USER MANUAL OF EMERGY SIMULATOR

Raphël Valyi (Ecole Centrale de Lyon ”generalist engineering” student)

July 13, 2005

Contents

Chapter 1

Modeling at different levels

1.1 Overview of the modeling methodology

While also being open for simpler usages (like diagram editing), EmSim attempt at reflecting the steps you would do when modeling ecological economics systems:

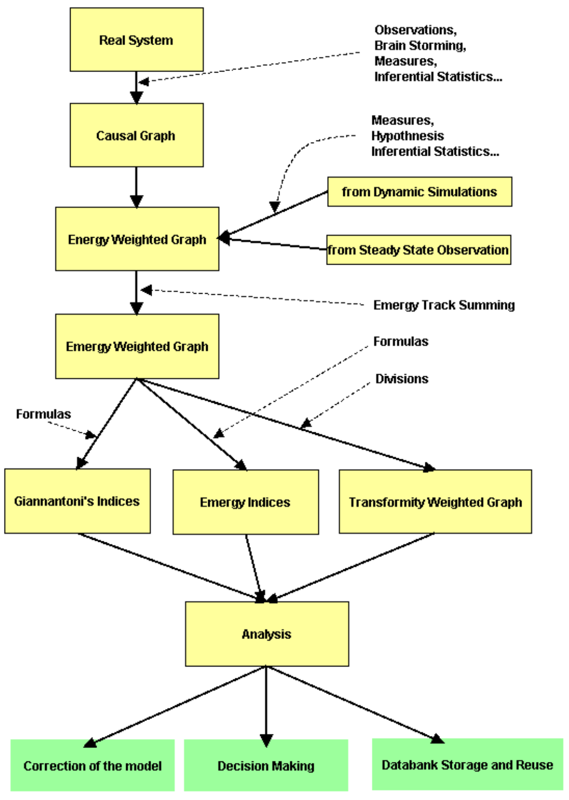


Figure 1.1: simulation methodology

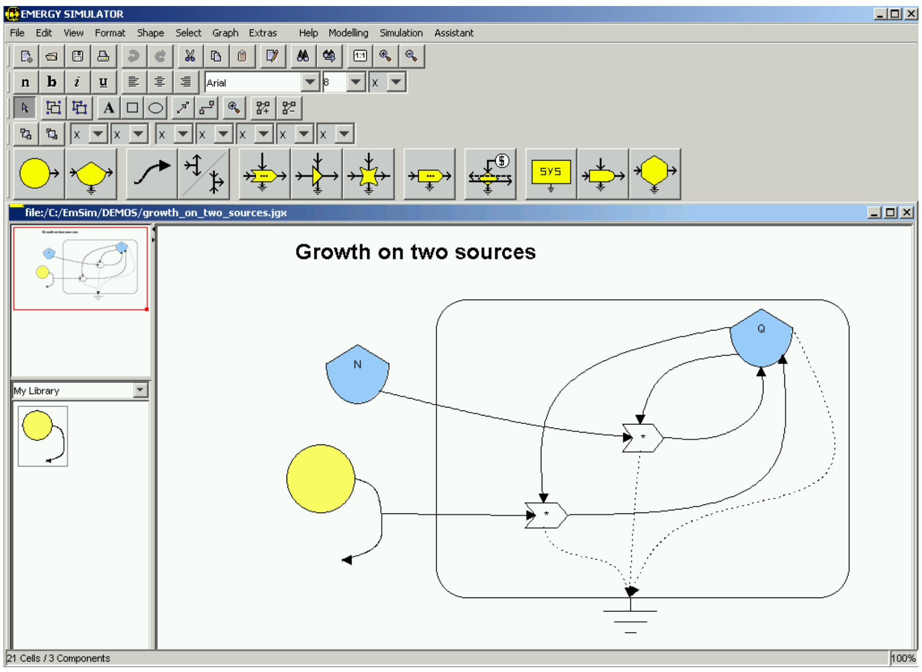

EMERGY SIMULATOR
File Edit View Format Shape Select Graph Extras Help Modelling Simulation Assistant
file:/C:/EmSim/DEMOS/growth_on_two_sources.jgx
Growth on two sources
N
Q
My Library
21 Cells / 3 Components
100%

Figure 1.2: an overview of the Emergy Simulator interface

1.2 The different levels of modeling with Emergy Simulator

1.2.1 Drawing an energy graph

That's the first level of using Emergy Simulator. You can draw an energy diagram respecting some conventions enunciated by H. T. Odum [5]. You can ignore the upper modeling levels and forget the dynamical constraints. For that purpose, Emergy Simulator offers all standard features you can expect from a commercial diagram editor. See the graphical user interface for more informations.

1.2.2 Running a static emergy analysis

Once your energy diagram is done, if you didn't get enough data to build a dynamic mathematical model, you could nevertheless by hand allocate static available energy (=exergy) flows to each pathway in order to process a static emergy analysis.

The only thing you need to be really cautious is to determine whether energy splits are extensive split branchings or co-productive split branchings, see **??**. Indeed, consistent emergy algebra has only been built upon this basis. Lots of people do their approximations around those two kind of primary branchings because they don't manage to fit real flows in any class of elementary branchings. Clearly some more theoretical research should be undertaken in this field. But at the moment, considering all I've read, I can't find any scientific way for accounting non elementary branchings. On the other hand, it seems that a profound investigation of physical phenomenons always permit to express a flow in various primary flows. At the moment, **Emergy Simulator only pretends to run emergy algebra with elementary branchings**.

Once this allocation is done, you will be allowed to run a static emergy track summing, that's a propagation of emergy in the network respecting the emergy rules ennuciated by H. T. Odum.

1.2.3 Running a dynamic quantity simulation

To do this, you will need to build your energy diagram while keeping in mind the dynamic mathematical meaning of each component. No abusive aggregation will be allowed here. That's is the model will be based on elementary elements only, no groups without any internal description will be allowed.

However, hiding a complex subsystem by aggregating it into a producer, a consumer or a generic subsystem symbol should be allowed in the future. But I'm quite sure I won't have enough time to code such a feature. Instead, after a compete model definition, the user could be allowed to artificially make the complex system smaller by zooming out and eventually could hide it behind a group symbol while this would not be interpreted by the software.

1.2.4 Running a dynamic emergy simulation

If you can build a dynamic model and if you can tell what is the value of each input independent emergy source, then Emergy Simulator will perform a track summing algorithm respecting the emergy algebra and will plot the emergy to each node along the time and the transformity in each pathway.

Please notice that **very few dynamic emergy studies have already been undertaken. The reason is probably lack of real data, lack of competence in dynamical systems and finally lack of simulation tools.** I should say that what I've seen from the Extend source code convince me that no track summing was performed and thus, for complex systems, emergy algebra wasn't respected. I've only been told to build a dynamic energy simulation tool, I argued that the track summing algorithm isn't really hard to implement once you make the effort of knowing a graph handler, so the situation may slightly change with emergy simulator, well hopefully. **One should keep in mind that a dynamic simulation (even wrong) is far better than a static one since it allows historical perspective and thus validation or invalidation potentiality.**

Chapter 2

Introduction to Odum's dynamic energy language modeling

2.1 A language similar to bond graphs

The energy language introduced by H.T. Odum in [4] is very similar to the bond graph language [2]. In 1983, in [4] pages 87-88, Odum was already aware of this similarity but in 2004, it seems that this similarity should be even more exploited. Indeed, in our todays information globalized world, bond graphs became a very efficient way to make various systems interact with each other by exploiting the very universal energy interface. Also lot of research about bond graphs have been done. Today it seems to be a standard of simulation and communication.

In the sake of some universality, as much as possible, in this report, similarities between Odum's energy language and bond graphs are explained. Thus systems ecology gets closer to this universal tool also really appropriate for the emergy algebra instead of going in an esoteric direction.

2.2 Fundamental principles

2.2.1 Describing a system evolution by means of state variables

First, the user should understand that the Odum modeling language, like many others require to define a set of variables that characterize the state of the system, they are called state variables. Those state variables are changing along the time. **Notice that the in the Odum energy language we are only looking at the state variable evolution with regard to the time and not with regard to other variables**. Thus we will describe our time evolution by means of ordinary differential equations by opposition to equations with partial derivatives. For more information about this point, look at **??**.

$$\text{time dependent system state} \Leftrightarrow \text{set of time dependent state variables}$$

2.2.2 Diagramming the first order state variation

As being a graphical language, the Odum energy language requires the user the "draw" a diagram expressing the constraints linking the state variables of the system to simulate.

The first step is to draw the state variables. For each state variable, design a storage component in the graph area. Be careful, storages aren't always capacitive, they could also be inductive while this is less common in ecological modeling. Also notice the you will then express the variations of first order of those state variables because the system should be ordinary and of degree one. So you should design as much new storage as required to get such a system, look at **??** for more details.

state variable $\Leftrightarrow$ storage symbol

Basically, storage variable are supposed to be either energies, either variables which are linked to an energy, like material quantities. **To fulfill the energy language strict definition, when those quantities are flowing across the pathways of the graph, the energy transfered should be proportional to the flow**. At least that's the condition for applying emergy concepts on your diagram.

Thus each pathway of the energy network do have an energy power expressed as:

$$P = e \cdot f$$

were:

P : is the energy power in Joule per second (S.I. Units)

e : is the **pseudo-effort**

f : is the **pseudo-flow**

2.2.3 Diagramming a state variable variation

For each state variable, your diagram should express the way this state variable evolve along the time. So basically, the first order variation of this variable is a linear combination of terms and, each term is diagrammed as a flow inflowing or outflowing at the storage:

state variable variation term $\Leftrightarrow$ pathway connected to the storage

By connecting the storages in various way, you are building a **graph strucure** (and even multi-graph actually) composed by **nodes** and **edge or pathway**. Nodes are controlling energy flows through their connexion ports whereas edges only represents those flows and determine if flows have a special behavior depending only on the force and back force the two pathway extremities.

2.2.4 How an edge flow is defined

So you know that each term of a state variable variation is equivalent to the flow in an edge connected to the corresponding storage. So now you preoccupation is to define that flow properly.

The flow through a pathway depends both on the edge type and both on the two components connected at each side of the edge. Be careful, some components behave differently depending on the port they are connected.

We will explain how is computed the flow crossing an edge by default at first:

1. if the component connect at the target of the edge is demanding a flow, then this flow is pumped to the component connected at the source of the edge.
2. else if the component connected at the source of the edge impose a flow, then, this flow is imposed to the component connected at the target of the edge.
3. else the component connected at the source of the edge should provide a force. The resulting flow is the product of this force times the pseudo conductivity of the edge.

Moreover, depending on the edge model, the flow can be altered. For instance a sensor edge only pass the information and have a null flow. An edge with back force provide a flow proportional to the force gradient when no flow is imposed. At the end, you would need to study the following design element section to understand better how each component works.

2.2.5 Adding other nodes than storage to control the flows

So you know that there are pathways and nodes. But at the time you only heard about storages which embodies the system state variables. Actually, to fully control the flows defining the storage variations, you need to add other kind of nodes.

A basic classification of those other nodes would be:

sources: they are working like storages (providing force or flow) but we neglect there variation at the scale of our system.

regulations: the flow crossing a regulation node (has two ports) is **an unary function** of the force or force gradient between the two adjacent nodes.

interactions: between various components. All interactions can be composed by several three ports interaction which let pass in each edge a flow proportional to **a binary function** taking as argument the forces provided by two inputs.

macro-interactions: components with more than 3 different ports that provide in fact an association of simple interactions.

discrete event providers: those component impose discrete variation on storage quantities when a boolean condition is verified.

2.2.6 Force and flow providers

A basic concept of the Odum's energy language is that models either provide a force either provide directly a flow. When forces are provided to pathway edges, then the flow is a resulting consequence. On the contrary, we can assume that the flow is forced by a flow provider. Such a component is assumed stable enough to provide a given flow at any force. Such elements are for instance flow sources or inductive storages. Other elements behave differently depending of the port they are connected!

Chapter 3

Dynamical energy language design elements

3.1 Pathways

internal parameter: by default a pathway only have one internal parameter: the pseudo-conductivity factor k. In the case that force providers elements are connected at the both extremities, the resulting flow in the edge is obtained by multiplying the force gradient by the pseudo conductivity.

no back force : Except if the pathway is connected to an interaction symbol, then the flow is only proportional to the the origin node force e_1:

$$f = k \times e_1$$

with back force : Except if the pathway is connected to an interaction symbol, then the flow is proportional to the force gradient, respectively e_1 at origin and e_2 at the target:

$$f = k \times (e_1 - e_2)$$

with back force but valve : The flow behave like it does with a simple pathway with back force but only if the flow is superior to a threshold m. This threshold is an additional internal parameter.

$$f = \begin{cases} k \times (e_1 - e_2) & \text{if } f > m \\ 0 & \text{else} \end{cases}$$

sensor without flow : This pathway emphasis that an parameter from one vertex is passed to an other by the pathway without any appreciable energy flow (an information for instance). For this edge type, the pseudo-conductivity becomes useless. In fact such a pathway only propagate an information.

$$f = 0$$

energy sink : Behave like a classic pathway with back force except that this pathway drive energy that isn't concentrated enough to drive any process in the system. In in the future releases of Emergy Simulator, we will make the pseudo conductivity slave from the other pathway incoming to the origin node since all remaining energy should outflow to the sink. Although we might draw several energy sink, they all will share the same conceptual model (that is the same instance of the sink model class). The power can only be inflowing to that element (half arrow in bond graphs).

$$f = k_{slave} \times e_1$$

3.2 Node classification and number of ports

Although graphically each node do have several ports in order to be connected. One should keep in mind that it's somewhat artificial: it only stands for graphical purposes. On the contrary, mathematically, ports are out from the graph theory. Instead of having ports as children, a node should simply be connected to ports as they would be with other nodes (without this special parent-child relationship).

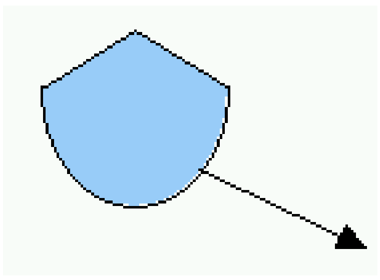
On the other hand, this Jgraph [6] parent-child relationship is really useful in energy language, because a single node may have different behaviors depending on the port it's connected and this should be shown graphically (see for instance the interaction symbol).

But to make clear the difference between the graphical displaying and the mathematical meaning, it should be said that: **although vertex do have lots of ports, they only have a limited number of different port types (ports models)**. For instance a source has got several graphical ports but they only have one port model. That's why I did the following formal classification as people generally do when dealing with bond graphs **??**:

3.3 One-port elements

3.3.1 Storages

internal parameter: the stock which is part of the system state vector.



capacitive storage : examples are: springs, torsion bars, electrical capacitors, gravity tanks, accumulators etc. When encountering a capacitive storage, EmSim builds a differential equation of firts order:

$$\frac{dQ}{dt} = \sum inflows$$

direct consequence:

$$Q = \int_{-\infty}^{t} f \cdot dt$$

inductive storage : examples are: electrical inductance, inertia of mechanical systems etc. When encountering an inductive storage, EmSim builds a differential equation of firts order:

$$\frac{dJ}{dt} = -\sum connectedforces$$

direct consequence:

$$P = \int_{-\infty}^{t} e \cdot dt$$

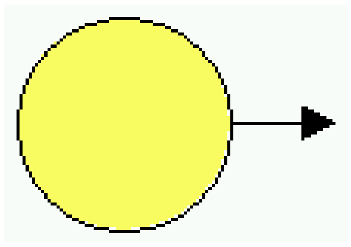
and:

$$f = \alpha \cdot P$$

where α is proportionality coefficient

3.3.2 Sources

In fact a source is only a storage whose variation isn't appreciable within the considered time and space scale. So the resulting force or flow is assumed constant or time dependent but not sensible to the outflowed quantity.



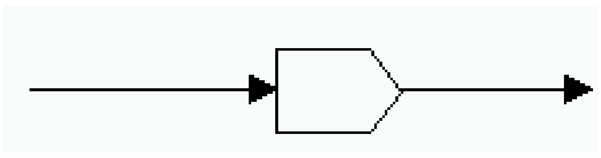
controlled force source : internal parameter: $Force(time)$

controlled flow source : internal parameter: $Flow(time)$

3.4 Two-ports elements

3.4.1 Regulation symbol

A pathway that has a regulation has its flow function of the inforce:
$outflow = f(inforce, time)$
$demandedflow = outflow$ Notice that a regulation represents an unary operator. Notice that there is also an other similar regulation but that provides an $outforce$ instead of flow.



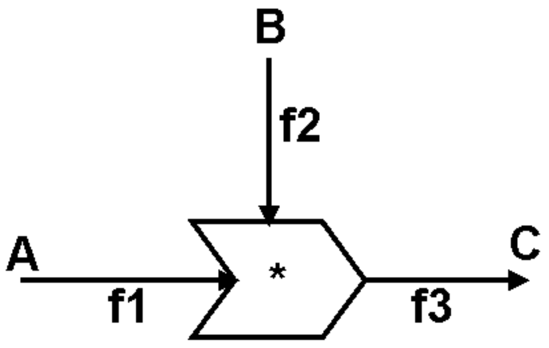
3.5 Three-ports elements

3.5.1 General interaction symbol

Regulations are unary operators whereas interactions are binary operators. Indeed in general the three connected flows (two inputs and one output) are functions of the two inputs. The functions can be different but a rule should be respected, the output can't exceed the input sum in energy. If the output is less, then, a sink pathway should be added downside this element and it should drive the remaining flow. It becomes more complex as several interactions are connected or when pathways are of different kind.
$outforce = f(inforce1, inforce2, time)$ $demandedflow = outflow$

3.5.2 Some specific interactions

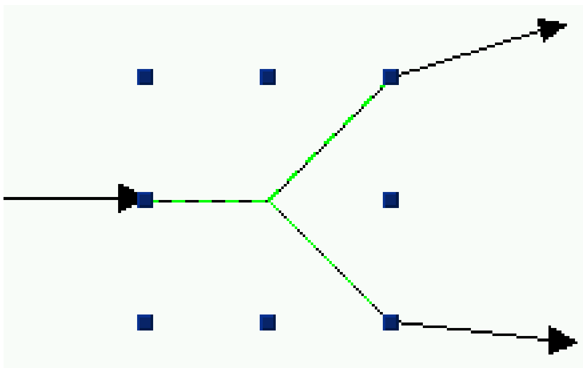
multiplicative: When all pathways are without back force, then: $outforce = inforce1 \cdot inforce2$ $demandedflow = outflow$



amplifier: similar as the multiplicative interaction except that at an entry a null flow is demanded while the out flow is demanded to the other entry.

3.5.3 Split branchings and junctions

A split or a junction of flows only happen one special kind of nodes: the split/junction node. This component is present in the modeling bar and has three ports. An other important characteristic is that its view can be either aggregated, either expanded. By default the view is expanded so the three ports are distinct. In the popup menu, you have the opportunity to switch for the aggregated view, do it only when the split / junction component is connected.



The left port will sum every inflows. Connect several inflowing pathways to that port if you want the behavior of a flow junction.

The upper-right port has the following out force: $\frac{\sum inflows}{(1+\sum M_i)}$

The downside-right port has the following out flow: $\frac{\sum inflows}{(1+\sum M_i)}$

If the connected component provide an aspiration ratio A, then $M_i = A \times R$ Where R is the proportion of the flow naturally going through that port. For instance the multiplicative interaction component will provide an aspiration factor equal the force of the other input force.

However most of components don't provide aspiration factors, in that case: $M = R$. R is an internal parameter of the split that you can set up.

Thus the sum of the outflow equal the inflow.

If you don't want several flows splitting, then only connect the output to the right downside port and set R to 0.

If you want more than two outputs at a split, then you should decompose your split in a cascade of two-outputs splits. Then set up correctly the parameters to obtain the right behavior. An other solution would be you code an other specific model. Look at the developer manual if you are interested to do so.

There is also an other used split: split with proportional out:

The left port accept only one $inforce$

The upper-right port provide a force $force = inforce \cdot edge_conductinity$

The downside-right $remainingflow$

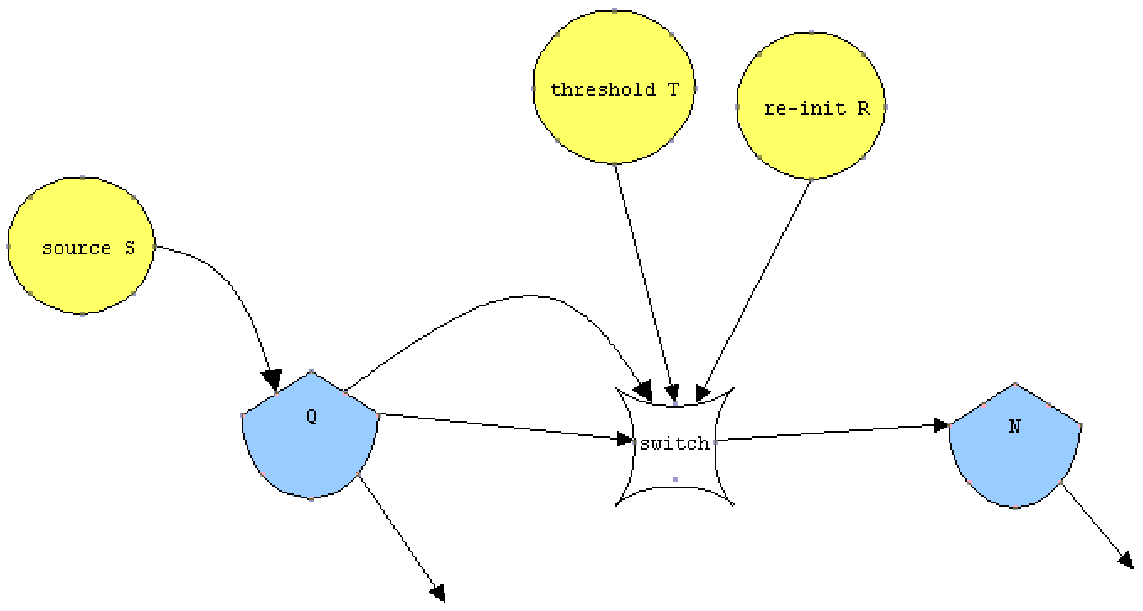
3.6 Four-ports elements

3.6.1 Transaction

An input force is used as ratio that will slave a out force to be proportional to a master entering flow (and flows are conserved). No time to explain it, see how the demo files use it.

3.7 Five-ports elements

3.7.1 Switch



The left port should be connected to a storage to empty

The right port should be connected to a storage to fill

The upper-left port should be connected to the $sensorforce$

The upper-middle-port should be connected to the $threshold$

The upper-right-port should be connected to $re - int$ force value

If $sensorforce > threshold$ then the a storage is set to the $re - int$ value while the other is filled with the same quantity.

Chapter 4

Introduction to Odum's emergy modeling

In order to compute emergies properly across the diagram, a proper modeling is required. Nevertheless, you will see that for steady state emergy considerations, the modeling components have less influence on the results while you could generate some kind of syntax errors.

4.1 Steady state emergy modeling

Simply build you network. The only important things are:

1. the sources you are drawing are supposed to be independent,
2. every flow bifurcations that are not on split models are considered co-productions.

4.2 Dynamical emergy modeling

Similar as the steady state modeling but your diagram should be also a dynamical diagram like for quantities evolutions.

Chapter 5

Working with files and documents with Emergy Simulator

Emergy Simulator handles multi-documents edition. Each document contains a graph and also simulation settings. By using the File button in the menu bar, you can open a new document either empty, either already saved, you can save your current document and also open and save libraries.

Copy (CRTL + C) and paste (CTRL + V) is allowed between documents, libraries and even between several session of Emergy Simulator. However copying cells without modifying parameters can lead to bugs for dynamical simulation purposes. Be careful.

5.1 saving files as XML

Once you diagram is drawn and eventually once you set up the components parameters, save your work by choosing save or save as in the File button of the menu bar. The file will be given a .jgx extension.

Notice that this format of storage is an XML data type. This is really powerful since you could then open your file by any kind of application. Typically this would be useful if emergy diagrams were sent to a server that will build a global emergy knowledge data bank.

An other advantage of such XML file is that there is a big chance that your work will be directly compatible with future emergy simulator versions. Even if not, by knowing the changes (ask for the developer), then you could modify lightly by hand you .jgx file in order to adapt it. For that purpose, open it with a text editor (TextPad for instance) and adapt it.

To open a saved file, choose File > Open and then find your .jgx file in the explorer.

5.2 Saving files as binary files

On the opposite to typed text files like XML, the basic way of saving a work is to store the numeric value of all created objects. In this case you get a very heavy binary file. Libraries of Emergy Simulator are stored this way.

To make a library, select the desired cells of your diagram. Then in the popup menu, choose insert to library. Then in choose File > Library > Save as in the menu bar in order to save the library. You will get a .lib file.

Opening an existing library is done by choosing File > Library > Open.

Chapter 6

The general graphical user interface

We explain here the main features to help you to draw an energy diagram. Let's start with an empty new file.

Use the modeling bar to drop in the desired component into the graph area (main white area).

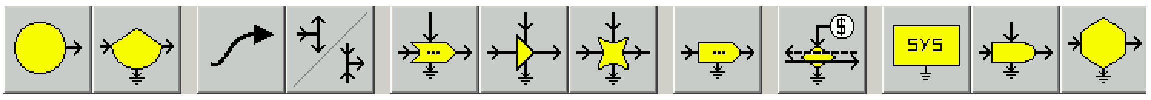


Figure 6.1: the modeling toolbar

Warning: you can drop other components than the ones that are present in the modeling bar. But this should only stand for graphical purposes since those other cells don't have mathematical models neither will store parameters.

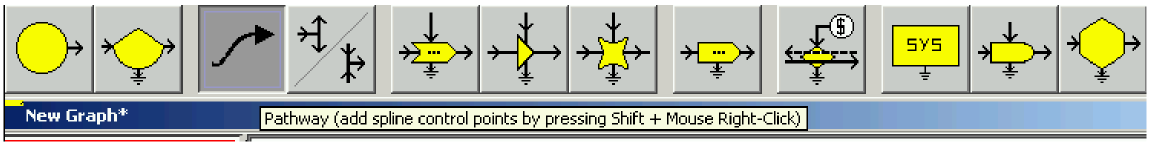
As you can see, before clicking on a button, simply by rolling over with the mouse you get a tool tip text giving some explanations about the action you are about to start.

After selecting one of the buttons of the modeling bar, you should draw a rectangle with the mouse in the graph area. The first click define the first corner of the rectangle while the second click define the opposite corner. The rectangle becomes the bounding box of the component selected.

Be careful, add the node component before connecting them (we called node every component that isn't a pathway).

Once you added some node components. You can connect them with pathways thus building a graph structure. The pathway component should be selected in the modeling bar (on the left). Be careful, pathways should be properly connected: they should be connected to cell ports. Once you selected the pathway button. Approach the cursor from a drawn cell. When getting close to the cell; a live preview of the closest port appears. If you are satisfied do the first click. This define the source of the pathway. Then approach the cursor to an other cell port. By doing the second click you define the target of the pathway.

Now your pathway is displayed. You have the opportunity to blend it by adding some control points. This is done by right clicking on the right place while pressing the shift



Pathway (add spline control points by pressing Shift + Mouse Right-Click)

key. Then move this control point where you want. Add as much as you want. To delete a control point, select it with the cursor and again right click while pressing shift.

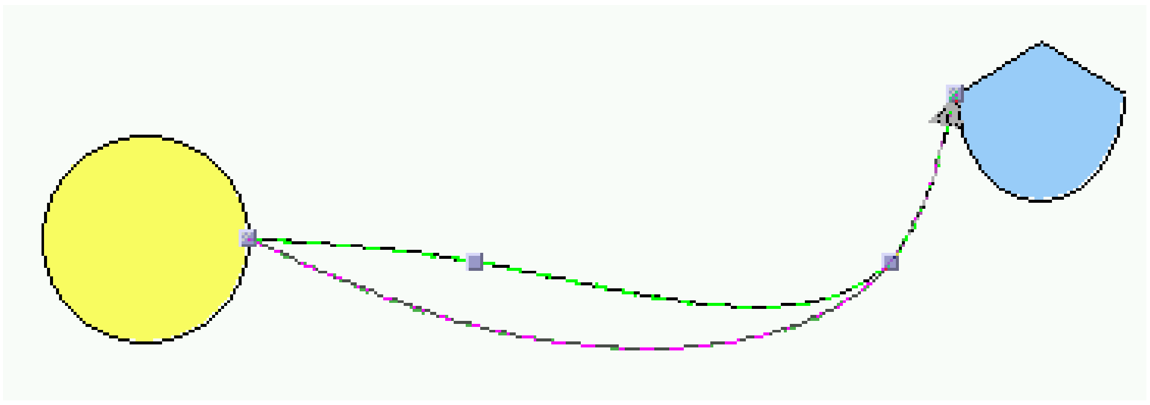
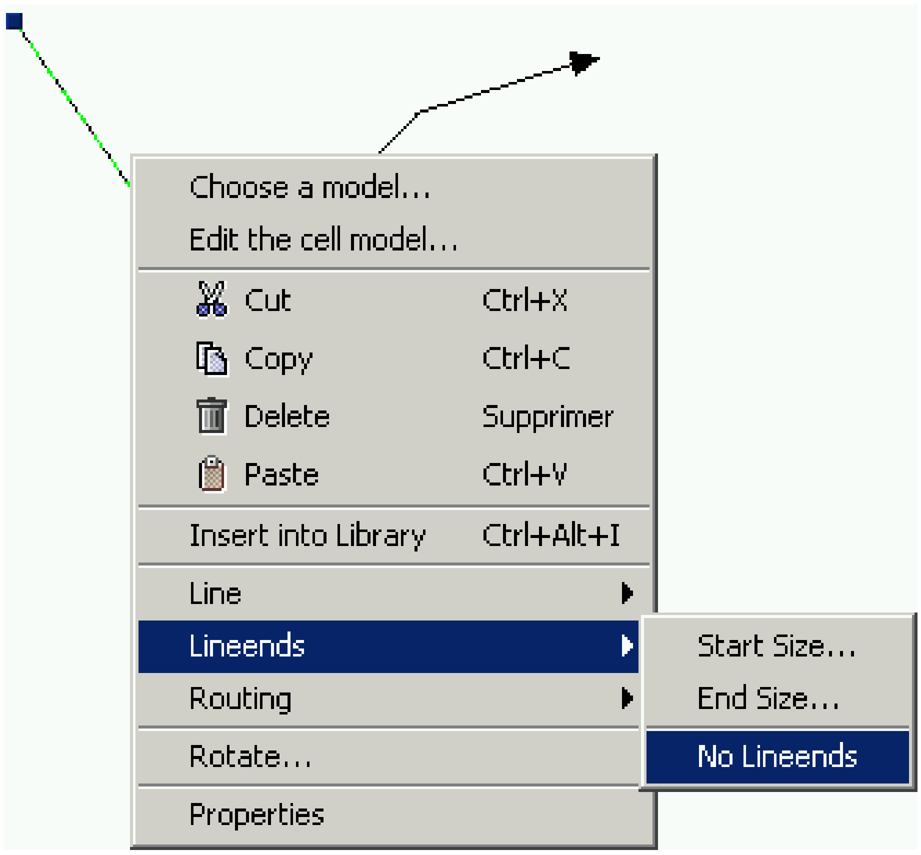
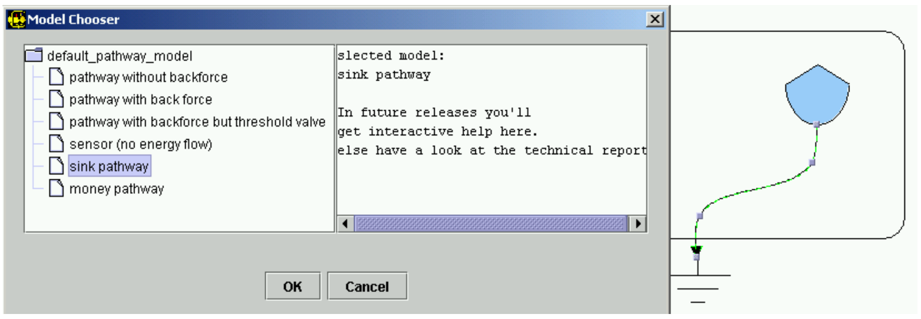


Figure 6.2: blending a pathway symbol

An other important graphical feature for edges is the possibility to delete the arrow symbol. This can be done through the popup menu:



A way to automatically fit the view of a pathway with its meaning is to choose the appropriate model for the pathway. For instance selecting a "sink pathway" model will make the pathway to be displayed with a stroke:



```
slected model:
sink pathway

In future releases you'll
get interactive help here.
else have a look at the technical report
```

Figure 6.3: making an edge to be diplayed with stroke

Notice than other operations with pathways can be obtained by the tool bars (width, color or arrows editing).

Editing text into cells can be done by double clicking on the cell. You can put text either on node either one pathways. When it's a pathway, you can then select and move the text. If that's a node, choose align options to control the text position. Notice that you can choose the police, the size, the color or the display of any text by using the tool bar.

Multi lined text can be drawn two way: either by selecting a text area cell in the tool bar, either by using HTML. The first solution isn't so good since it doesn't allow to hide the cell border neither to have a fine control of the text position. In fact, you can write in any node cell using HTML. Any HTML text will be displayed. So you can even use an HTML editor to provide the HTML code and then copy it into the cell. For simple multi lined cell: adapt this code:
write <HTML> first line < br > other line
Warning, if one line is to long compared with the cell size, your text won't be centered anyway. So use as necessary the line separator < *br* >.

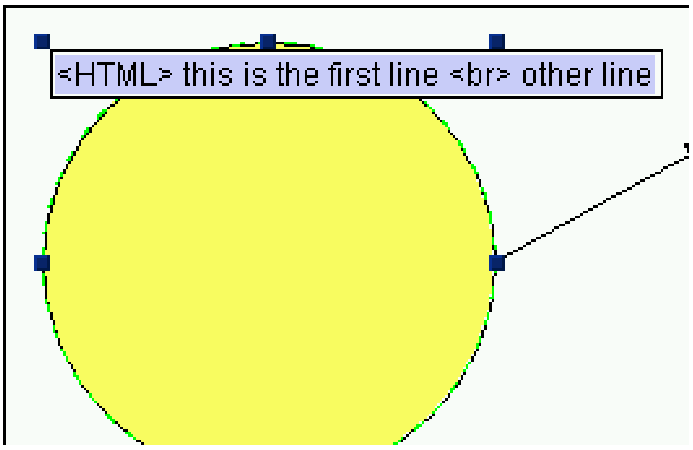

<HTML> this is the first line <br> other line

Figure 6.4: writting multi lined text in a cell

Then, thanks either to the shape popup menu or either to the menu bar, you can access lots of intuitive features like:

editing the color of the cell

editing the width of the cell

no border

rotate one or several cells

zooming

scrolling

copy and paste

select all

search

group / ungroup

send to back / send to front

align

display ruler

display grid

change magnetism

Finally, don't hesitate to use the util.lib library or to build your custom library in order to be more efficient. Once a library is opened, just select a component and drag it into the graph area.

Chapter 7

The static emergy analysis interface

Emergy Simulator has been thought to allow different level of use.

7.1 Static emergy forcing

This first level of use isn't a simulation use. Emergy Simulator simply allows you to save emergy data (that is quantities, energies, emergies and transformities) in the XML archive that will be created with your diagram. If you choose that use, then you are supposed to find your emergies and transformities by yourself with your own logic. I don't believe one second that there are other logics than the ones respecting the emergy laws, but in fact most of emergy users don't question much and simply estimate emergies by multiplying quantity flows by transformities they find miraculously in an Odum's book. I personally don't support this use because I think this discredits the emergy methodology. However, such user can have the opportunity to save their work as a compatible data bank XML file thanks to Emergy Simulator. This mature attitude could enhance slightly the peer review since it links the diagram with the data.

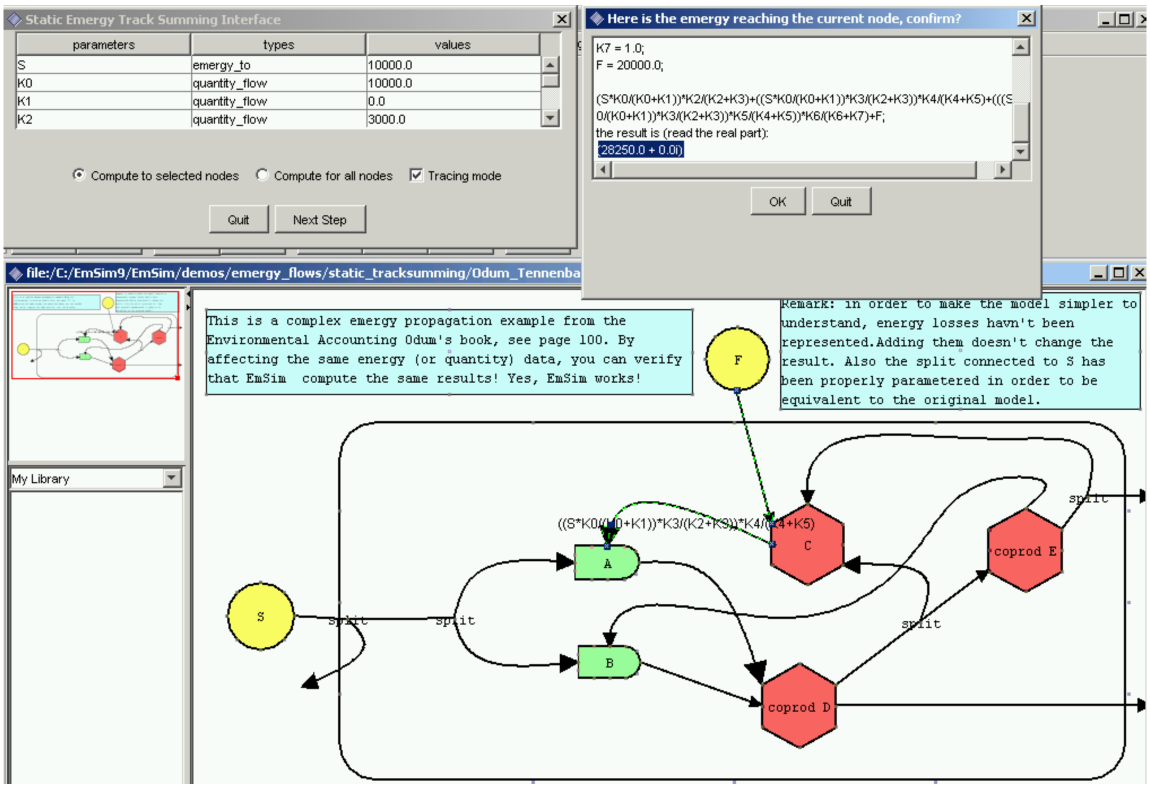
To use Emergy Simulator this way, simply edit the model of each node and pathway of your diagram. Enter your emergy an transformity values. They will be saved at the right place. And they could be used by a data bank system.

7.2 Static emergy track summing

This level is much more serious since at least you will respect the rules of energy conservation by respecting the rules of the emergy algebra defined by H.T. Odum. (see the part about the emergy in this report).

T**he minimum entries are the energy (or quantity) that flows in each splitting pathways and the emergy of every independent sources.** The best to enter those parameters is to to it after having started the simulation, inside the parameter panel.

Then when you choose simulation a static track summing is performed and emergy to nodes are computed. Here are some examples form the Odum_Tennebaum provided example:


Static Emergy Track Summing Interface

parameters	types	values
S	emergy_to	10000.0
K0	quantity_flow	10000.0
K1	quantity_flow	0.0
K2	quantity_flow	3000.0

[✓ Compute to selected nodes] Compute for all nodes [✓ Tracing mode]

Here is the emergy reaching the current node, confirm?

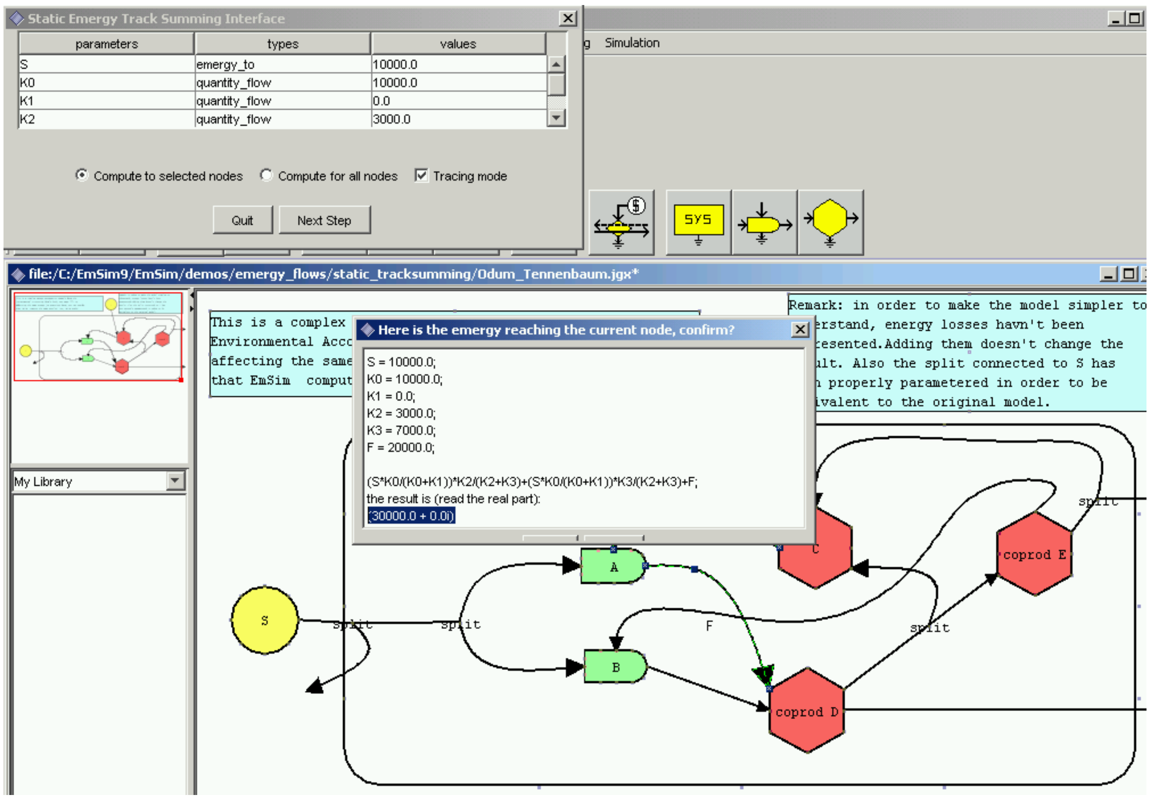
```
K7 = 1.0;
F = 20000.0;

(S*K0/(K0+K1))*K2/(K2+K3)+((S*K0/(K0+K1))*K3/(K2+K3))*K4/(K4+K5)+(((S
0/(K0+K1))*K3/(K2+K3))*K5/(K4+K5))*K6/(K6+K7)+F;
the result is (read the real part):
(28250.0 + 0.0i)
```

file:/C:/EmSim9/EmSim/demos/emergy_flows/static_tracksumming/Odum_Tennenba

This is a complex emergy propagation example from the Environmental Accounting Odum's book, see page 100. By affecting the same energy (or quantity) data, you can verify that EmSim compute the same results! Yes, EmSim works!

Remark: in order to make the model simpler to understand, energy losses havn't been represented.Adding them doesn't change the result. Also the split connected to S has been properly parametered in order to be equivalent to the original model.


Static Emergy Track Summing Interface

parameters	types	values
S	emergy_to	10000.0
K0	quantity_flow	10000.0
K1	quantity_flow	0.0
K2	quantity_flow	3000.0

[✓ Compute to selected nodes] Compute for all nodes [✓ Tracing mode]

file:/C:/EmSim9/EmSim/demos/emergy_flows/static_tracksumming/Odum_Tennenbaum.jgx*

Here is the emergy reaching the current node, confirm?

```
S = 10000.0;
K0 = 10000.0;
K1 = 0.0;
K2 = 3000.0;
K3 = 7000.0;
F = 20000.0;

(S*K0/(K0+K1))*K2/(K2+K3)+(S*K0/(K0+K1))*K3/(K2+K3)+F;
the result is (read the real part):
(30000.0 + 0.0i)
```

You can verify through those examples that even the most complicated cases of track summing are correctly handled by EmSim since its version 0.9! See

the ?? chapter for more details about this pedagogical example.

A good feature for the future would also be to let the user the opportunity to pick up some flows directly in the graph to include them in an emergy table. This table could be saved as an Excel spread sheet thanks to the java open source jexcelapi [3] package. From the table, the emergy indices will be calculated. Then basic pie chart or histograms could be made and saved thanks to the JFreeChart [**?**] package.

Chapter 8

The dynamic simulation interface

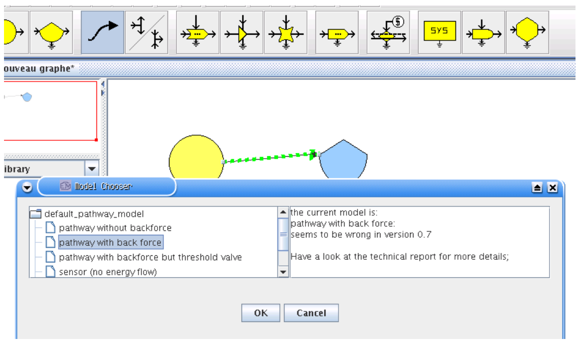
This is one of the most advanced feature of Emergy Simulator. You can indeed build dynamical system using the graphical energy language defined by H.T. Odum. For instance most of the examples from the Computer Minimodels Odum's handbook [1] will be handled by Emergy Simulator without any major modification.

To afford dynamical simulation special care should be taken with the models of the cells you'are connecting. You will also be asked to enter initial conditions (initial storages) and parameters.

1. So the first step is to draw your energy dynamic diagram while paying attention to the logic of your drawing. First draw the nodes.

2. Then connected the nodes with pathways. Be careful to connect properly the components. Some components require a fix number of connections.

3. Then, select each component one by one and choose the appropriate sub model. The first time you are choosing a sub-model, you will be asked to also choose the kind of simulation you are undertaking. Choose "dynamic quantity simulation".

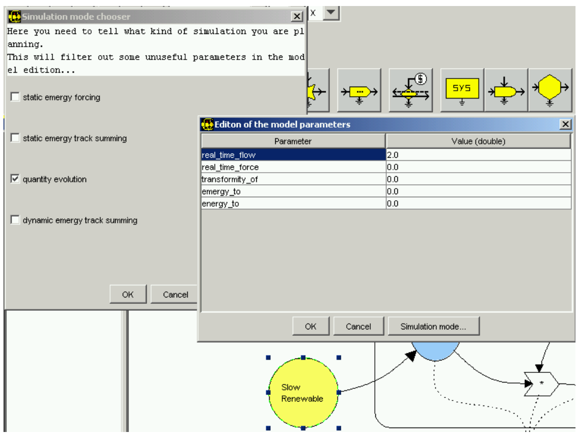


default_pathway_model
- pathway without backforce
- pathway with back force
- pathway with backforce but threshold valve
- sensor (no energy flow)

the current model is:
pathway with back force:
seems to be wrong in version 0.7

Have a look at the technical report for more details;

4. Then you will need to enter the parameters of the models. You have two alternatives to do so: either edit each model and enter the required parameters, either do it later in the simulation panel. This second method is the faster. Indeed, that way you will be sure to enter exactly the right number of parameters. You won't forget one and you won't be asked for non useless parameters. See the following section about the dynamic simulation panel fore more explanations.


Simulation mode chooser

```
Here you need to tell what kind of simulation you are pl
anning.
This will filter out some unuseful parameters in the mod
el edition...
```

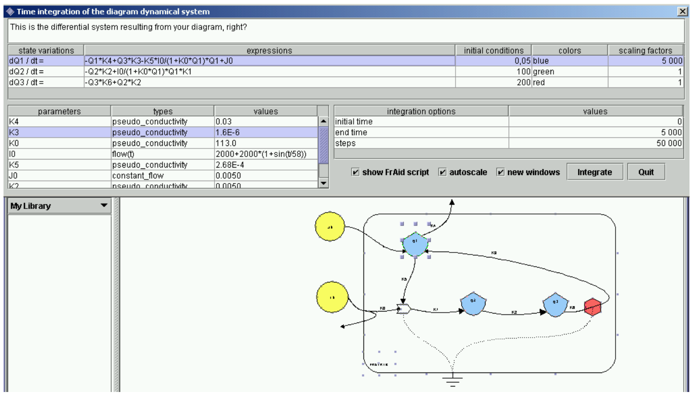
- [ ] static emergy forcing
- [ ] static emergy track summing
- [x] quantity evolution
- [ ] dynamic emergy track summing

Editon of the model parameters

Parameter	Value (double)
real_time_flow	2.0
real_time_force	0.0
transformity_of	0.0
emergy_to	0.0
energy_to	0.0

8.1 Using the dynamical simulation panel

Once the diagram is done and sub-models are chosen, in the menu bar, choose Simulation > Start a new simulation. Then the dynamical simulation panel appears:


Time integration of the diagram dynamical system

This is the differential system resulting from your diagram, right?

state variations	expressions	initial conditions	colors	scaling factors
dQ1 / dt =	-Q1*K4+Q3*K3-K5*I0/(1+K0*Q1)*Q1+J0	0,05	blue	5 000
dQ2 / dt =	-Q2*K2+I0/(1+K0*Q1)*Q1*K1	100	green	1
dQ3 / dt =	-Q3*K6+Q2*K2	200	red	1

parameters	types	values
K4	pseudo_conductivity	0.03
K3	pseudo_conductivity	1.6E-6
K0	pseudo_conductivity	113.0
I0	flow(t)	2000+2000*(1+sin(t/58))
K5	pseudo_conductivity	2.68E-4
J0	constant_flow	0.0050
K2	pseudo_conductivity	0.0050

integration options	values
initial time	0
end time	5 000
steps	50 000

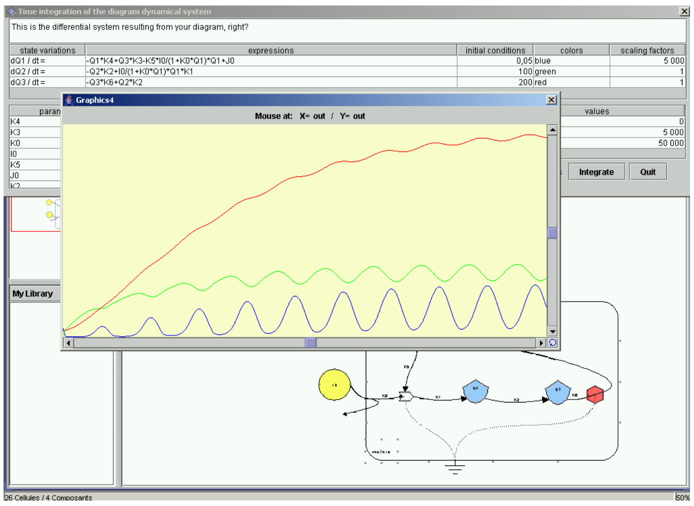
- [x] show FrAid script
- [x] autoscale
- [x] new windows

First look on the right to the differential equation: is there some explicit error? If yes:

- may be your diagram isn't correct. If there are error message, try to find out the wrong component and fix it by choosing other's models or modifying the connections.
- may be you tried to use models that aren't yet implemented in the current version of Emergy Simulator. Look in the What's is new or Readme file with your version to be sure that models are implemented. If not feel free to jump into the code to implement it by yourself. Then have a look in the developer manual.
- Finally it could be an Emergy Simulator bug. If you suspect that case then tell it to the project responsible using his e-mail or better using the forum at the sourceforge website.

If the differential system is OK, then enter or modify the initial conditions and the parameters. **when selecting a line of a parameter or an initial condition, it automatically selects the corresponding component and highlight it so you can know which one it is. If required move the simulation panel to see what component is selected.**

Finally, click on the integrate button to start the Runge-Kutta integration of your system and the plotting. After several seconds, a plotting window appear. Resize it if necessary. Probably your curves won't appear on the graph so you need to do the following: **right-click in the plotting window in order to get the popup menu, then choose autoscale**. This will scale the window for the curves to appear. Of course you can use the other FrAid options for this window and also save the picture where you want:


Time integration of the diagram dynamical system

This is the differential system resulting from your diagram, right?

state variations	expressions	initial conditions	colors	scaling factors
dQ1 / dt =	-Q1*K4+Q3*K3-K5*I0/(1+K0*Q1)*Q1+J0	0,05	blue	5 000
dQ2 / dt =	-Q2*K2+I0/(1+K0*Q1)*Q1*K1	100	green	1
dQ3 / dt =	-Q3*K6+Q2*K2	200	red	1

paran	values
K4	0
K3	5 000
K0	50 000
I0	
K5	
J0	
K2	

Graphics4

Mouse at: X= out / Y= out

Chapter 9

Getting help

Three solutions:

- read the following F.A.Q section
- ask a question or read at the Emergy Simulator forum
  http://sourceforge.net/forum/?group_id=102093
- contact me: Raphael Valyi: rvalyi44@hotmail.com

Chapter 10

Samples

The distribution of EmSim comes with a series of examples. In an EmSim session, open the examples from the following directory: EmSim/demos.

Among those examples, a large part of the dynamic examples from the Computer Minimodels [1] Odum's book are provided. It's very interesting to look how the models have been rebuilt, what components and models have been used...

This illustration work has been made at the LEIA by Antoine Markey. It's still unclear if the LEIA will provide the examples comments and instructions in the future (such data would better saved inside the SimulationSettings java bean since it will automatically write and read an XML corresponding code). If you are interested in those examples, try to have a look at the book or contact the LEIA.

Other examples deal with emergy simulation. The most famous is called Odum_Tennenbaum...

Chapter 11

FAQ

11.1 Is it possible to include pictures inside symbols?

Yes since this is a JGraph feature. With the cell popup, select the image choice and choose an image. Supported formats are jpeg, png, gif. This is specially interesting for communication and pedagogical purposes, since you can illustrate by a picture a symbol or even completely hide for the user a complex assembly...

11.2 How EmSim deals with transparent pictures?

Like JGraph does. People often like Visio to do fine presentations, but for instance EmSim handles well transparencies. If you yant to import a transparent picture, then choose a gif format. If no transparent color has been defined for that picture, do it with a picture editor. The free IrfanView does it fine for instance. When exporting EmSim pictures, you can also do it by saving as gif and selecting a transparency to apply with IrfanView. Then simply add your picture inside a pdf, word or even power point document...

11.3 How to draw mullti-line label or text?

To create multi line labels you can use HTML. Here is an example:

```
<html>first line<br>second line</html>
```

Remark: it's even possible to write almost any HTML text in a cell. If that's really important for you, then learn HTML, or copy and paste an HTML text you generated with an HTML editor.

11.4 How to use the sink symbol?

A sink symbol can be linked to any other symbol that support such a connection. To create it, select the cell you want to link to the new sink symbol, use the popup menu (select your cell and do mouse right click). Then in the popup menu select "connect to sink". This will create a sink symbol under your cell.

You can also directly connect the cell to any other existing sink symbol.

11.5 How to specify a graph cell submodel?

By default, when you are drawing using the modeling tool bar, the symbols are generic rather than specialized. The consequence is that this allow you to draw anything but on the other hand you can't run a dynamic simulation this way. To specialize your symbols, use the popup menu (mouse right click on the cell) and then select "choose a submodel". Then you we need to enter the right parameters still using the popup menu or in the dynamic simulation panel.

11.6 How to blend a pathway symbol?

You can add a pathway connector using the pathway button in the modeling bar. To blend the symbol, you should add some more points that the two extremities. Those points are called control points. To add them, simply move the cursor upon the pathway and press the mouse right click while pressing Shift on the keyboard. Deleting a control point is done by the same procedure exactly upon the point you want to delete. You can add as much control points as you need to blend the edge. Of course, you can also move the points with the mouse cursor.

Notice that sometimes, the control point you add isn't ordered correctly with the other points. This is a minor bug occurring when you add lots of points. A solution is to displace by hand the previous points in order to properly adjust the curve.

11.7 Is it possible to choose the plotting color of a curve?

Well, not yet since the FrAid plot function doesn't receive the color map trough its plotting XML argument. As a workaround, you can still change the colo of any curve once it's plotted. To do that right click on the plotting window and change the color map.

Bibliography

[1] Howard T. Odum Elisabeth C. Odum. *Computers Minimodels and Simulations Exercices for Science and Social Science.* Center for Environmental Policy - University of Florida, 1994. extracts (Portuguese version!) at http://www.unicamp.br/fea/ortega/ecosim.

[2] bondgraphs.com. About bond graphs. http://www.bondgraphs.com, 2004.

[3] Andy Khan. the jexcelapi java open source .xls converter. http://www.andykhan.com/jexcelapi/, 2004.

[4] Howard T. Odum. *Systems Ecology An Introduction.* John Wiley & Sons, 1983.

[5] Howard T. Odum. *Environmental Accounting* EMERGY *and Environmental Decision Making.* John Wiley & Sons, 1996. extracts at http://www.unicamp.br/fea/ortega/htodum/emergyaccount.htm and http://www.unicamp.br/fea/ortega/agroecol/emergy.htm.

[6] the jgraph team. the jgraph and jgraphpad open source projects. http://www.sourceforge.net/projects/jgraph, 2004.

Index