How to use Lagom generiC

Steffen Fürst, Antoine Mandel, Wiebke Lass, Frank Meissner, Carlo Jaeger

April 1, 2009

Contents

1 About this Text

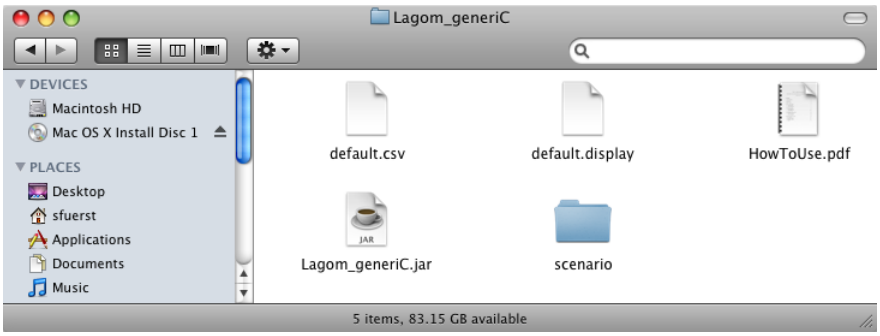
The purpose of this text is to introduce you to the graphical user interface of Lagom generiC . This is done in a tutorial style, you don't have a reference manual in your hand. Still, the program is much simpler than word processors etc., so this tutorial should suffice to get you started.

The text doesn't describe the model itself. You can find such a description in the ECF working paper "Lagom generiC: an agent-based model of growing economies"[1], available online under http://ecf.pik-potsdam.de/News/ecf-working-paper-1-2009.

The graphical user interface is based on the MASON java library, small parts of this description are taken from their Tutorial 0. Let us take here the opportunity to thank the MASON team for making their software publicly available.

2 Installation

After you have download and decompressed the zip–archive, you should have a folder that looks like this[2]:



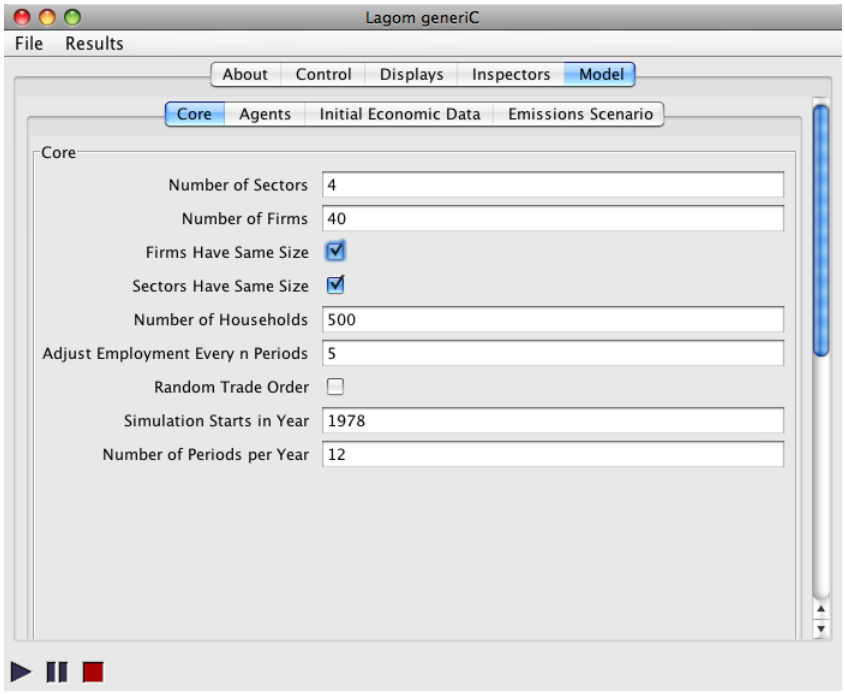
For the moment, only the file `Lagom_generiC.jar` is of interest to us. The usage of `default.csv` and `scenario` are explained in Chapter 4.2 *Parameter Files*, this of `default.display` in Chapter 4.3 *The Displays Panel* and the pdf–file `HowToUse.pdf` that is the document you are currently reading.

3 First Contact

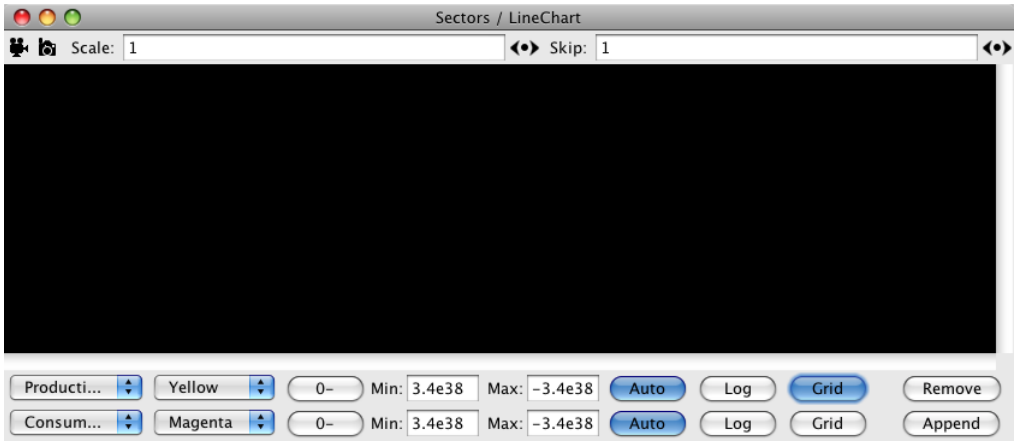
Now let's launch the Lagom generiC application by double–clicking on the `Lagom_generiC.jar` icon. Two windows should appear:

[1] Mandel, A., Fürst, S., Lass, W., Meissner, F. and Jaeger, C. (2009) "Lagom generiC: an agent-based model of growing economies", ECF working paper 1/2009.

[2] Of course the actual look depends on the operating system and system–preferences of the operating system. E.g. it's possible that the .jar and .pdf file–extension are not shown.



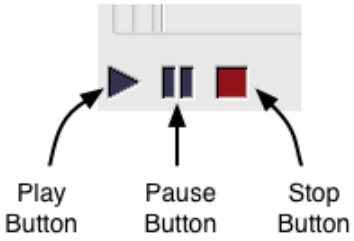
The Console



The Sector Line Chart Display

The *Console* is the main window in which you can edit the model parameters, see the numerical results of the simulation and open display windows showing the results through different types of diagrams. The sector line chart is one of the displays available. It will show you graphs corresponding to values aggregated over all firms of a sector. Other displays will show you values at the level of the firm. At the moment our sector line chart display is completely black because the model isn't initialized yet.

Take a look at the lower left side of the Console, there you can see the *Transport Section*:

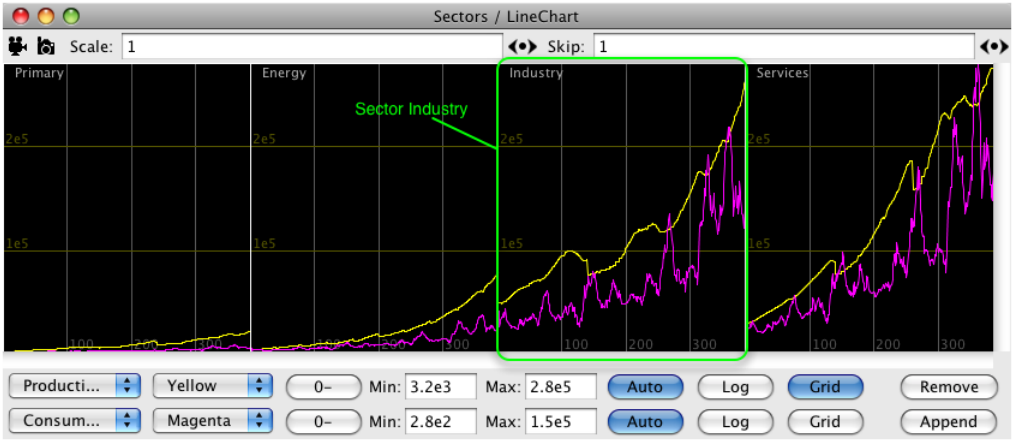


These buttons have the following functions:

- The *Play* button starts a simulation. If the simulation is paused, pressing the play button will resume the simulation.
- The *Pause* button pauses the simulation. This may be quite handy for inspecting the current state of the simulation. Once paused, the simulation may be either continued by pressing the play button, or advanced by a single step (by pressing the pause button again). When pause is pressed, while the simulation is stopped, the model will only be initialized.
- The *Stop* button completely stops a simulation. Once stopped, a simulation cannot be resumed, instead the next time the play button is pressed a new simulation run will be calculated.

So let's just press the play button and let the simulation proceed. Beside the Stop button, a time-counter is launched. It indicates the last date[3] for which the simulation values have been computed:

Also something has happened in the sector line chart display, the simulation results are charted in realtime:



[3]The date 1991.25 has to be interpreted as: The first quarter of the year 1991.

In the sector line chart display you see eight different curves, in two different colors. The yellow curves show the number of units produced in each sector, the magenta ones the number of units consumed.

The horizontal grid lines show the number of calculated periods (the corresponding "real-world" interpretation of one period is a parameter of the model itself, in the current run it is set to one month).

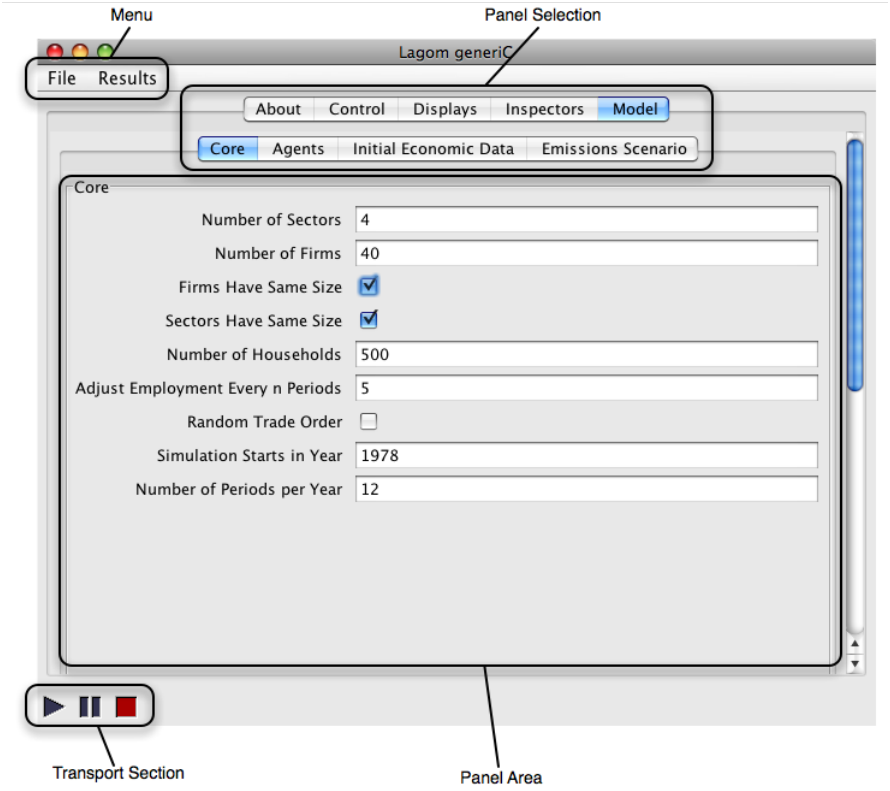
For the "units produced" curve horizontal grid lines are also shown. They serve as a graduation for the corresponding curve. Please be aware that the consumption units do not necessarily have the same scale. The scaling can be controlled with the user interface elements in the footer of the display, this will be described in Chapter 5.1 *The Display Footer*.

With the leftmost element of the footer, you can change the displayed variables. If you do that while the model is running, the curve of the new variable will not be charted from the first period. This is due to the fact that the model implementation doesn't store all values calculated for all periods, because the amount of data can get huge really fast. But it's possible to write all results into files, see Chapter 6.3 *Write Result Files*.

You can also restart the model after having changed the displayed variables by pressing stop and play again. You will then get slightly different results. This is caused by the random generator. By default it will use a different random seed for each model run. We will come back to this in Chapter 6.2 *The Control Panel*. We first want to change some of the model parameters. This can be done in the Console.

4 The Console, Part I

Let's start with a short overview of the elements of the Console:

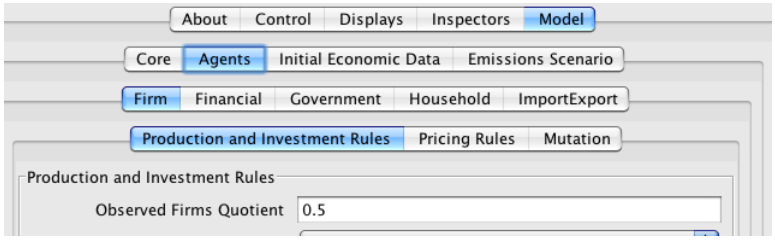


We have already discussed the transport section at the lower left corner. At the upper left corner there is a small menu which is mainly dedicated to file handling. The usage of the *Panel Area* depends on the selected panel:

- The *About* panel gives some short information about the model and its developers.
- The *Control* panel contains settings that are controlling the simulations run, like the random seed, the numbers of periods to calculate, etc.. Details are described in Chapter 6.2 *The Control Panel*.
- The *Displays* panel is used to open and close displays like the sector line chart display. See Chapter 4.3 *The Displays Panel* for more details.
- The *Inspectors* panel is used to investigate the results values. Inspectors are created from the displays. How this is done is described in Chapter 6.1 *The Inspectors Panel*.
- But we will start now with the *Model* panel, which contains the model parameters.

4.1 The Model Panel

The *Model* panel has a hierarchy of subpanels used to organize the different parameters of the model. Eg. all agents are grouped in the panel *Agent*, which has subpanels *Firm*, *Financial*, *Government*, *Household* and *Import/Export*. And for the firm, which have a lot of parameters, you will get additional subpanels as shown here:

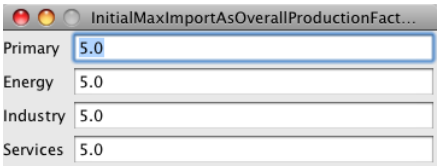


Let's try to modify one of the parameters. Namely, let's change the numbers of suppliers a firm observes [4] and possibly buys from. In the model, there is a parameter that determines this number as a fraction of the total number of firms.

To change this fraction, you should first select the *Production and Investment Rules* panel in the Panel Selection area as shown above. Click on the field behind `Observed Firms Quotient` and reduce this value from 0.5 to 0.3. When you restart the simulation, you can see that the production is lower. When you set the quotient to zero, firms will no longer be able to buy inputs, so that the production goes down to zero after a while.

The firms can transitorily produce goods, because they can import part of their inputs, but they reduce their production, because they do not find enough customers for their products. Now, you can also disable the imports by switching to the *Import/Export* panel. There you have the parameter `Initial Import Limitation Factor` with an *Edit* button behind. This parameter can have different values for each sector. To change these values press the *Edit* button, and a new window will appear:

[4] In this context observing means that the firm asks for the price and deliverable quantity.



When you set one of the values to zero, the firms can't import the corresponding goods anymore. But in the default simulation, firms use each kind of goods as input, so that production will break down fast.

4.2 Parameter Files

We want to go back to the original *scenario* (the default set of parameters). This can be done using the File–Menu in the upper left corner of the Console:

- *Open...* opens an existing scenario. In the folder `scenario`, you have already got different interesting scenarios installed (see Chapter 2 *Installation*).
- *Save* saves the last opened scenario with the same filename (so the scenario will be overwritten). In the case where no scenario had been previously opened, the file `default.csv` will be overwritten.
- *Save As...* opens a file dialog which allows you to set the filename of the scenario.

The `default.csv` file besides `Lagom_generiC.jar` is automatically read when Lagom generiC is started. So we select *Open...* in the file–menu and point the appearing file–dialog to the `default.csv` file.

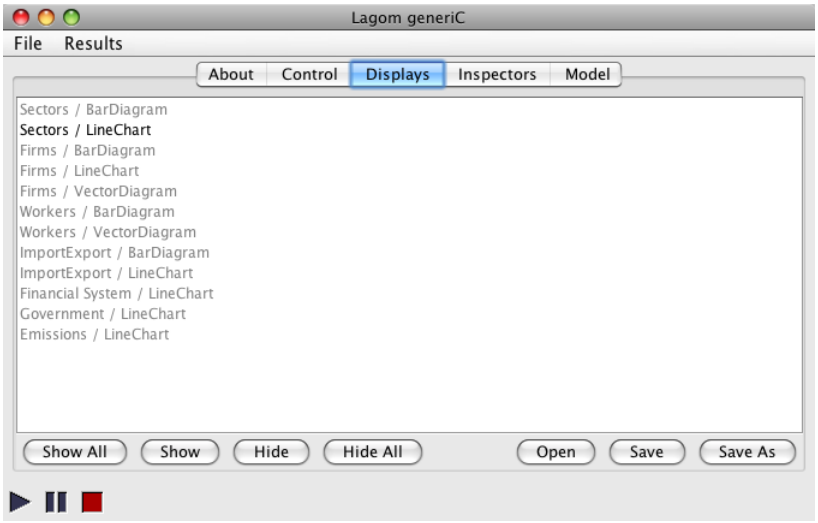
The parameter files are in simple file format[5], that can be edited with any text-editor and most spreadsheet applications. Details of the file format are described in the Appendix A.

4.3 The Displays Panel

We know now how to feed the model with input, but we have only scratched the surface when it comes to the issue of getting access to the results. Let's go a little bit deeper into this issue.

In our former simulation runs we used the sector line chart displays to take a look at what's happening. But the model has many more entities, e.g. the financial system, which isn't part of a sector. Different displays exist for the different entities. They can be opened in the Display panel.

[5]The file–extension csv stands for Comma Separated Values, the file is just a big table written as ASCII text with commas as delimiters between fields



The Display panel allows one to select which displays will be shown, and which will not. Each display is identified by a unique name, the first part names the shown entity, the second part names the diagram type.

To hide a display, you must select it in the list of displays, and then press the *Hide* button. Alternatively, clicking on the display's close button will hide it. To show a display, you must select it in the list of displays, and then press the *Show* button. You may show/hide all displays by pressing the *Show all* / *Hide all* buttons.

With the *Open*, *Save* and *Save As* buttons it's possible to store and reload the current state of all displays. The state includes the information about which displays are shown and in which position, but also the information on which parameters are shown in the displays with which scaling. The default display file that will be loaded when the application is launched is called `default.display`.

The simulations may run much faster without the visualization component[6]. Therefore, one may choose to hide the display, and show the display later on. If you have a lot of displays open, you can save the display state in a file, press *Hide all* while running the model and use *Open* to restore the display configuration.

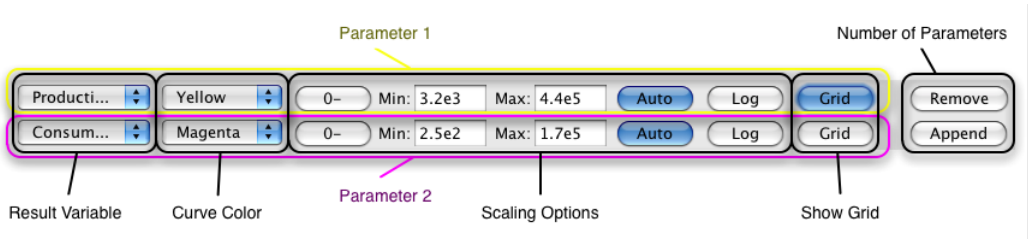
5 The Result Display

Now we know how to open and close displays, in this chapter we will learn how to use them. Let's first try to show the evolution of prices in our sector line chart, that we have used in the Chapter 3 *First Contact*.

5.1 The Display Footer

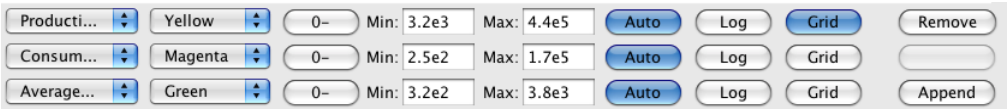
Each display has a footer, which allows us to determine which variables are shown and to scale the codomain. The details of the footer differ for the different diagram types, the line charts have the most complex footer, the bar and vector diagram have less options. At the moment the footer of the shown sector line display should looks like:

[6]The speed improvement depend on the display type, model parameters and, in the case of line charts, on the number of already calculated periods.



Each row in a footer is dedicated to one variable. The example shows two variables: in the upper row the number of produced units and in the lower row the number of consumed units. The *Append* button on the left side adds an additional row, the *Remove* button removes the lowest row.

We need an additional row for the prices, so we press *Append*. Now the footer looks like this:



The additional displayed result variable was selected automatically[7], so it would be very lucky if we would see our price variable already. But we can select it by clicking on the *Result Variable* combobox on the left side. When we do this, we will get a list of all parameters that can be shown in the sector line chart display. The *AverageTradePrice* will show us the traded value divided by the traded quantity. That's what we were looking for.

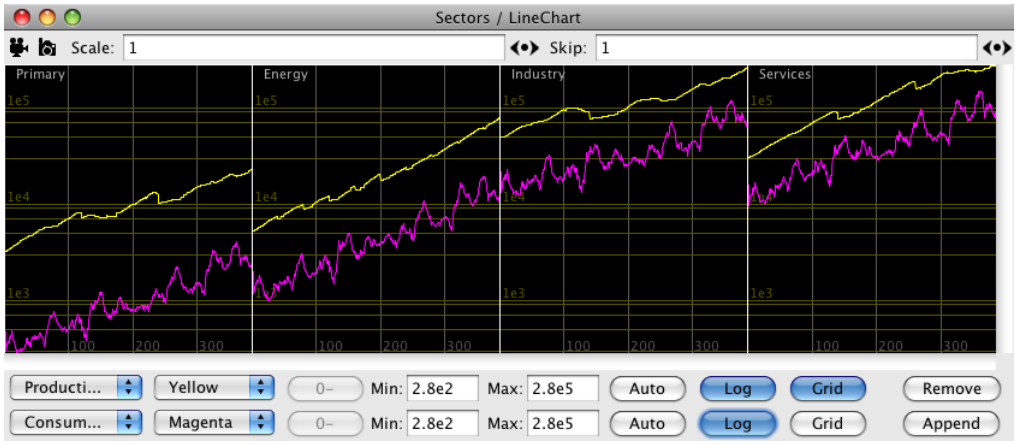
With the current scaling settings, the codomain of the curves will be automatically adjusted so that - independently for each parameter - the minimal value over all periods and sectors are shown at the bottom of the display and the maximal value at the top. This has the consequence that parameters can have different scales. This can be misleading, e.g. when you look at the display on page 5 and want to compare the number of produced and consumed units, you can get the impression that in sectors Industry and Services the number of consumed units is sometimes higher then the number of produced units.

But when you look to the *Min* and *Max* values in the *Scaling Options* area, you can see that this is not the case, the codomain of the production units in the screenshot goes from 3200 to 280000, but the codomain of the consumption area goes from 280 to only 150000. To get the same codomain for both variables click on the *Max* field of the Consumption variable and enter here the Max production (in our example 2.8e5), and also change the *Min* field of the Production to the minimal Consumption.

After you have changed the min and max value it's still difficult to compare the growth rates for the different sectors because of their exponential behavior. The *Log* button allows us to switch to a logarithmic scale.

Also for comparing the consumption and the production, the price curve isn't helpful. Still, we may need it afterwards again. So instead of removing the last row, we can hide the curve via the *Curve Color* selector. Apart from different colors there is also an entry "Hide Line". After we have hidden the price curve, our display should look approximately like this:

[7] In the n–th row it's the n–th parameter of the parameter list.



Let's resume the simulation for a while and we will see that some of the curves will "leave" the display. This happens because when we changed the *Min* or *Max* value, the automatic calculation of the minimal and maximal values gets disabled. Curve fragments that are outside the minimum to maximum interval will be signalized with red lines on the top or bottom of the display screenshot:

You can press *Auto* to enable the automatic scaling again.

5.2 The Display Header

Let's now take a look at the header:

It is really useful to demonstrate your work to others. For this purpose, we included a *Record Movie* and a *Take Snapshot* button. Movies usually fit nicely into presentations, talks, and possibly web pages. Additionaly, snapshots could improve papers and reports. Taking movies is easy: click on the *Record Movie* button, and select an encoding. The movie will contain a frame for each frame that is displayed. Therefore, it may be a good idea to reduce the size of the movie by increasing the number of skip frames (as described below). To stop recording the movie, press the record movie button again. Taking snapshots works also very easy: upon pressing the *Take Snapshot* button, a small dialog pops up asking for the name and location of the snapshot.

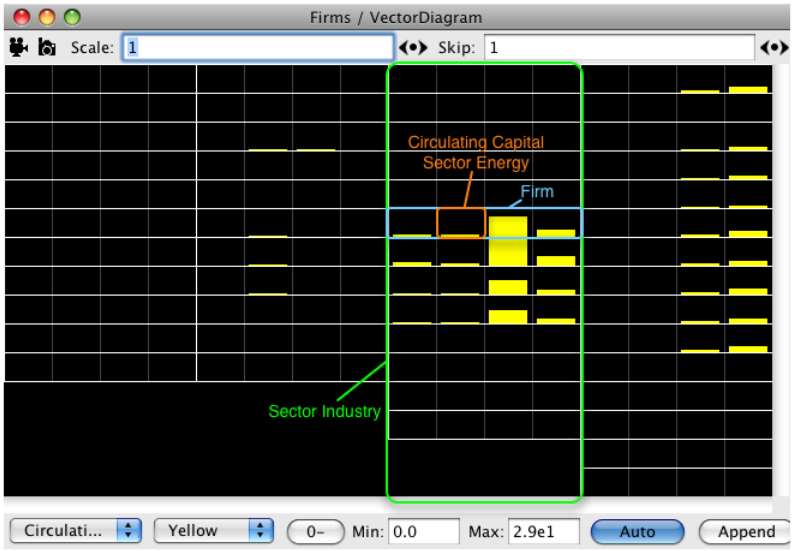
You may scale up or down the visualization in order to concentrate on specific parts of the display. To do that, you may in the *Scale Display* area either set a desired scale, or play with the arrows to halve or double the current scale. The scale may also be reset to 1 by pressing the belly button.

Sometimes the result values may not change drastically between consecutive time steps. However, redrawing everything may slow down the entire simulation. In such cases, it may be desirable to skip some of the frames. The *Skip Frames* area specifies how many time steps will pass between frame updates. The default value is 1 - all frames will be displayed. By setting the value to 2 for example, only every other frame will be displayed. This is especially useful for the line charts, which is the only diagram type we have seen till now.

5.3 Diagram Types

But besides the line charts diagram we have two other diagram types, which only show the current state of the result variables, using bars. The difference between the two is the type of variables that can be shown: the *Bar Diagram* gives access to the same variables as the line charts, but the *Vector Diagram* allows us to take a look at variables that are vectors. In our case these are variables where we have a value for each sector, e.g. the circulating capital stock of the firms. Let's take to look at these, what we see here can be easily transfered to the Bar Diagrams.

Therefore first stop the model run and then select the Displays panel. Here you select the *Firms / VectorDiagram* display and then press the *Show* button. Restart the model, after some periods it will look like this:



The display shows the circulating capital stock for each firm. Therefore each firm has its own region in the display, the regions are seperated by light–grey lines. In the screenshot above, the region of one firm is highlighted with a blue rectangle[8]. Inside of the firm–region you can see up to four bars, separated by dark–grey lines. These symbolize the amount of circulating capital stock that the firm has of different goods. The goods are sorted in the same order as in the sector line chart display, so that the bar that is surrounded in orange shows the amount of energy that the firm has left in his stock.

[8]This rectangle isn't part of the display and was added only afterwards for this manual. The same is true for the other highlighting elements.

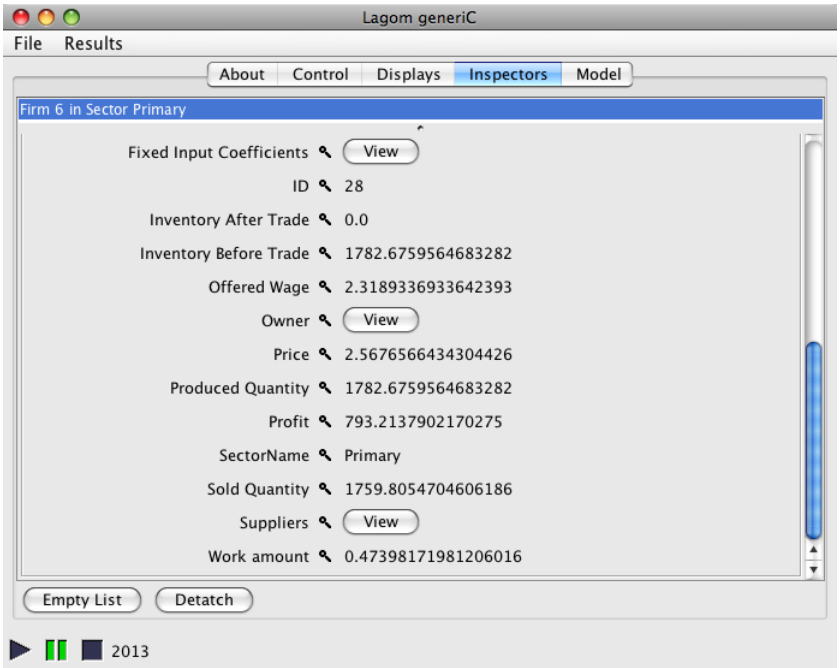
The firm regions are also sorted. The columns determine the sector to which the firm belongs, also sorted in the same order as in the sector line chart display. As you can see, we have a different number of firms per sector. We started with ten firms per sector, in our example screenshot we got thirteen firms in the industry sector. This sorting exists only for the firm displays, because the households can't be assigned to a single sector.

You can get the exact values of the stock by moving the mouse over the region of the firm you are interested in. A tooltip appears that shows the values in a numerical manner. But even better, you can also double-click on the region, then the Console will show the corresponding firm in the Inspectors panel, which will be explained in the next Chapter .

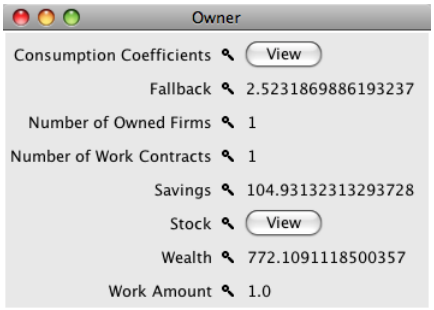
6 The Console, Part II

6.1 The Inspectors Panel

In the *Inspectors* panel you get an overview of the complete state of a single entity (e.g. a single firm, a single sector, the financial system ...):



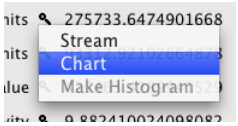
At the top of the panel you can see which entity is shown, below are listed its different variables. If the variable is a vector, like the `Fixed Input Coefficients` in our example, a *View* button is shown after the variable. Pressing that button will open an additional window that shows the vector. Also the result variable can be a reference to another entity. E.g. the owner of a firm is a Household, so when you press the *View* button beside `Owner`, a new windows appears that contain the state of this Household:



The *Inspectors* panel contains two additional buttons: Empty List and Detatch. The first one will empty the inspector panel. This will eliminate the extra cost of updating/displaying information for the inspectors, and the simulation will get faster.

The other button detaches the inspector information into a separate window. Why is this helpful? Because if you want to inspect another entity, the inspector panel doesn't show the previous entity anymore. E.g. when you want to compare two different sectors, this can be done by detaching the inspector of the first sector and then selecting the second sector. You can detatch as many entities as you want. Once detatched, the inspector may be closed as a regular window.

If you click on the *magnifying glass* icon, as shown on the right, a pop-up menu appears showing additional plug-in inspectors for various properties.



For example, we can chart a value or stream it out. Select *Chart*. A charting facility will appear which will track the value of your variable. You can put multiple result variables on the same chart. E.g. if we want to have the number of produced units per sector on top of each other instead of side by side, we can use the charting facility. The following steps must be taken to archive this:

1. We want to have the chart beginning from the first period, so first stop the current simulation run by pressing stop.

2. To setup the chart, the model must be initialized, this is done by pressing the pause button.

3. Open the first sector in the *Inspectors* panel by double-clicking on the sector in the sector line chart dispay

4. Click on the magnifying glass beside `Production` and select *Chart*. A new window should appear.

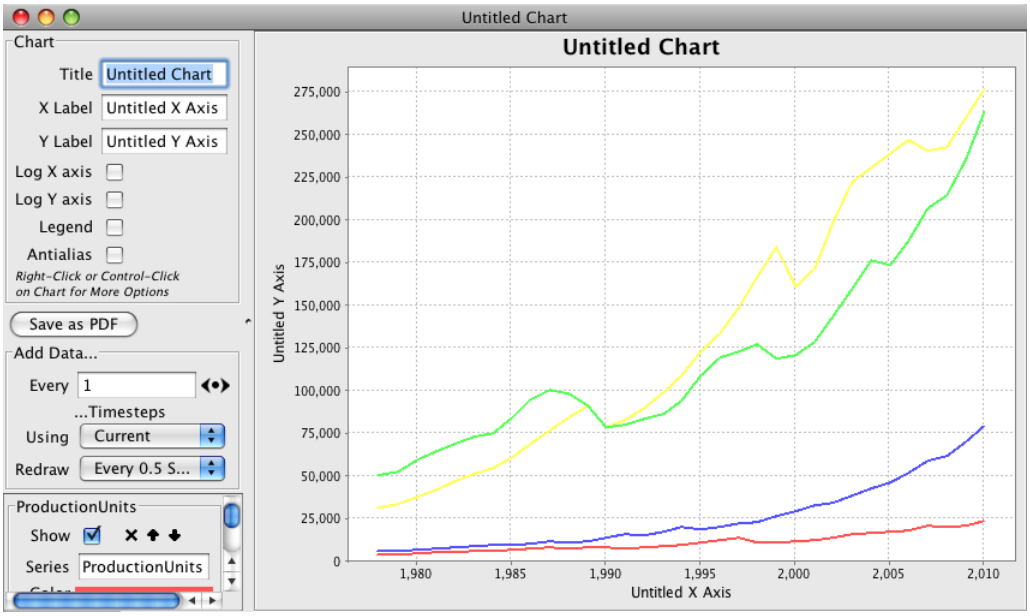
5. Open the second sector in the *Inspectors* panel.

6. Click again on the magnifying glass. This time a dialog will appear, that has the option to create a new chart or to add the variable to an existing one. We haven't named our chart, so it's called `Untitled Chart` in the selection box.

7. Repeat the two last steps for the other sectors.

8. Start the simulation run by pressing the start button.

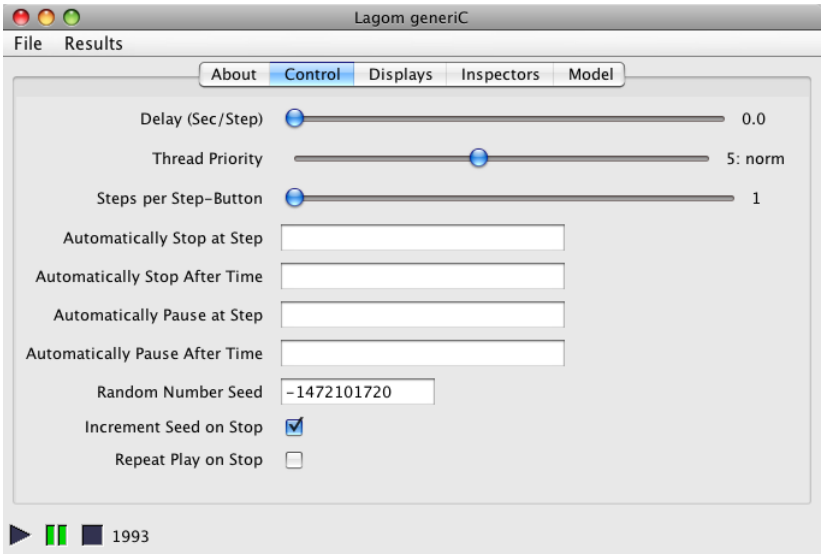
You should now get a chart that looks like this:



Please notice that this chart facility has a different behavior regarding new simulation runs than the diagram displays. The charts are detached from the model when you stop the simulation, when you run the simulation again, you will still have the chart of the older simulation. But the chart is still shown in the chart selection dialog that appear after you press the magnifying glass. That allow you to plot the result of different simulations runs into one chart, e.g. when you want to compare the results of runs with different random seeds. How you can control the random seed is described in the next Chapter .

6.2 The Control Panel

The *Control* panel contains useful settings for the simulation:



The first three slides allow one to insert some delay between each time step (this can be useful for the Vector- and Bardiagrams), to increase the priority of the simulation thread, and to execute any number of steps upon pressing the pause button while the simulation is paused.

Next, there are four text areas where you can specify the time step where the simulation should stop or pause (once stopped, a simulation cannot be resumed). In the *Automatically Stop/Pause at Step* fields the number of calculated periods are used as stopping criteria, in the *Automatically Stop/Pause After Time* fields the corresponding year (which is also shown right beside the Transport Section).

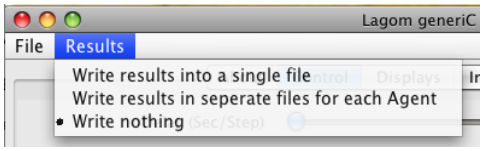
The next two elements control the random seed. In the text field you can set the random seed for the simulation to a certain value. By default, the random seed is incremented whenever the simulation is stopped (so that the next time the simulation is played, it runs with a different seed). This can be changed with the *Increment Seed on Stop* check button. The seed is also stored in the *Parameter Files*, but not the *Increment Seed on Stop* status.

Setting the last check button, *Repeat Play on Stop*, sets Lagom generiC to automatically start a new simulation whenever the old one is stopped. This is mostly helpful for kiosk modes, but also allows us to generate files with the results of different runs (with different random seeds) automatically.

6.3 Write Result Files

Two different types of result files exist. In the first variant, all result values are written into a single file, in the second variant, a new folder is created with a file for each entity. In most cases the second solution is better, because the files contain tables that can be easily used in spreadsheet applications.

You can configure the creation of result files in the *Results* menu of the Console window:



When you select *Write results into a single file* each time you start a simulation run, a file with the name `Results "timestamp"` will be created beside the `Lagom_generiC.jar`. In case you select *Write results in seperate files for each Agent*, a folder is created, and inside of the folder the files are named after their Entity and Id.

And here we are now at the end of our tutorial. We hope you have enjoyed it and enjoy now playing around with Lagom generiC .

A Parameter files file format

For loading and saving the parameter values of a model into a file, we use a generic mechanism that is model independent. Which means that there is no loading function in the form "The first value is the number of sectors, the second value the name of the first sector...". Therefore the file must contain the information about the parameters itself. The parameters in a file can be also a subset of the model parameters.

The file is a comma-separated value (csv) file, which is just a text file which uses comma or semicolons (and line endings) as separators between table entries. The whole file should be seen as a table where the number of rows is equal to the number of lines, and the number of columns is equal to max(3, the number of sectors).

In a row (line) we can have the following content:

- Comments
- Parameter Header
- Parameter Values

A.1 Comments

Comments have a `//` in the first column, the following columns can be used arbitrarily. e.g.: `//,this is just a comment`

A comment line can be placed anywhere.

A.2 Parameter Header

The first column must contain the type of the parameter, which can be

- `Int`
- `Float`
- `FloatVector`
- `FloatMatrix`
- `Boolean`
- `Text`
- `TextVector`

In the second column the name of the class must be given, as seen in the Source Code Documentation (the ...InitValue classes), and in the third column the name of the parameter, as shown in the Source Code Documentation in the Private Attribute list. So in our Lagom generiC implementation we have as an example for the number of sectors: `Int,SectorInitValues,numSectors`

A.3 Parameter Values

The format of the line(s)/row(s) after a line with parameter header depends on the parameter type. In addition, for matrices, only one line is used for the values:

- `Scalar`: In the case of a Int/Float/Boolean or Text type, only the first column is used for the value of the parameter (in case of Boolean type the text "false" or "true" must be used).
- `FloatVector` or `TextVector`: A vector is created with the same size as the used columns in the row. The value of the first column will be read as the first element of the vector ...

- `FloatMatrix`: A quadratic matrix is created with the same size as the used columns in the first row. The columns of the matrix are the columns of the csv file, the rows of the matrix are the rows that follow the parameter header row.

A.4 Summary Example

Here we have a possible start/part of a csv-file:

```
//,Example
Int,SectorInitValues,numSectors
4
//
TextVector,SectorInitValues,nameArray
//,a string can be also set inside of double quote signs, they will
//,be removed while loading the file
Energy, "Agriculture", Construction, Other
//
Boolean,EnvironmentInitValues,firmAdjustWorkforce
//,value isn't case sensitive, so also FALSE, False or even fAlSe would
//, be allowed (and of course also all variations of true
false
//
FloatMatrix,SectorInitValues,capitalStockMatrix
10236.5,12443.2,24421,29582
9289, 14202, 3326.74, 12058
0,0,0,0
//,also semicolons can be used a delimiter
23215.4,22365.24;69565;2325.5
```

A.5 Hints

A good start to create a csv-file is to run the model implementation, set the correct number of sectors in Model/Sector/NumSectors field and select "Save As..." in the file menu. To edit the file you can use any text editor or applications.