

Ecological modelling with Simile

Lecture 2

Part A: Introduction to submodels

Part B: Submodels for modular modelling

Part C: Multiple-instance submodels

Robert Muetzelfeldt
Jasper Taylor
Jonathan Massheder

www.simulistics.com

Feb 2006



Part A

Introduction to submodels

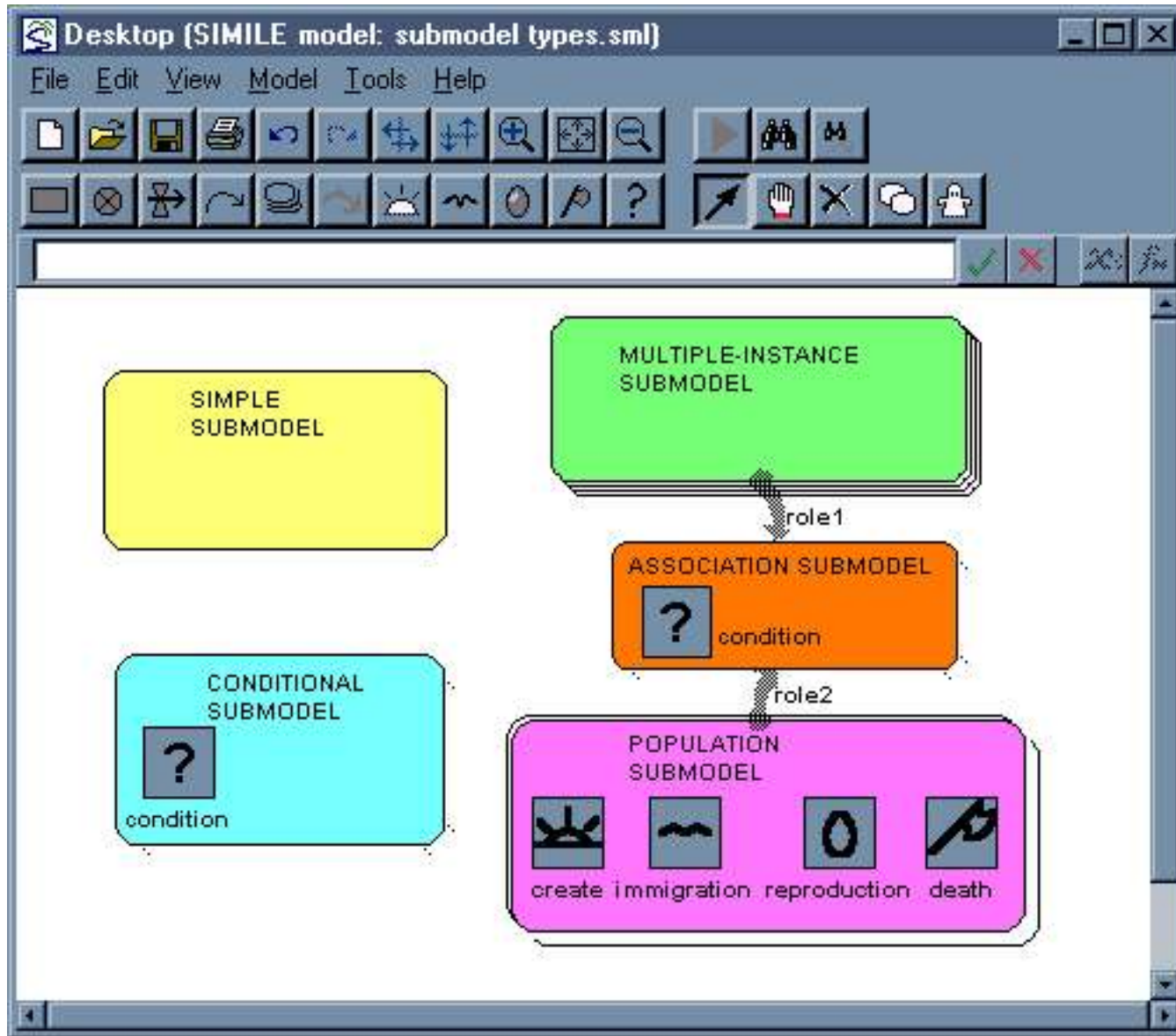


Submodels can be used for...

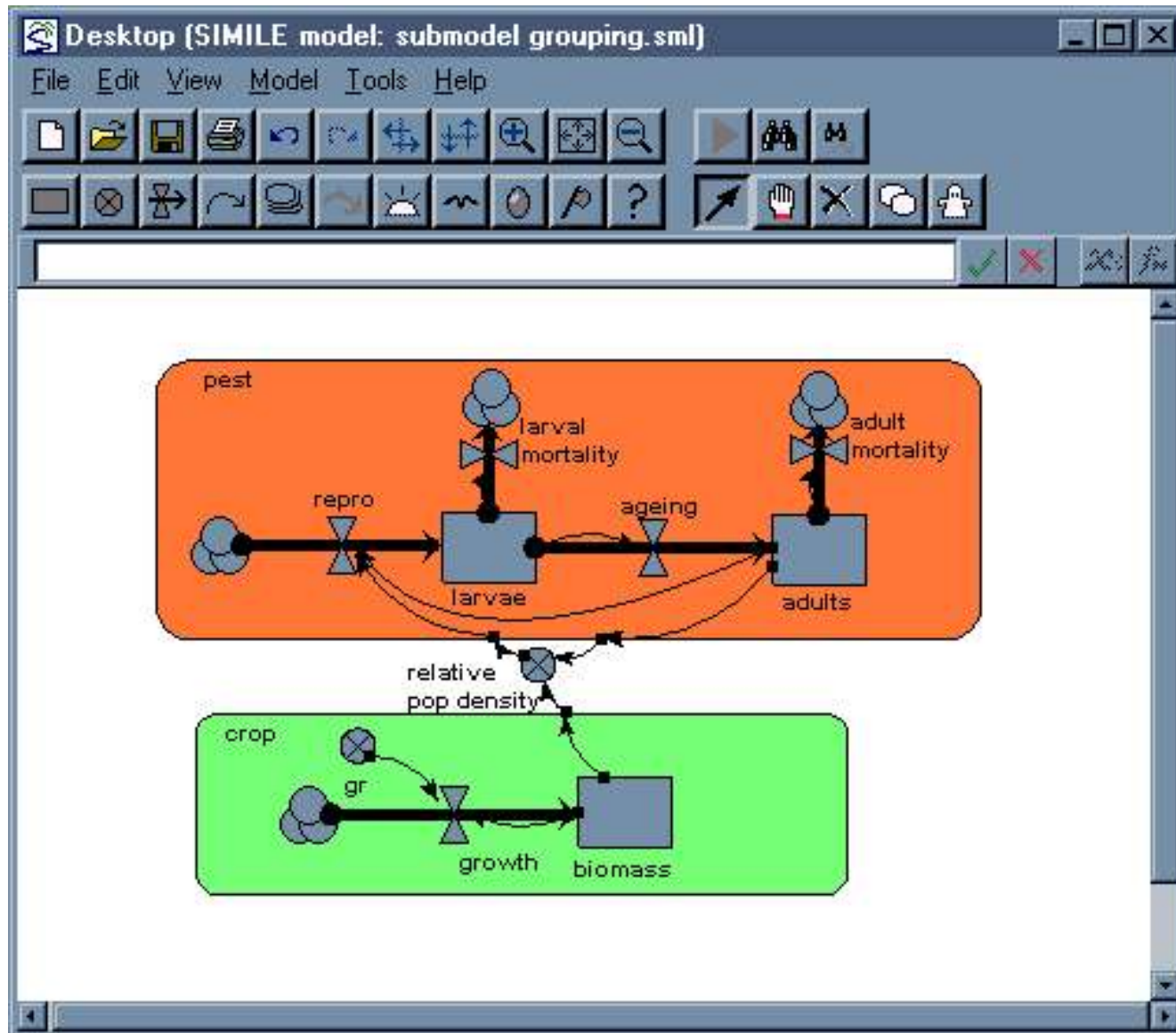
- visually separating part of the model
- controlling the appearance of a complex model
- enabling separate saving and loading
- moving parts of a model around
- creating multiple instances
- specifying associations between objects
- allowing parts of the model to exist conditionally
- specifying dynamically-varying populations
- specifying different time bases for different parts of the model



Types of submodel



Submodels for visual grouping



Part B

Submodels for modular modelling



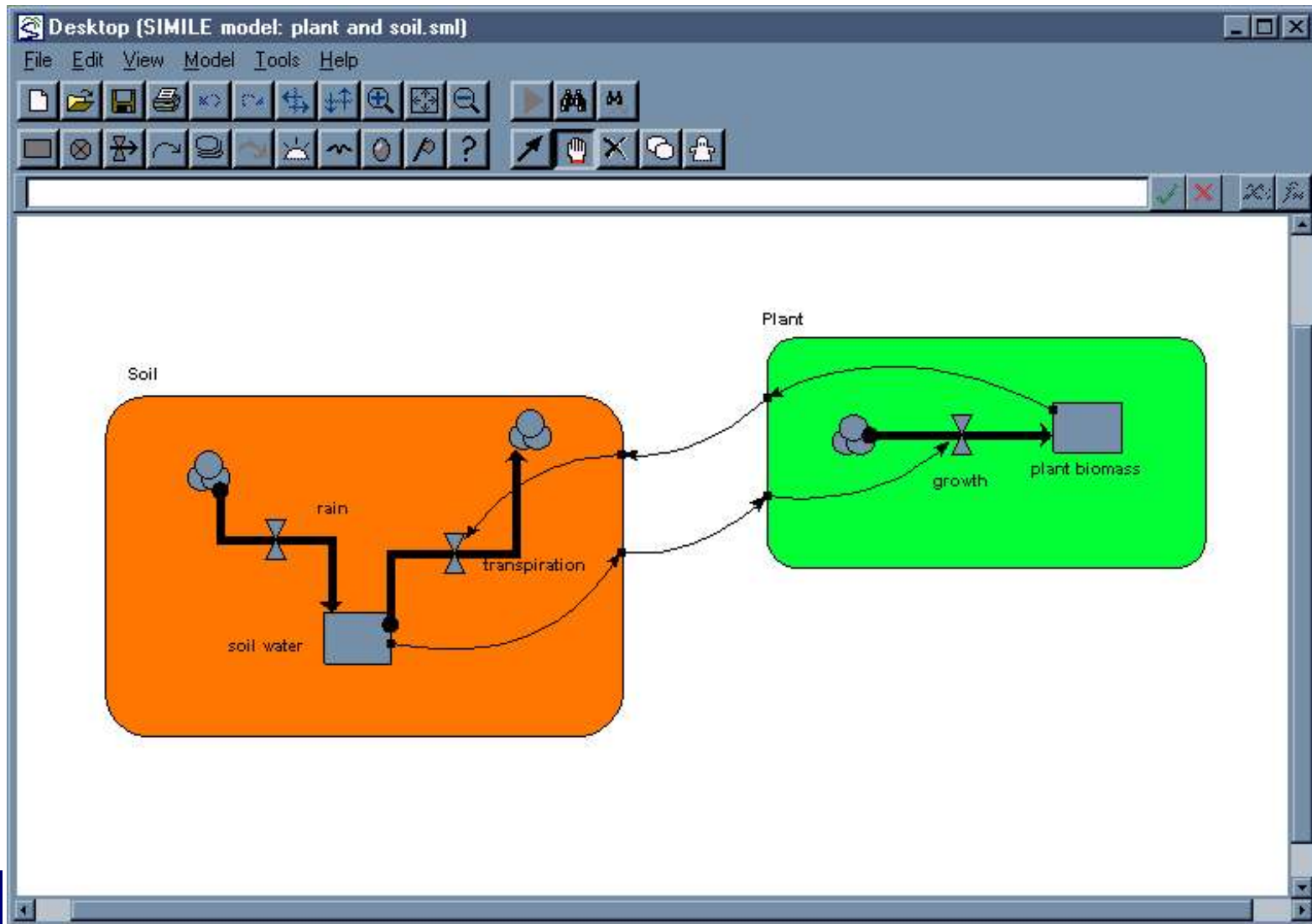
Submodels for modular modelling

- A whole model can be inserted into an ‘empty’ submodel
- The submodel interface spec allows automatic reconnection of links between submodel and the rest of the model (“plug and play” modularity)
- A submodel can be separately saved as a model in its own right (“unplug and play”)



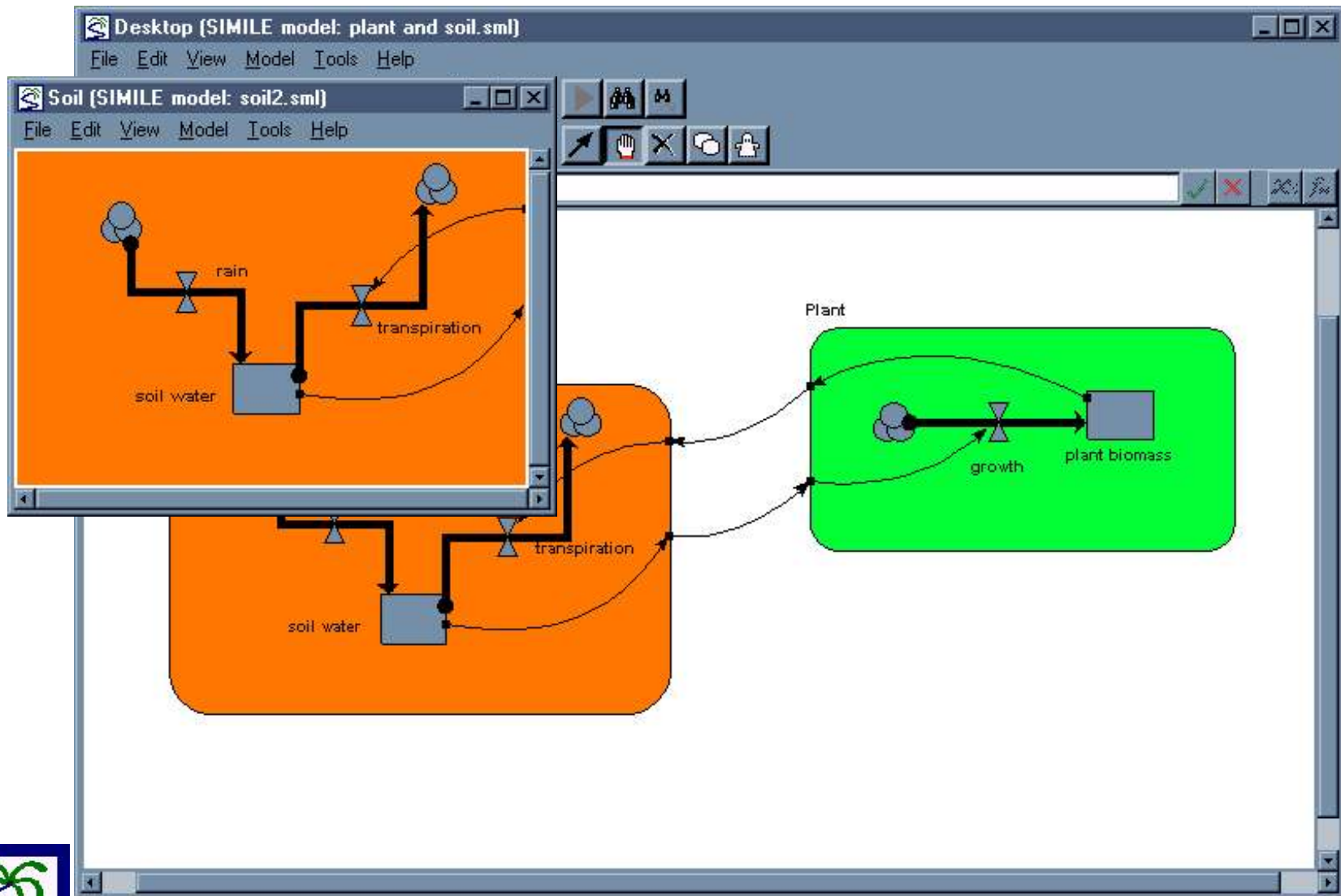
Modular modelling

Step 1. Original Simile model



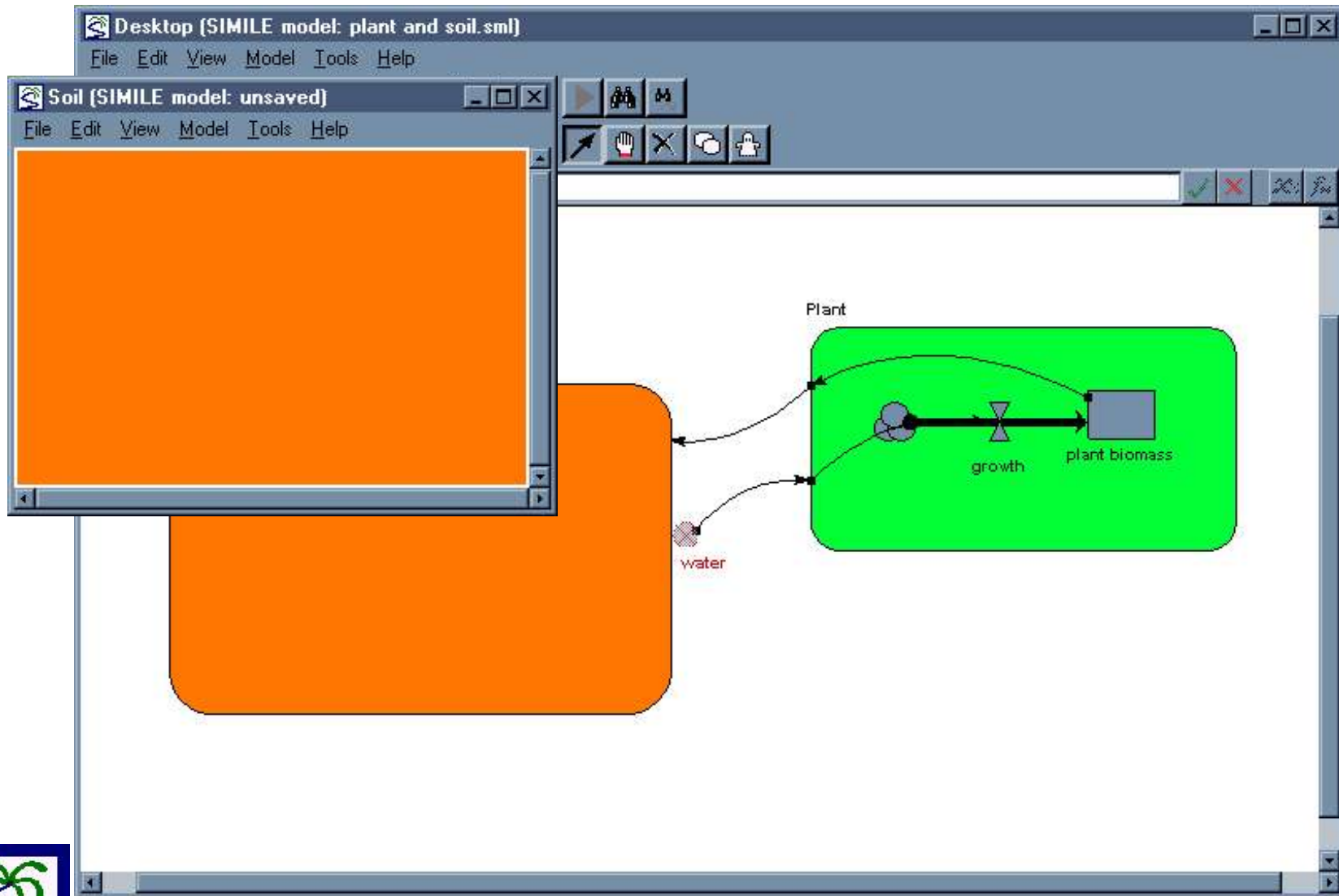
Modular modelling

Step 2. Submodel extracted



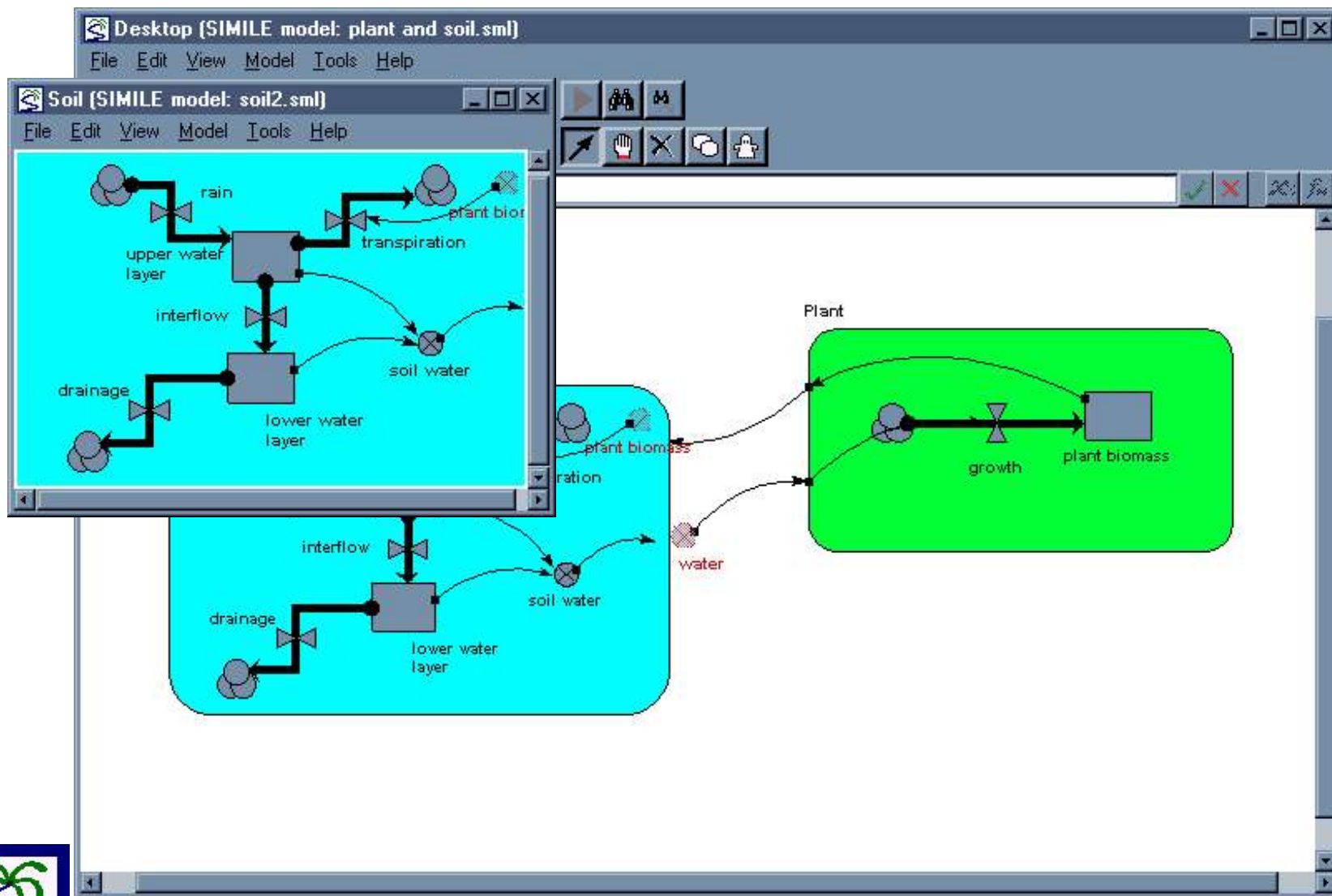
Modular modeling

Step 3. Submodel cleared



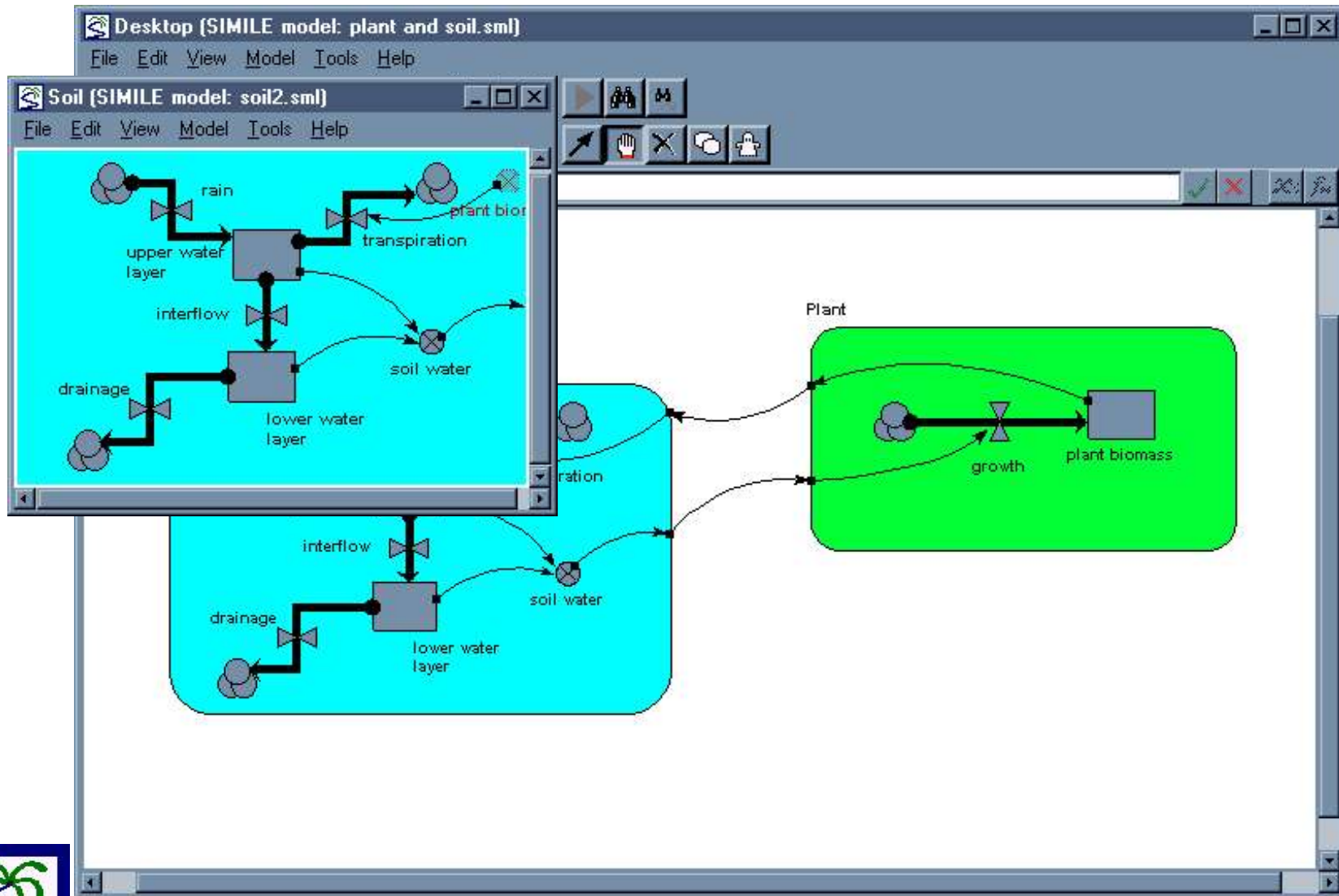
Modular modelling

Step 4. New submodel loaded



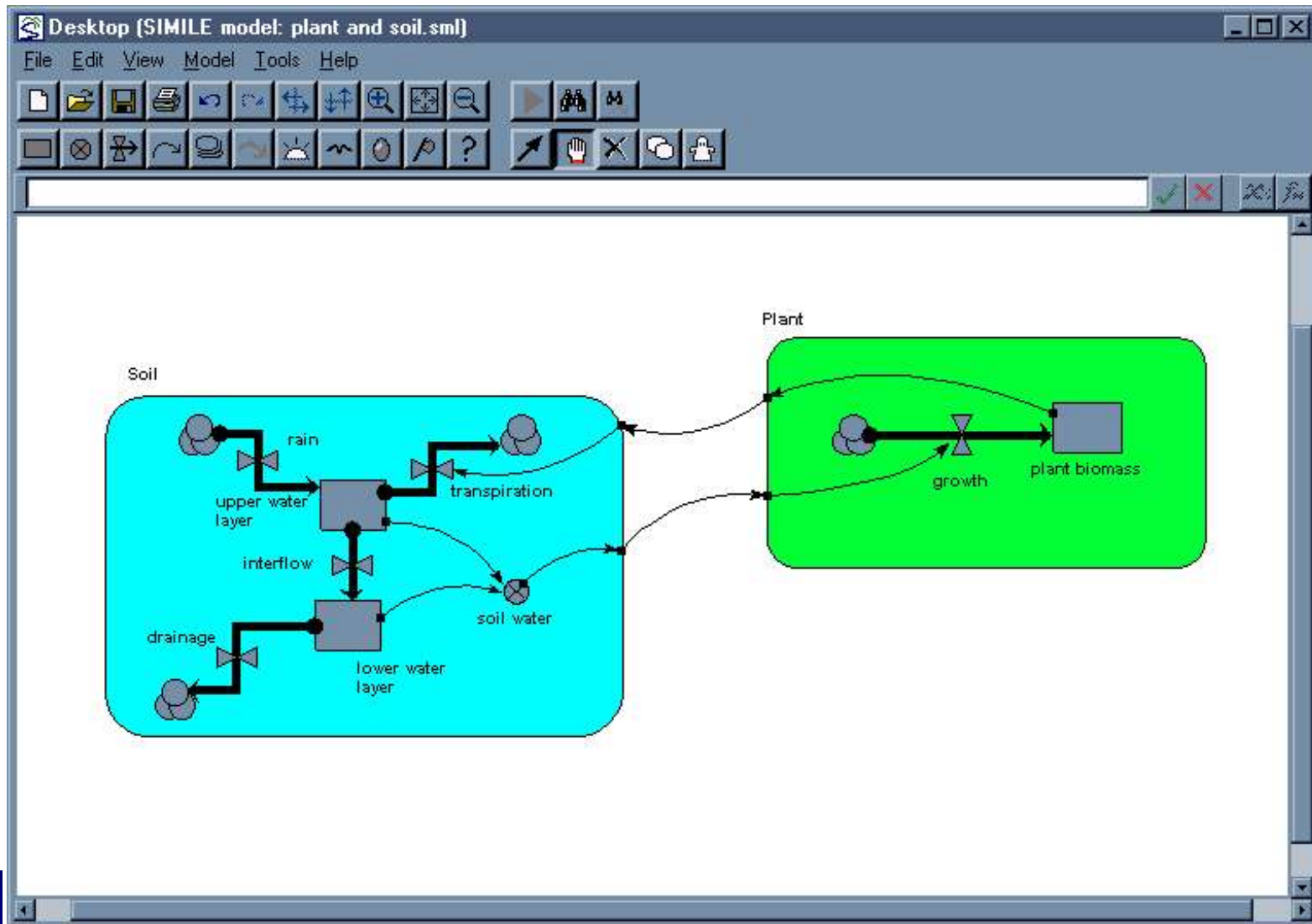
Modular modelling

Step 5. Submodel interface loaded



Modular modelling

Step 6. Completed model

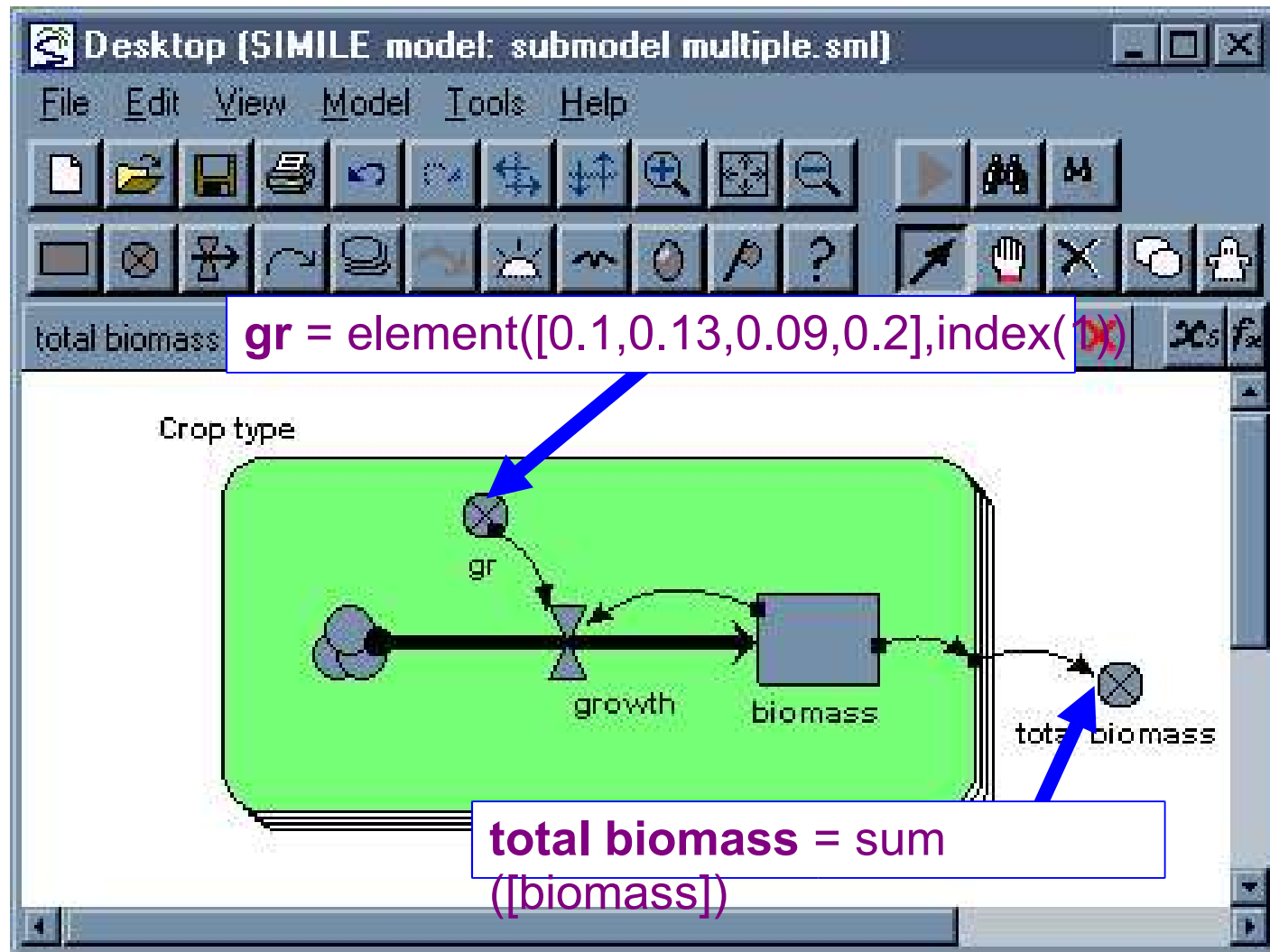


Part C

Multiple-instance submodels



Submodels for multiple things



Arrays and lists: notation, functions and operations

[biomass] denotes an array called biomass

[6,3,15] makes an array containing the values 6,3,15

{biomass} denotes a list called biomass

element([biomass],3) returns the value of the 3rd element of array
[biomass]

sum([biomass]) returns the sum of the elements in [biomass]

sum({biomass}) ditto

3*[biomass] returns an array with all the values in [biomass] multiplied by
3

[biomass]*[coef] returns an array with each element being the
product of the corresponding elements of [biomass] and [coef]

if [biomass]<5 then 3 else 6 returns an array of 3's and 6's,
depending on the corresponding value of [biomass]



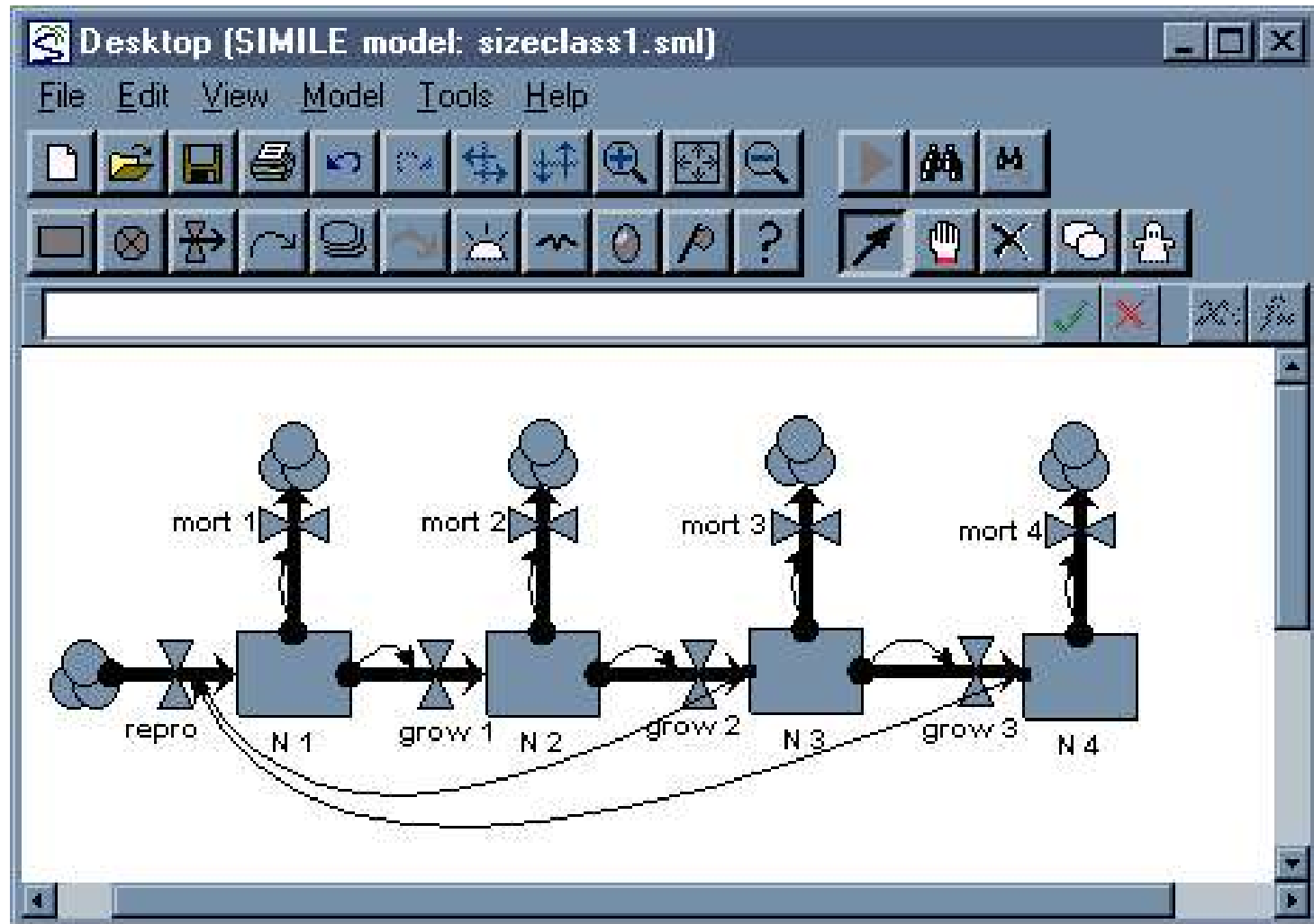
Using an array to handle the interactions between submodel instances

- The problem: how can each instance know the values of the other instances of the submodel?
- Examples:
 - age/size class population modelling
 - multi-layer soil water/nutrients model
- The method: have an array variable outside the submodel containing all the values for a variable inside the submodel



Size-class modelling (1)

- not using submodels



Size class modelling (2)

- using a submodel

