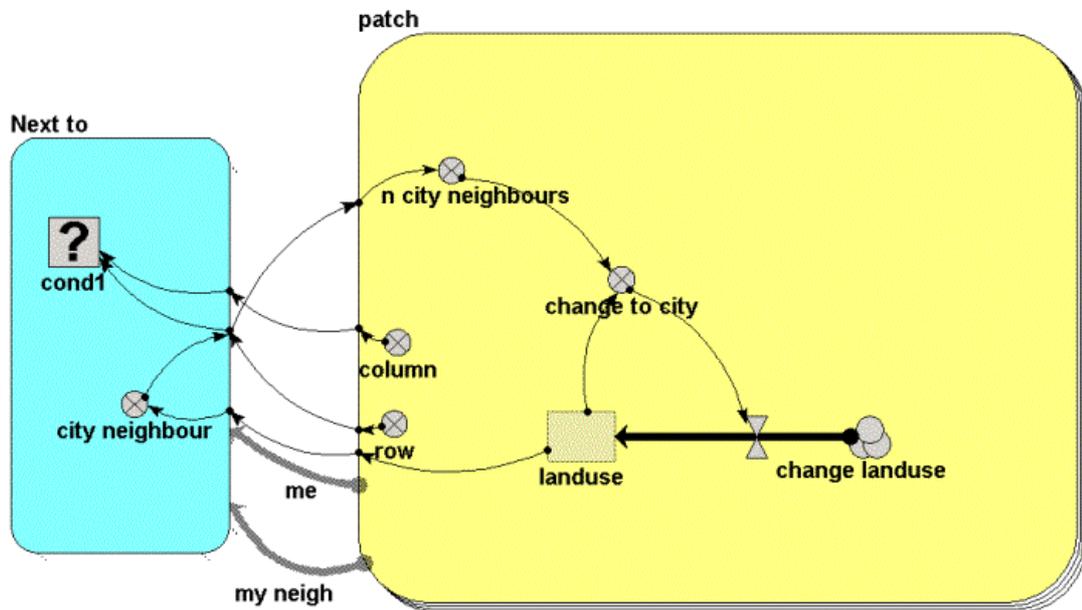


Simile: finally bridging the gap between temporal and spatial models?

Date: June 2004



By: Edu Boer



WAGENINGEN UNIVERSITY

WAGENINGEN UR

Simile: finally bridging the gap between temporal and spatial models?

Edu Boer
Wageningen, June 2004

Table of Contents

SIMILE: FINALLY BRIDGING THE GAP BETWEEN TEMPORAL AND SPATIAL MODELS?	2
1 JUSTIFICATION	5
2 SPATIAL MODELLING VS. TEMPORAL MODELLING	5
2.1 SPATIAL MODELLING	5
2.2 TEMPORAL MODELLING	5
2.3 INTEGRATION OF SPATIAL AND TEMPORAL MODELS	6
2.4 OBJECTIVES	6
2.5 METHODOLOGY	6
3 SIMILE VERSION 3.2	7
3.1 INTRODUCTION	7
3.2 SPATIALITY	7
3.3 SIMILE AND THE VECTOR-BASED MODEL	8
3.4 SIMILE AND THE RASTER-BASED MODEL	8
3.5 LIMITATIONS	8
4 ARISFLOW AT CGI	9
5 OTHER SOFTWARE OPTIONS	10
6 CONCLUSIONS	11
6.1 SIMILE	11
6.2 ARISFLOW	11
6.3 SIMILE VS. ARISFLOW	11
7 REFERENCES	12
ANNEX 1	13
7.1.1 <i>A simple Simile model explained</i>	13
ANNEX 2	17
7.1.2 <i>Modelling in Simile version 3.2</i>	17

Table of Figures

Fig. 1: Bank account model in Simile 7

1 Justification

As part of my UNIGIS study I have chosen to do a subject at CGI, the Centre for Geo Information of Wageningen UR. For the choice of a subject there should be a win-win situation: I should be able to experience a variety of activities in the actual use and implementation of GIS where CGI wanted to integrate results of the present study into their curriculum.

There is a visual modelling tool, called Simile, which recently developed integration with GIS. This is a very interesting development, but presently Simile is insufficiently known at CGI. Moreover, the tool could be integrated in the *Spatial Modelling* course of CGI when it proves to be valuable.

2 Spatial modelling vs. temporal modelling

Real-world situations in which modelling can play a role usually have two important dimensions:

- spatial
- temporal

2.1 Spatial modelling

GIS's are powerful tools to build *spatial models*. A GIS can effectuate many types of spatial analysis. Overlaying and buffering are just two important examples. Modelling change over time is more complicated in a GIS, although software development is still continuing in this direction.

Visual modelling tools may increase the efficiency in building spatial models. The *ModelBuilder* has been integrated in ArcGIS version 9 early 2004. This tool, however, can only manage ArcGIS operations in a macro type of actions. ArisFlow is a tool used at CGI to design and manage workflow in a GIS environment. This tool can integrate different data types and actions from a number of programs.

2.2 Temporal modelling

On the other hand we have *temporal models*. An important aspect of these types of models is that they are reiterative: the previous state of a variable (e.g. river level) is used to calculate the current state, which may be repeated for a number of time units (e.g. river level in 24 consecutive months). In a formula this will look like this:

$$\text{Current state} = \text{previous state} + \text{change}$$

The change is then calculated for every time step by the model; a GIS cannot use the previous state and the change (probably based on different values of the involved parameters) in a reiterative process. A GIS could only be used in temporal modelling for 'post-processing' or visualisation.

Temporal models, which focus on changes over time, are used in a variety of ways. Generally they are used in scenario studies: what will happen in the future when certain measures are taken (or when no measures are taken). An example is the change of river level influenced by rainfall, water storage capacity, vegetation, etc. Other examples of these models include crop yield models, ecological models and urban expansion models.

There are many software tools to create temporal models. Increasing in importance are the *visual* modelling tools. The models created in this way look like flow diagrams

or flowcharts. Visual modelling tools do not work with writing lines of text and it is not necessary to be able to program or to know at least one programming language. This will help bridge the gap between the model developer and the model user, traditionally different people who had to rely on each other competences. Examples of visual modelling tools are Simile, Stella, ModelMaker and PowerSim.

2.3 Integration of spatial and temporal models

The concept of both models is diverging, and, more seriously tended not to be able to integrate the other concept. GIS and spatial models are time-fixed: the data they are dealing with are a snapshot in time. In vector-based GIS time has been integrated as an additional attribute of spatial features, but little or no modelling can result from this. In raster-based GIS the time-fix could be overcome e.g. by using cellular automata; an example is a fire-spreading model. This, however, requires additional programming for that particular problem.

Temporal models are often geometry-fixed and applicable to a certain area with a particular characteristic and not for a heterogeneous area.

Integration of spatial and temporal models is desired for instance in the case of landuse modelling, but difficult as explained above. There is now a software package, Simile, for temporal modelling which claims to have a working link to GIS. This would be of interest for CGI to avoid time-consuming and expensive dedicated programming. Moreover, for educational purposes it would be highly appreciated if programming, including that for cellular automata, could be avoided.

2.4 Objectives

The objectives of the present study are:

- To assess the modelling tool Simile (version 3.2), particularly with regard to its interoperability with GIS packages;
- To assess the dataflow management tool ArisFlow currently in use at CGI;
- To advise CGI on the use of both packages.

2.5 Methodology

To assess a software package one has to become acquainted with it. The Simile *Help* function as well as its website will be consulted. A number of articles have been written on the software and a search of Internet resources will be undertaken. Existing Simile models will be studied and people have modelling experience in Simile will be consulted.

A spatial model will be built in Simile and the GIS link will be made operational.

To become acquainted with ArisFlow the online CGI course *Geo-Information Tools* will be studied.

These experiences will be essential to assess the strengths and weaknesses of both packages and how these can enhance the use of GIS at CGI.

3 Simile version 3.2

3.1 Introduction

Simile is a visual modelling tool for ecological, biological and environmental research. It can model all kinds of processes from a simple 'bank-account-with-interest' model to a complex land use change models. Simile can handle the following model types:

- system dynamics
- differential / difference equation modelling
- disaggregation
- matrix models
- individual / object-based modelling
- object-oriented modelling
- spatial modelling
- modular modelling
- population dynamics

Moreover, it can combine these types of models into a single model. When running a model it is built in C++ or Tcl, two programming languages which transform the model to a computer-executable form. Then the model can be run for the number of time steps the users specifies (see Figure 1.).

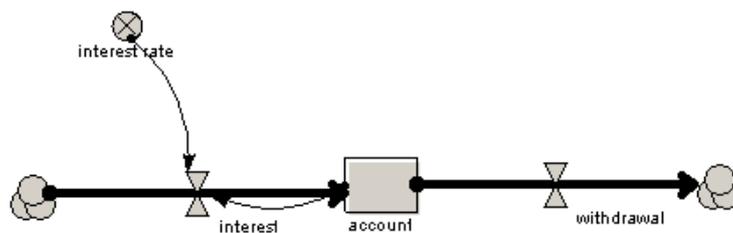


Fig. 1: Bank account model in Simile

3.2 Spatiality

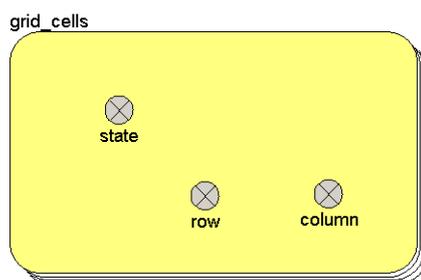


Fig. 1: Figure 2. A simple grid model in Simile.

For us it is interesting to examine the spatial models Simile can build. In fact it is a particular form of disaggregation, in which an area is divided into a number of similar units: typically grid squares or polygons. A spatial unit is defined as a single *instance* in a *submodel* in Simile with its attributes. Then it is defined that the submodel has many instances (i.e. one for every spatial unit, in Simile jargon a *multiple-instance submodel*). Each instance (read: each grid cell or polygon) should have a type of identifier. In the case of grid cells each instance has an x and a y (or row and column) coordinate

(see figure 2). The product of x and y is the total number of instances. In the case of polygons each polygon will have a unique number and the number of polygons should equal the number of instances in Simile.

3.3 Simile and the vector-based model

Early 2004 Simile designed a link to ArcMap to be able to visualise a running model in this GIS environment. The attributes of polygons can be seen changing while the model is running, but as a matter of fact the polygons themselves are not changing. For crop-yield models this doesn't pose a problem, as it will not be necessary to change the field boundaries as described by the polygons. As the number of polygons and their shape cannot be changed in or by the Simile model this seriously limits the modelling possibilities in a vector-based model.

Simarc is a Simile-ArcView plug-in, a tool to link Simile directly to a GIS. For input in Simile it can read the data directly from the database file in ArcView for initialising the Simile model. Simarc joins for every simulation step the generated data directly into the ArcView database file. Although data in- and output is better integrated than in the ArcMap link

Simile itself can also display a running vector-based model. The x-y coordinates of the (static) polygons in the model can be drawn from another submodel or may be read from an external file.

3.4 Simile and the raster-based model

In a grid environment the number of grid cells doesn't change and as such Simile is a much more powerful tool. It is for instance possible to model neighbourhood relations. This can be achieved by including a conditional submodel (see Annex 1, where a simple Simile model is explained). The main disadvantage of this in Simile is that the number of operations is the product of the number of instances in the spatial submodel: based on a 10×10 grid, which is extremely small in GIS applications, already 10,000 calculations for neighbourhood analysis need to be performed. It is stated on the Simile website (www.simulistics.com) that the version 4 will perform here better, i.e. it will perform relative to the number of grid cells and not to the square of the number of grid cells.

A (raster-based) GIS has many spatial analysis tools. Simile doesn't have these built-in, but it is expected that these may be modelled in Simile as well. Separate layers can easily be combined in a single submodel by assigning more than one value to a certain grid cell and operations on these values can also be included. In figure 3 there are 2 'layers', landuse and soil and a 'layer' derived from these, forest on clay.

Simile again has the possibility to display a running raster-based model. To do so the user only has to indicate what variable represents the columns (see Annex 1).

3.5 Limitations

Modelling using Simile in a vector-based model is limited to the change of attributes; polygons cannot be changed. The link with ArcMap is therefore also of limited use.

Raster-based models in Simile hold promise, but the calculations at present are too slow. There is little experience with (spatial) modelling using Simile and therefore ready-to-use models for certain operations (e.g. buffering) are not readily available.

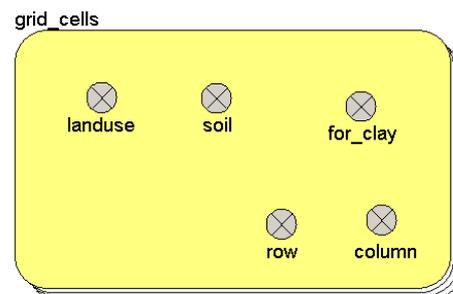


Fig. 2: Figure 3. A Simile model containing more layers

4 ArisFlow at CGI

The ArisFlow programme is used at the Centre for Geo-Information (CGI) at Wageningen University to better manage GIS research. ArisFlow is a dataflow management tool for managing automated processes. The ArisFlow flowchart is a graphic representation of the data managed, the processes acting upon these data and how these data and actions are connected.

From this description it can be seen that it is not a modelling tool, but it is used as a tool to keep track of the generally complex nature of GIS operations. It can also be used for automated GIS operations (e.g. clipping data sets to the study area, testing different alternatives, etc.).

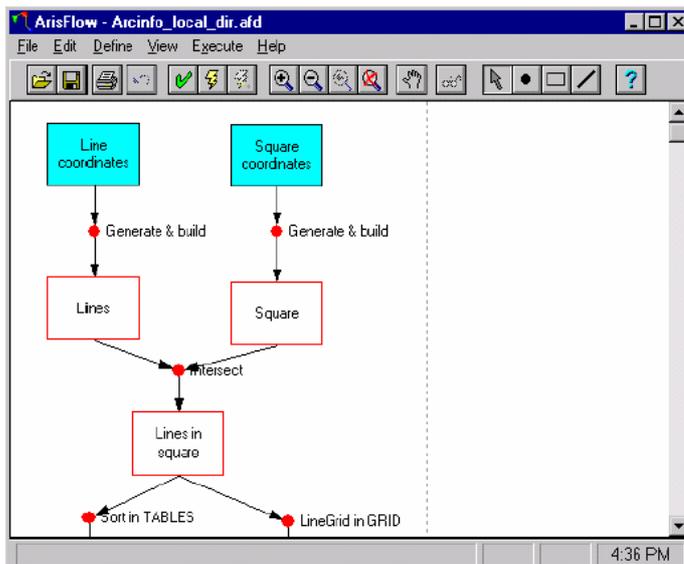


Fig. 3: Figure 4. Example of ArisFlow flowchart.

The data in the data-action model in ArisFlow can be of a multitude of types, which should be indicated in the model as well as the location of the data. The actions can be I/O executables, DDE's (dynamic data exchange) or in the form of macros in a client-server environment.

ArisFlow Commander can be used to run the ArisFlow model with e.g. different input data or different parameters, which is often used in scenario type of studies.

Loops and cyclic processes like those in temporal models cannot be modelled in ArisFlow.

5 Other software options

ArcGIS 9 has now an integrated modelling tool, which is the *ModelBuilder*. Its capabilities for temporal modelling were assessed. It was slightly disappointing to realize that the *ModelBuilder* is merely a tool for automating repetitive tasks and cannot be used for temporal model building. The *ModelBuilder* is a tool very similar to the *Macro Modeler* in IDRISI, which was already fully integrated by the beginning of 2002.

PCRasterGIS claims to be a high level integration of a GIS and a modelling environment, but time didn't allow me to assess its assets. See for more information: <http://pcraster.geog.uu.nl/>.

6 Conclusions

6.1 *Simile*

A visual modelling environment tool is easy to use, as it doesn't require programming skills. Moreover, as the design is in one hand the user can more directly interact with the model. *Simile* is such a tool, its use for people without prior modelling skills needs some practice of how to model simple systems. However, in my experience this can be acquired in a reasonable short time.

Simile does not perform satisfactorily in a vector-based model, as the spatial model (the polygons which serve as input data) cannot be changed.

Simile holds promise for raster-based models where in principle *all spatial analyses* commonly executed in a GIS environment can be executed. The '*Simile* toolbox', however, is not well equipped as yet. Another drawback at this moment is the fact that models involving larger grids take a long time to be run.

6.2 *ArisFlow*

The flowcharts made with *ArisFlow* serve the objective of accountability and repeatability. *ArisFlow* is more powerful than visual modelling tools as integrated in e.g. ArcGIS 9 and IDRISI, as it can work with more data type (e.g. databases, spreadsheets) and it keeps better track of changing or changed model elements by providing these with a 'time stamp'.

Again, users need to get accustomed to this software, which may not pay off for small projects or basic (spatial) analyses; the visual modelling tools integrated in a GIS may then be more efficient to use.

6.3 *Simile vs. ArisFlow*

The two tools serve a different purpose where *Simile* has a much wider array of applications. It is not excluded that *Simile* evolves in such a way that it can make *ArisFlow* superfluous by including the data flow (as now managed by *ArisFlow*) into the *Simile* model.

For the moment *Simile* models can be run as one of the 'actions' performed in *ArisFlow* either as spatial models or as non-spatial models.

7 References

Consulted websites:

PCRaster. <http://pcraster.geog.uu.nl/> Accessed: June 2004.

Annex 1

7.1.1 A simple Simile model explained

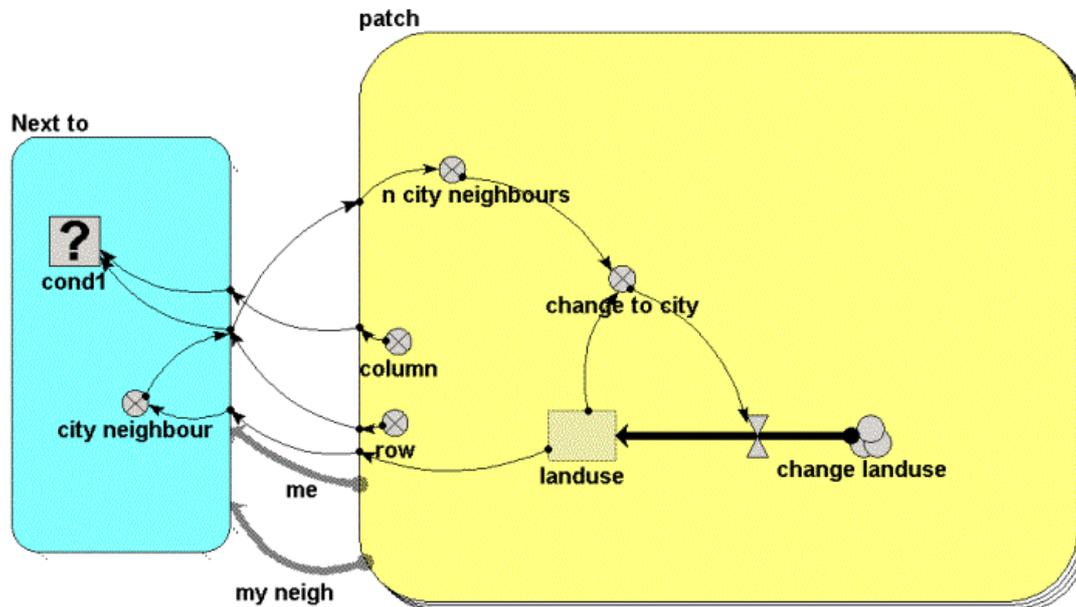


Fig. 4: Figure 5. A city expansion model.

A Simile model has been designed to show hypothetical landuse change, where growth of cities is shown at the expense of agricultural land. Only large cities can expand; waterbodies will remain unchanged.

The model is implemented in a 10×10 grid (the yellow-coloured **patch submodel** has 100 instances). It can be thought of as a stack of 100 cards (each representing a grid cell) which all have values for *variables*, *compartments* and *flows*, but these values can change from one card to the next. The starting values of **landuse** are read from a table. A square represents a *compartment* in Simile.

The *variables* row and column define the row and column values of the grid in such a way that each instance in the model has a unique row-column pair.

Two *relation arrows* connect the **patch submodel** with the **Next to submodel**, here we see that for neighbourhood relations the necessary calculations relates to the square of the number of grids. (The 100 grid cells of *relation me* are all compared to those of **my neighbour**).

The **Next to conditional submodel** (in blue) is only executed when the condition is met, in this case for directly neighbouring pixels (maximum 8). Only then the *variable city neighbour* exists.

n city neighbours gives the total of city neighbours of each pixel; when this is more than 2 **change to city** is set to 1 for agricultural pixels with a probability of 20%. This models that along the city margins agricultural land may (about 20% of the cases) be transferred to city. Water bodies are not affected, small cities (2 x 2 pixels

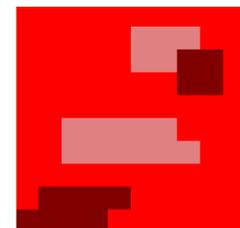


Fig. 5: city

Fig. 8: Figure 6. Initial situation.

or less) will not expand. When the *variable change to city* equals 1, the **change landuse flow** will be 1, thus changing the **landuse** from 1 (=agriculture) into 2 (=city).

Running the model several times for the same time span will yield different results as the randomised chance of 20% to change will work out differently in 2 consecutive runs.

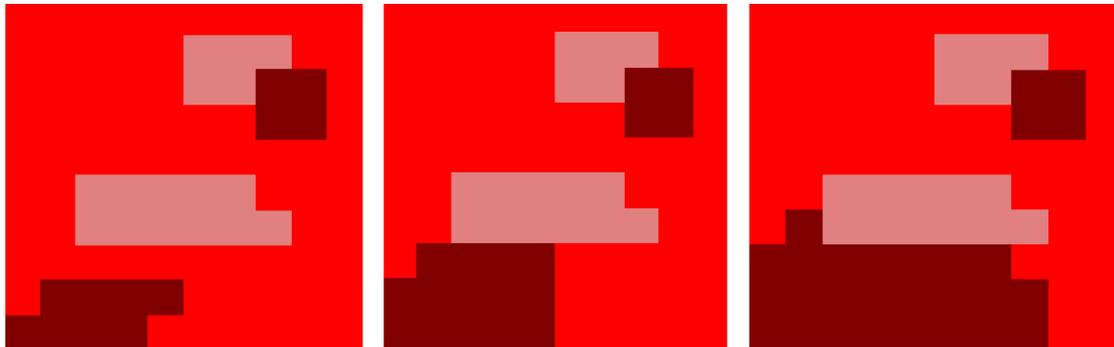


Fig. 9: Figure 7. Spatial grid display after 0, 10 and 50 times running the model.

The total area of landuse = city could be easily integrated into the model by a variable counting the number of instances (read: pixels) with landuse = city. In this way city expansion is easy to monitor through time (e.g. by means of a graph). In the example shown in Figure 7 after 50 time steps 16 pixels have changed landuse.

MODEL ENTITIES	FORMULA	DESCRIPTION
----------------	---------	-------------

compartment

landuse	-	holds 3 classes
----------------	---	-----------------

variables

column	$\text{fmod}(\text{index}(1)-1,10)+1$	yields 10 times the values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
row	$\text{floor}((\text{index}(1)-1)/10)+1$	yields 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, etc
city neighbour	if landuse_me==1 then 1 else 0	sets all non-city land to 0
N city neighbours	$\text{sum}(\{\text{city_neighbour_my_neighbour}\})$	sums the number of neighbouring pixels with landuse = city
change to city	if landuse==2 and $\text{rand_var}(0,1)<0.2$ and $\text{n_city_neighbours}>2$ then 1 else 0	20% change that an agricultural pixel (landuse = 2) will change having at least 3 neighbouring city pixels

flow

change landuse	if change_to_city==1 then $-1/\text{dt}(1)$ else 0	when variable change to city = 1 then landuse agriculture will become city
-----------------------	--	--

condition

cond1	! (column_me==column_my_neighbour and row_me == row_my_neighbour) and abs (column_me- column_my_neighbour) <1.5 and abs(row_me-row_my_neighbour)<1.5	only true for direct neighbours
--------------	--	---------------------------------

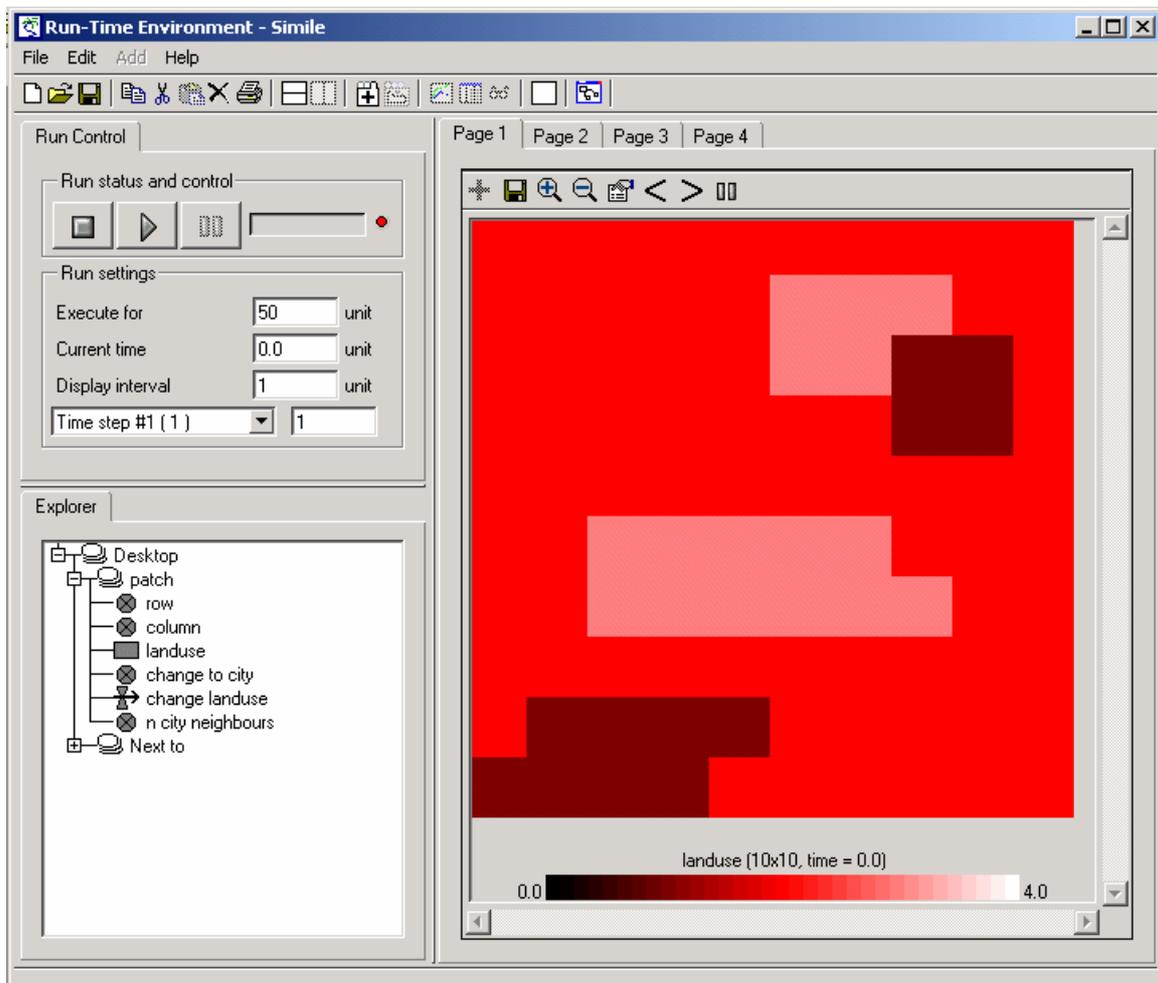


Fig. 10: Figure 7. Screenshot of the grid display before running the model.

As can be seen in figure 7 there are more possibilities (page 1 – 4) to view the results of a running model, e.g. as a table, as a graph of the different variables vs. time. In the table one can see for instance at what time exactly a certain pixel changed from agriculture to city. Other variables like **change to city** could also be viewed as grid as every grid cell has a value for this variable.

Annex 2

7.1.2 Modelling in Simile version 3.2

The component toolbar in Simile looks like this:



These are the components with which to build models. The different components will be briefly explained hereafter.

compartment

It represents a quantitative state (e.g. liters of water, tree height) and might be thought of as a container with some substance. Flows can go in and out to change its value.

Do's

- ✓ can go negative
- ✓ can have flows *to* it and *from* it

Don'ts

- cannot have influence arrows *to* it (unless initializing)
- cannot hold other substance (in other units) when connected by a flow to another compartment

variable

Holds a constant or variable quantity. The value or values come from a mathematical expression. The expression may simply be a number, or it may be a complex mathematical expression involving various variables, operators (such as + and -), functions (such as log or square root), and conditional elements. The value of a variable may vary during the course of a simulation, if it is calculated from other parts of the model that change over time, or it may be constant.

Do's

- ✓ can have influence arrows *to* it and *from* it
- ✓ can hold more values

Don'ts

- cannot have flows to it or from it

flow

A process moving a substance between compartments. The net rate of change of a compartment is the sum of all the inflow values minus the sum of all the outflow values. In most ways it is treated just like a variable and can hold mathematical functions and conditional elements.

<p>Do's</p> <ul style="list-style-type: none"> √ should have the same units as the connected compartment √ can have influence arrows <i>to</i> it and <i>from</i> it (and to each other) 	<p>Don'ts</p> <ul style="list-style-type: none"> – cannot be connected to anything but compartments
--	--

influence arrow

Represents the fact that one quantity is used to calculate another. When using a variable to calculate a flow, there needs to be an influence arrow from the variable to the flow.

<p>Do's</p> <ul style="list-style-type: none"> √ can be drawn to and from variables and flows √ direction is important √ can only be drawn to a compartment to initialize (give it a first value), after that the compartment is calculated by the net change of in- and out-flows 	<p>Don'ts</p> <ul style="list-style-type: none"> – cannot be drawn to other influence arrows – cannot be drawn to a submodel – cannot be drawn to a role arrow
---	---

submodel

A group of model elements, representing an object. It is used for a variety of purposes, like making the model visually better understandable (separating plant and animal influences for example), or to be able to save and run the submodel separately or swap it with other submodels. For GIS it is important to be able to create multiple-instance submodels disaggregating space into grid squares, polygons, or some other form of spatial unit. A grid square is then one instance; each instance has the same model but may have different initial values or variables.

<p>Do's</p> <ul style="list-style-type: none"> √ use submodels to compile complicated models for better control √ a multiple-instance submodel can be recognized by the multiple lines on the right- and bottom edges, like a stack of cards 	<p>Don'ts</p> <ul style="list-style-type: none"> – flows and influence arrows cannot be drawn to or from a submodel – by preference do not draw a submodel after you have indicated its elements
--	--

role arrow

Represents the fact that one object is associated with another, with each playing a role in the association. They can be drawn between 2 submodels, once or twice to indicate different associations.

<p>Do's</p> <ul style="list-style-type: none"> √ draw role arrows only between submodels 	<p>Don'ts</p> <ul style="list-style-type: none"> – never draw more than 2 role arrows between 2 submodels
---	--

? Condition

<p>Represents the fact that an instance of a submodel can exist only under specific conditions. It can take the form of a mathematical expression and returns a 'true' or 'false' value.</p>	
<p>Do's</p> <ul style="list-style-type: none"> √ can only be used <i>inside</i> a submodel 	<p>Don'ts</p> <ul style="list-style-type: none"> – include more than 1 condition in a submodel (but conditions can have multiple parameters)

The following buttons in the toolbar only refer to population submodels where the number of instances (individuals) can change when running the model:

 = initialization,  = migration,  = reproduction and  = extermination.

Multiple-instance submodels

A complex term, but of particular value when using Simile as a modeling tool for GIS: as indicated briefly above a multiple-instance submodel can be used for grid squares or polygons, each 'instance' corresponding to a grid cell or a polygon. The total number of instances cannot change, but their values can. One can define that variable (or compartments) of a submodel instance (corresponding with an attribute of a grid cell or polygon) change over time (e.g. the age of a forest stand, urban growth).