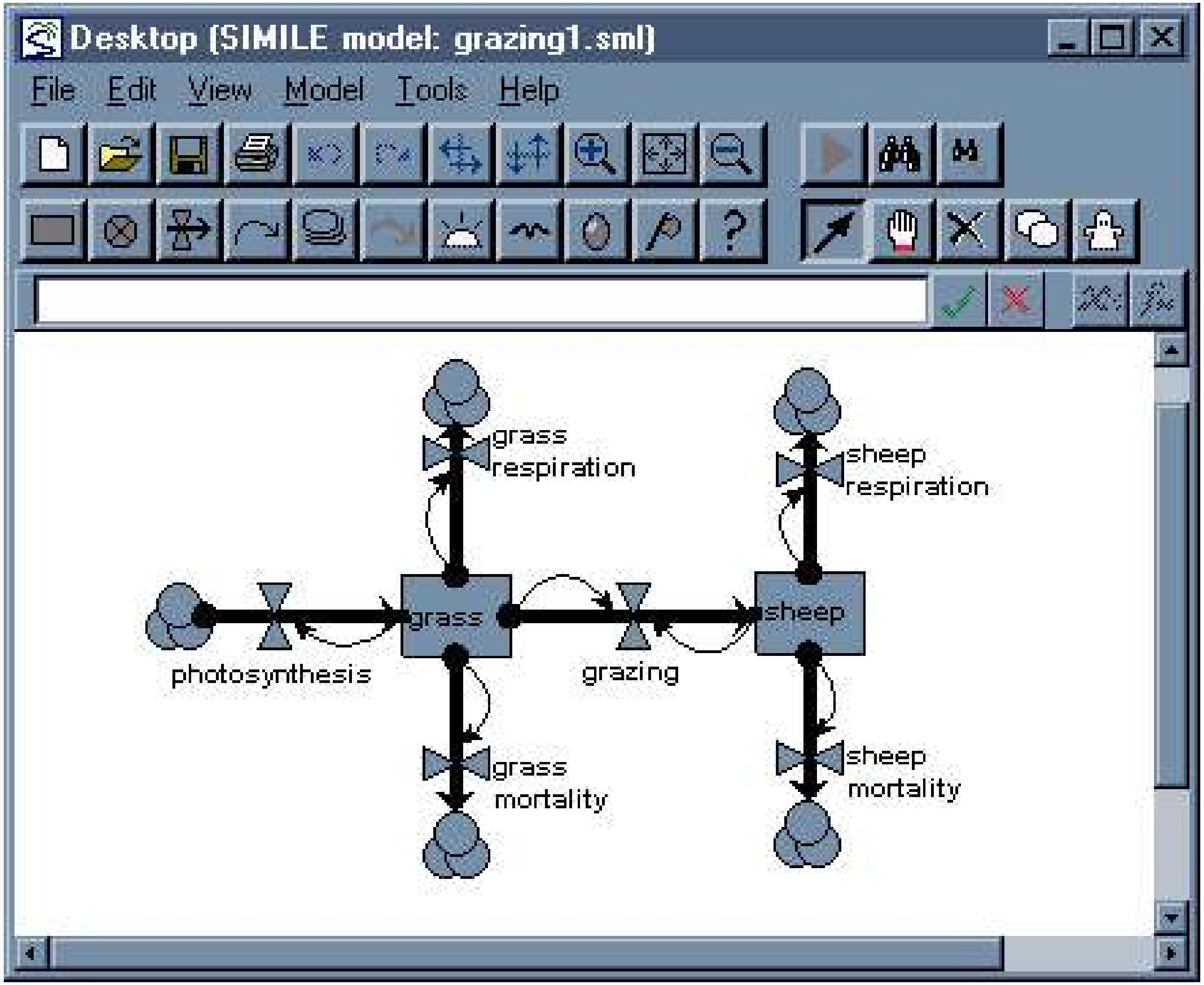


- says how a value for a variable is to be calculated
- often, represents the relationship between one quantity (Y), and one or more influencing quantities (X1, X2 etc)
- uses standard algebraic expressions
- may include a sketched graph function
- may include a tabulated function
- may include conditional elements
- in Simile, a single ‘equation window’ is used for all quantities (incl. parameters and initial compartment values)

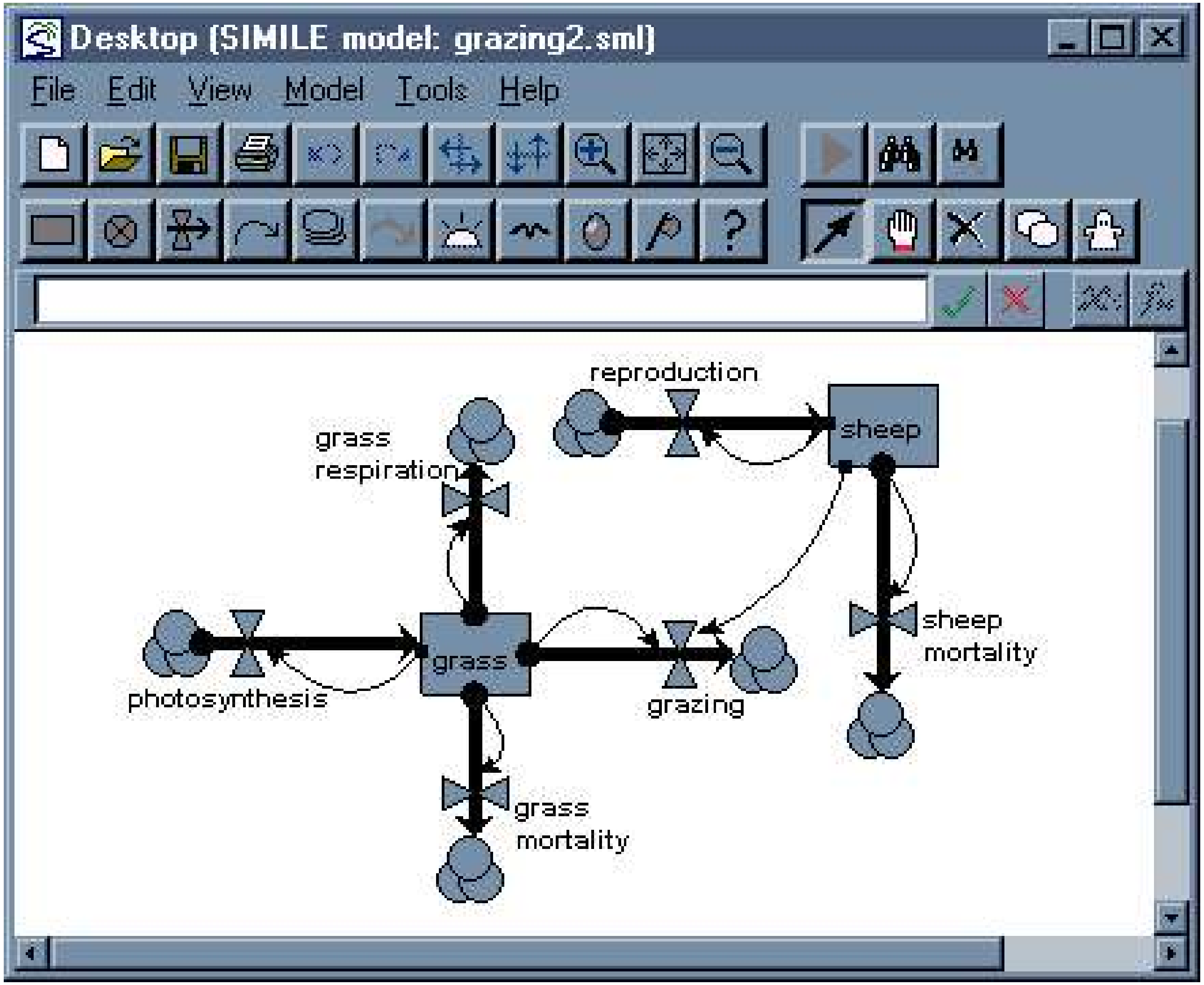
Compartments and flows

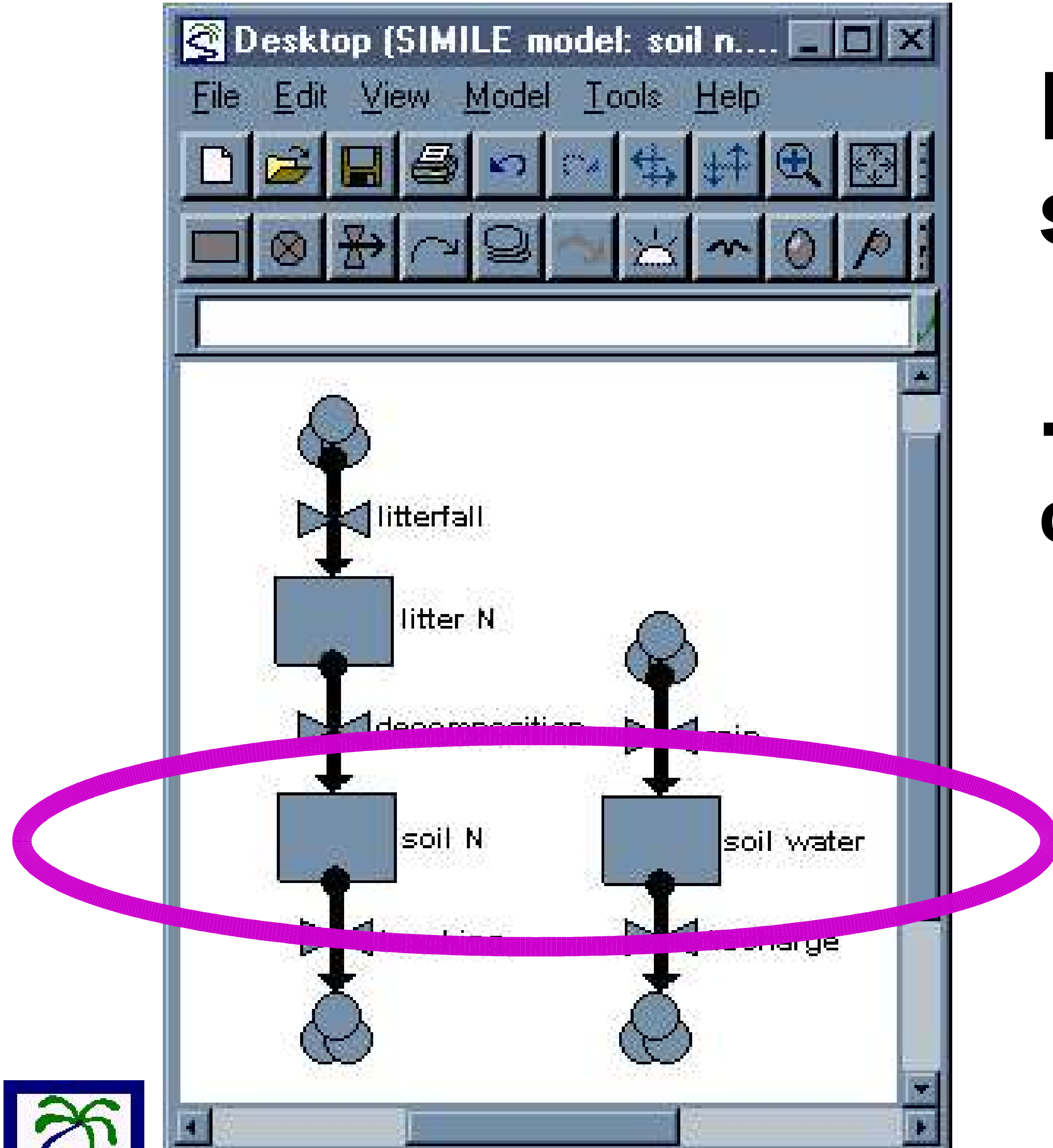
Object	Substance	Units	Flow	Units
grass	biomass	kg kg m^{-2}	growth, grazing	kg y^{-1} kg m^{-2} d^{-1}
grass	nitrogen	kg	uptake, grazing	kg y^{-1}
grass.roots	nitrogen	kg	uptake	kg y^{-1}
red deer			reproduction, mortality	y^{-1} ha^{-1} y^{-1}
tree	height	m	growth	m y^{-1}
bare land	area	ha	burning, re-colonisation	ha y^{-1}

Choice of substance (1) - biomass only



Choice of substance (2) - biomass and numbers





Multiple substances?

- use multiple compartments

Making your first model

- System Dynamics models (in Stella, ModelMaker…)
- ‘cycle’ diagrams (energy, nutrient, hydrological cycles)
- Differential/difference-equation models
- Design your own

Building models from 'cycle' diagrams

1. Find a 'cycle' diagram with stock and flow values

For each flow (F):

2. Choose a flow expression

$$F = k$$

$$F = k.C_1$$

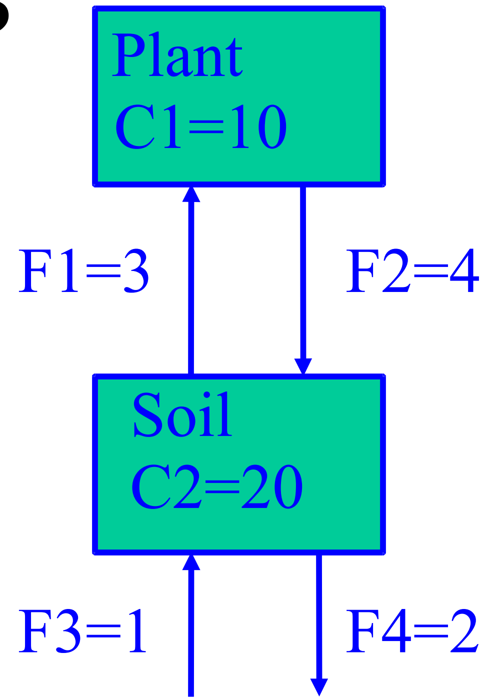
$$F = k.C_2$$

$$F = k.C_1,C_2$$

3. Re-arrange, as $k = f(F, C_1, C_2)$

4. Calculate k

5. Implement the model!



$F1 = k1.C2 \quad \therefore k1 = 3/20$

$F2 = k2.C1 \quad \therefore k2 = 4/10$

$F3 = k3$

$F4 = k4.C2 \quad \therefore k4 = 2/20$

Building models from differential equations

$$dX/dt = r.X - e.X.Y$$

$$dY/dt = c.e.X.Y - d.Y$$