In Ortega, E. & Ulgiati, S. (editors): Proceedings of IV Biennial International Workshop "Advances in Energy Studies". Unicamp, Campinas, SP, Brazil. June 16-19, 2004. Pages 349-360

EMERGY SIMULATOR, AN OPEN SOURCE SIMULATION PLATFORM DEDICATED TO SYSTEMS ECOLOGY AND EMERGY STUDIES

Raphaël Valyi[1], Enrique Ortega[2 2]

[1] Master of Science and Master of Materials Engineer, Ecole Centrale de Lyon, 36 Av Guy de Collongue 69134 Ecully, France. E-mail: raphael.valyi@ecl2004.ec-lyon.fr

[2] Professor, Laboratory of Ecological Engineering, Food Engineering School, State University of Campinas (Unicamp), CP 6121 Campinas, SP, Brazil. CEP 13.083-862

E-mail: ortega@fea.unicamp.br

ABSTRACT

To develop more objective and more efficient criteria for judging sustainable development, the interactions between human activities and the environment are being simulated using various forms of *systemic modeling*. H.T. Odum was a pioneer in applying modeling concepts from electrical circuitry and irreversible thermodynamics to systems ecology. The Emergy Simulator (EmSim) project is a computer implementation of the main concepts of H.T. Odum's Energy Systems Language. First, with EmSim it is possible to share models through the Internet or export them as drawings. Second, EmSim is able to translate Odum's Energy Systems diagrams directly into a set of ordinary differential equations that can be integrated and plotted. EmSim also can correctly compute emergies and transformities in energy networks. Finally, EmSim is an open source project that can be improved by other people interested in these kinds of calculations. Everybody is welcome to try it, learn more about it and participate in its implementation at: http://emsim.sourceforge.net .

1. INTRODUCTION

We prepared Emergy Simulator [1] to overcome some Systems Ecology limitations:

(a) Systems ecology models can obtain more acceptance only if there are widely documented for better comprehension and discussion. But sharing models requires standards. The energy systems diagram language of H.T. Odum is already a powerful standard but it needed an open source implementation and the use of specific computer file formats.

(b) A specific drawing tool was also required to communicate about models without depending on commercial software.

(c) Moreover, when Odum's energy diagrams stand for dynamical systems, researchers are compelled to translate graphs into differential equations systems but not always following the same rules, using commercial software like, MathLab or Excel.

(d) Very few researchers are able to compute the transformity of a product within a complex production process. As a result, there is an excessive use of values available in transformity tables and the original idea of emergy can be missed. This abuse leads some to question the accuracy of emergy indicators! Indeed, a transformity value is always dependent of the specific system where it was measured. Picking a transformity from one system and injecting it into another implicitly assumes that the products are made exactly the same way. Then you maybe start forgetting or double accounting some emergy. Remember the rules stated by H.T. Odum in [5] chapter 6. One kilogram of bananas have more emergy in Europe than in Brazil because of incorporation of transport energy.

(e) During ten months in 2004, the most relevant algorithms used by H.T. Odum for systems simulations were coded in the EmSim project. In order to continuously improving the project using the academic research works of others, it was conceived as extensible, participative and modular. Because the task is huge and almost with no funds, the best solution is a cooperative project. Therefore EmSim is conceived as an open source code (GNU GPL license) and accessible on the Internet through a Concurrent Version System (CVS) while providing all the facilities of modern software development like forums mailing lists and bug tracking.

(f) Moreover, it is programmed in Java 2 [2] so that it can run on line and can be easily coded on any computer under any operating system; for free.

This article deals with the main features of EmSim. To explain the algorithms, we will often refer to the traditional modeling done in systems ecology. For further details, we advise the reader to look at [3] and [4] for causal modeling, energy language and its dynamical meaning, and at [5], chapter 6 for the basics of emergy algebra.
EmSim allows ecological/economical modeling and the steps can be summed up as: searching laws, representing the model, mapping available information into the model, simulating the kinetics of the model and/or qualifying the thermodynamic efficiency of the model.

A series of steps has to be followed for modeling phenomena. Fix the spatial and temporal boundaries of the system as well as state the accuracy used in accounting for things. Find a pertinent basis for parameters to qualify the state of the system (e.g. state vector). Of course, these two steps remain subjective and intuitive. After that, it is necessary to investigate what resources are required to make the outputs, both qualitatively and quantitatively (the steady state energy intensity or the dynamics of the system) studying the causal links between phenomena by means of intuition, observation and even inferential statistics.

- Now it becomes possible to draw a causal network as an energy systems diagram explicitly documented qualitatively and also possibly quantitatively (using classical thermodynamics) describing the causal pathways so that specialists from various fields can discuss it. EmSim is a diagram editor that performs this function, see part 2.
- Models should be easy to share and remain open for further modifications, improvements and reviewing to make it possible to achieve consensus and credibility within the scientific community. This can only be achieved by means of standardized data, EmSim stores every model as an XML exchange model, a very useful standard. To learn how EmSim structures the systemic knowledge and stores it as XML, read the part 3.
- Once the network of causality is established, the kinetics and the dynamics of the model can be studied. The kinetics are defined by "set of ordinary differential equations". In [3] and [4] Odum gives various examples of how the "energy systems language" can be translated into a system of differential equations. EmSim automates this translation and makes it possible to plot the evolution of the system given an initial state, see part 4.
- Concerning the irreversible thermodynamics, the idea is quite simple: at any given time, whether at steady state or not, it is interesting to check how much emergy is required to maintain a resource, what is its transformity or even what is its quality (organization) measuring its empower and renewability. EmSim is able to calculate the emergy content in complex networks, see part 5.
- EmSim is incomplete and has some drawbacks, it demands additional work: see part 6.

2. EMSIM AS A DEDICATED DRAWING TOOL TO COMMUNICATE WITH ENERGY SYSTEMS DIAGRAMS

Basically, EmSim is an Odum Energy Diagrams editor that allows diagrams to be saved with plenty structured information and is able to export them as pictures. Instead of spending time re-inventing the wheel, our philosophy was to use object oriented programming and to derive the graphic EmSim features from existing standard tools, see figure 1.

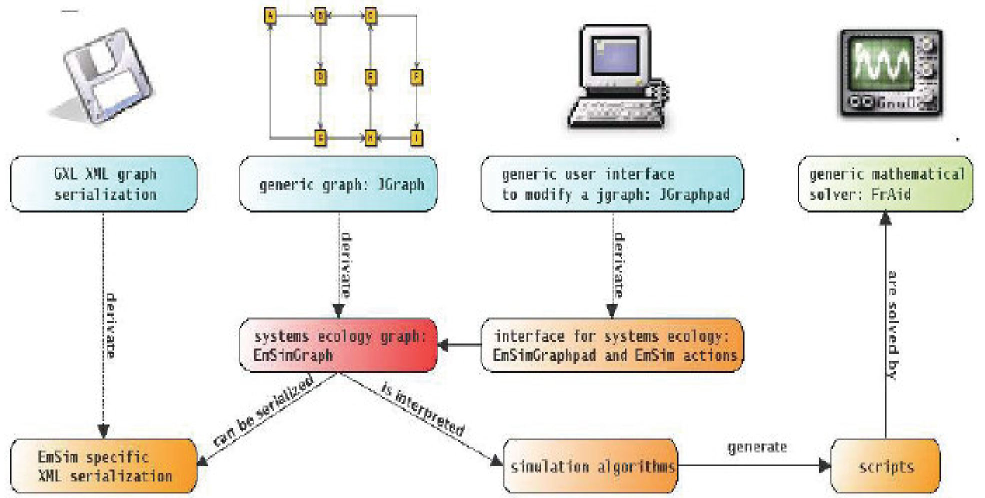

Figure 1. Obtaining the EmSim features by subclassing standard java open source objects

EmSim uses graph data models and visualization features from the JGraph library and invokes actions from the JGraphpad interface (see [6]) which acts as a controller over this MVC (Model View Controller) design pattern. We mainly sub-classed some objects and we also included a toolbar for the "energy systems language symbols". Finally, commercial software is no longer required to draw models as energy systems diagrams; instead good quality figures can be created intuitively using the EmSim diagramming tools. Its user interface allows drawing complex and elegant graphs very easily by dragging and dropping the required components from a toolbar. Among the graphical features are: group/ungroup, copy/paste, undo/redo, curved connectors (spline), color choice, transparency in gif pictures cell rotation ability, split (un)aggregated view and code lines.

3. EMSIM STRUCTURES THE KNOWLEDGE OF SYSTEMS ECOLOGY STUDIES, PROVIDING AN IDEAL USER-INTERFACE OF INFORMATION SYSTEM

3.1 Ecology researchers need a file format to communicate more efficiently

Energy analysis and systems ecology researchers are spread all over the world and very often they are working separately. They seldom reuse the work already done by others on input/output exergy, life cycle assessment or ecological footprints. Many data could be more efficiently used sharing information and methods (see [7] and [15]). It seems clear that a communication format to share data across the Internet is necessary, at least between emergy researchers. Indeed, often published papers don't provide all the elements required to get the results! For instance, many diagrams are drawn in a manner that makes it difficult to determine whether an outflow is a flow split, a by-product or even a hybrid case. Consequently, peer review, reuse, and improvements of existing work are jeopardized. What could happen if people were able to exchange standardized extensible data on energy systems that everyone could share just like people already do in other fields like Computer Aided Design? What if emergy knowledge could be formatted and available in a data bank for use in deducing statistical laws for the considered systems? Just imagine how this could improve the accuracy and credibility of emergy methodology. After drawing an energy diagram with EmSim, you are free to enter all the information you judge necessary exactly where it should be thanks to the XML formalism! Associate the energy or matter flow directly with the corresponding pathways. Explicitly mention the allocation rule you use at each divergence of flow. If a pathway has a flow proportional to the gradient of the forces provided at its two extremities, then simply choose the corresponding sub-model for that pathway! If a parameter that you enter has a Gaussian distribution, then associate such a distribution with the parameter to allow someone else to perform a stochastic simulation in the future…

3.2 The data structure handled by EmSim

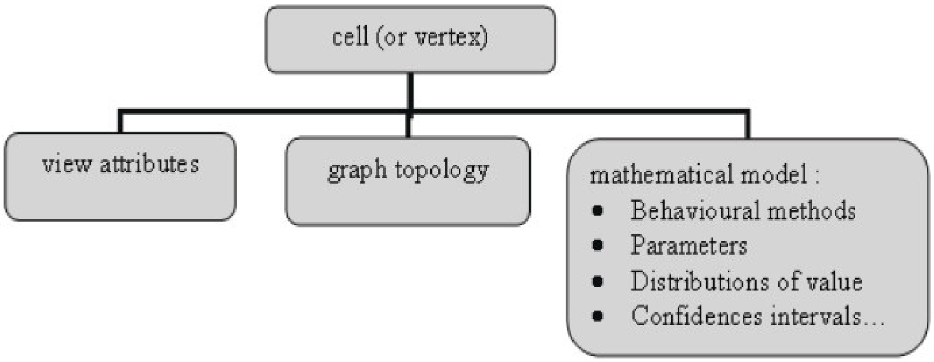

Figure 2. Information stored in an EmSim cell

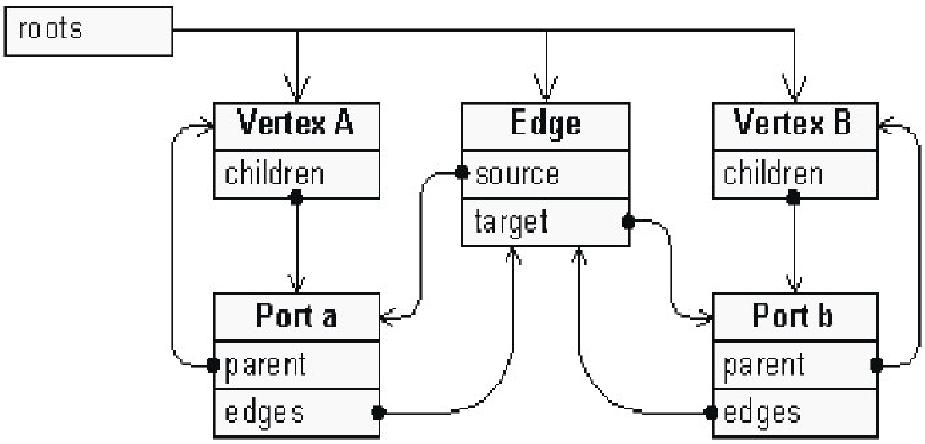

Figure 3. The graph topology using the JGraph [6] open library

The data structure handled by EmSim essentially consists of a graph made of nodes and edges that respectively stand for products and energy pathways. The user interface allows one to associate information with each element of the graph. Thus, each cell or node keeps track of the information about its view, but also about the graph structure and about its mathematical model; *see Figures 2 and 3*.

3.3 Storing efficiently systemic data as XML

The data structure generated by the user has to be persistently saved (serialized). The key strategy is to use a special file format relying on:

- Each piece of information being hierarchically organized;
- Describing with markups all pieces of information that you store;
- Not being dependent of the application you use;
- Being humanly readable;
- Being expendable while keeping compatibility with all applications.

So EmSim rigorously saves the graph structure using the XML formalism (see [9]). This means that the way EmSim writes the file as a text respects some standard conventions making it XML and thus very easy to plug in to other applications like a data bank or data mining. In this way, you could perform statistical analysis on emergy data and you could also make emergy data inter-operable with widely collected life cycle assessment data! The XML format is currently being smoothly migrated (to come in the version 2.0) in order to use the most powerful and the easier to maintain standard: encoding Javabeans collections using the XMLEncoder and XMLDecoder java Sun framework (see [1]). None of the information you enter (either by means of the graphical user interface or with a text editor) will be lost in the future. Indeed, even if the file format of EmSim files were to change a bit, then an XSLT transformation will bring your old data back to life. XSLT could also be used for interoperability with other simulators!

4. EMSIM TRANSLATES CAUSAL ENERGY DIAGRAMS INTO ORDINARY DIFFERENTIAL EQUATIONS (ODE) SYSTEMS

4.1 Statement of the problem

Systems Ecology [3] and Computer Mini-models [4] are two of H.T. Odum's publications that include lots of "energy diagrams" representing the structure of the differential equations used to model real systems. Odum wrote he created this graphical language to allow specialists from various scientific fields, ranging from sociology to ecology, to collaborate more efficiently in modeling human activities. However, few people seem to understand how this graphical language works! Is there a synthetic explanation for that?

Through a careful reading of the publications mentioned, we found that Odum's graphical language for dynamical systems was something very powerful and very universal, because the diagrams are directly translatable into a mathematical structure. We also found that a similar language (bond graphs, see [10]) is already widely used in other scientific fields. Then, by analyzing examples, we intuited the laws that make up the kernel of the language. The difficulty here was to re-formalize the rules so that they can be applied in the most general manner possible without requiring many specific cases. Second, the task was to implement that kernel inside EmSim. Finally, Odum's energy language is suitable for modeling dynamic systems where the functioning (differential equations) and a particular state (the initial conditions) are known. However ODE systems only apply for spatially discretized dynamical systems in contrast with partial derivative systems that would fit better for continuous spatial domain applications but still would require more advanced mathematics, like finite element integrations.

4.2 How EmSim converts causal energy diagrams into ODE equations

4.2.1 General equivalence between dynamical equations and graphical symbols in EmSim:

Given a causal energy diagram, it is possible to get the equivalent differential equations describing the evolution of the system. Like other simulation software, EmSim is able to perform such a translation. Here are the basic equivalences:

- Real dynamical system <-> "energy diagram" model;
- Set of state variables <-> set of storage symbols;
- First order variation of a state variable <-> set of edges connected to a storage;
- One term of the variation <-> the flow inside one pathway.

Then EmSim recursively rebuilds the differential system by "recursively reading" a diagram:

- For each capacitive storage symbol, (flow conservation); it writes:

$$\frac{dQ(t)}{dt} = \sum connected\_flows \,.$$

- For each inductive storage J (force equilibrium, little used in ecology), it writes:

$$\frac{dJ(t)}{dt} = -\sum connected\_forces \,.$$

Then, the simulation will consist of estimating at each time, t, the quantity for each storage, and plotting the evolution.

$$Q(t) = \int_{t0}^{t} \frac{dQ(t)}{dt} dt \text{ and } J(t) = \int_{t0}^{t} \frac{dJ(t)}{dt} dt$$

This is achieved using a classical 4th order Runge-Kutta method; see [11]. Source symbols behave like storages but their variations are neglected compared to other variations within the temporal scope of the simulation.

4.2.2 How flows along pathways are computed in dynamical energy language:

The flows into and out of a storage node component determine the first order variation of any state variable. Therefore, to get the freedom of modeling any ODE system, the user should have the freedom to generate any flow inside a connection. This is possible because the flow inside an edge is computed in one of several ways, depending on the following cases and always according to the so-called "integral causality":

- Flow imposed by the source component of the pathway $Flow = Inflow$
- Flow demanded by the target component of the pathway $Flow = Outflow$
- Classical pathway: $Flow = conductivity * Inforce$
- Sensor: $Flow = 0$; but the information is transmitted
- Gradient pathway: $Flow = conductivity * (Inforce - Outforce)$
- Gradient valve pathway:
  If *Flow > Threshold* then *Flow = conductivity * (Inforce – Outforce)*
  Else *Flow = 0*

4.2.3 How forces at the component connections are computed:

We explained that flows inside connections determine the terms of the expression for the first order variation of any state variable. However, it appears that lots of flows are actually driven by forces (also according to the "integral causality" of the integral-differential constitutive equations of the considered components). Once again, to model any ODE system, the user should have the freedom to generate any force function. This is possible because nodes other than storage allow building interactions. Basically, a storage provides a force proportional to its quantity at the time t. Examples are a water tank were the bottom pressure is proportional to the height of the water; other storages are electrical capacitors, mechanical springs, predation pressure and so on. Then, you can modulate and combine those forces using additional graph components:

- Regulations (2 ports, unary operator) $Outforce = F(Inforce, time)$ [1]
- Interactions (3 ports, binary operator) $Outforce = F(Inforce1, Inforce2, time)$
- Composite: a complex component. For instance, in a transaction component, one flow is the slave and the other is the master.

One of the great advantages of EmSim is that it allows the user to specify the functions used by customizing components. EmSim uses the parser of the FrAid solver to evaluate customized expressions made of standard mathematical formulae (e.g. polynomials, trigonometry, logarithms, and Boolean expressions).

4.3 Additional remarks about dynamic causal modeling

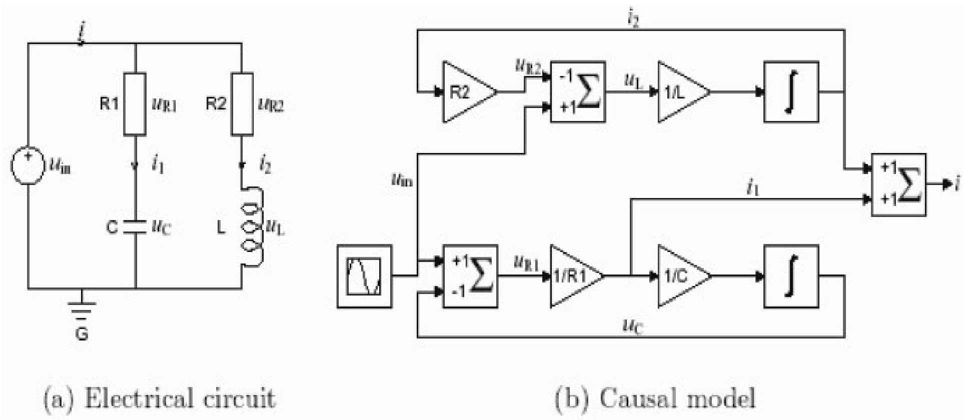

Figure 4. Differences between behavioral and causal modeling (here with Simulink)

When a storage has to experience a discontinuous variation, EmSim models it using the switch component; it makes the slope of the variation to be "almost infinite" at the discontinuity. Contrary to behavioral modeling, causal modeling lets the user accomplish the entire job of translating a real system into get a causal description using only differential equations of first order *(see Figure 4)*. But non-causal modeling algorithms (inferring the causality depending on the solicitations) are too complicated for an open source perspective. And moreover, the causal modeling is much more suitable to perform emergy analysis latter.

[1] Inforce stands for the force provided by the component from where the flow is coming whereas Outforce is the force provided by the current component.

The polymorphism of object oriented programming allows the various component models to have their own flow or force functions while being processed the same way when rebuilding formally the differential system. And the user interface allows switching the mathematical model of a graph cell and choosing it inside the inheritance tree of the models. Thus, for instance, a variable force source inherits from a constant force source but adds the possibility of entering a custom time dependent force function (f(t)=sin(t)*exp(-A*t) for instance).

4.4 Examples

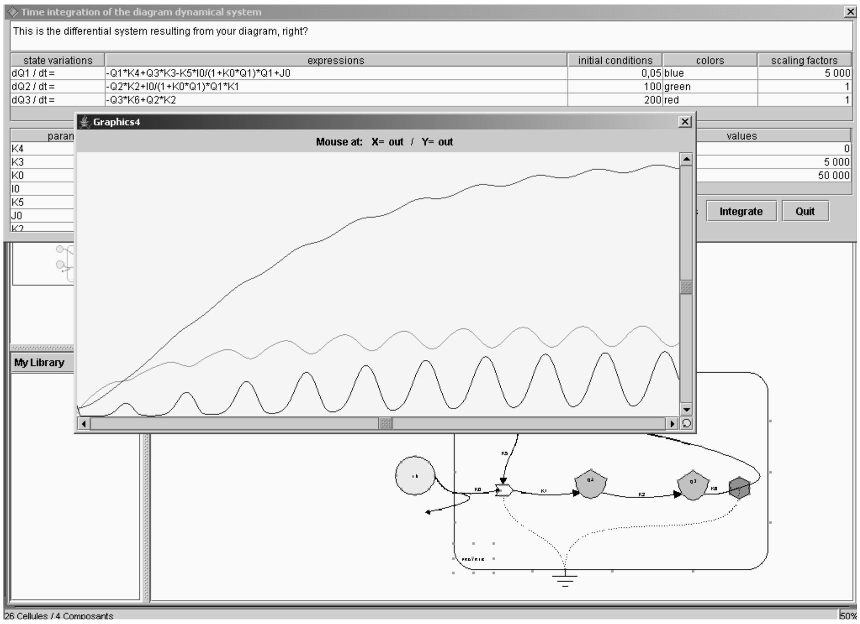

Figure 5a. EmSim successfully simulates the OSCILLAT model form [4]

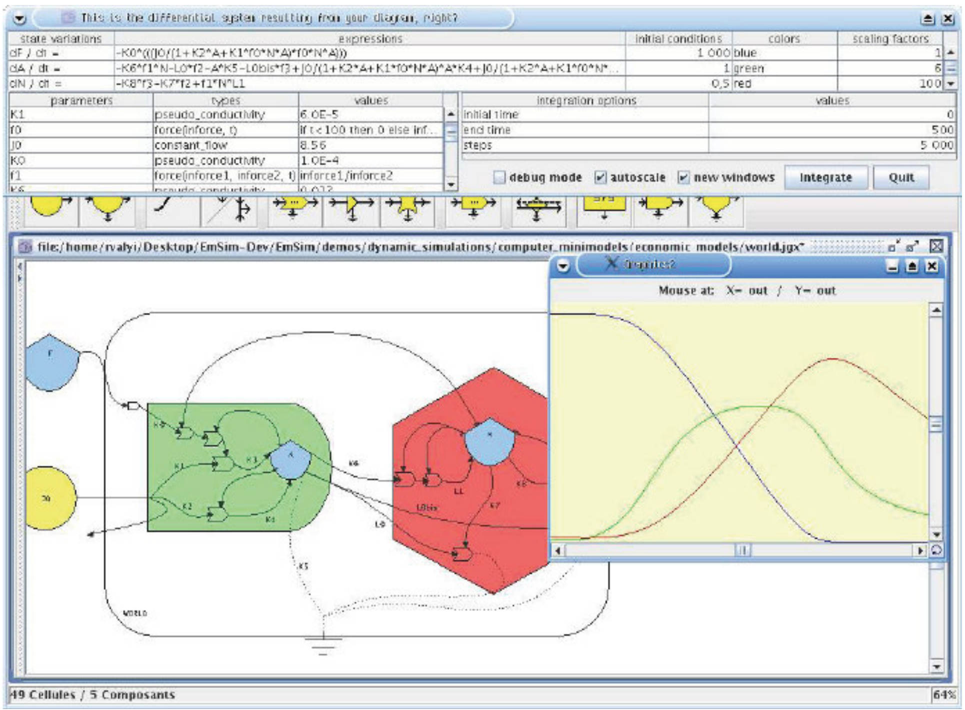

Figure 5b. EmSim successfully simulates the WORLD model from the Odum booklet [4]

For pedagogical purposes at the Laboratory of Ecological Engineering (Unicamp) a set of examples from the Computer Mini-models booklet is provided. to teach modeling and simulation.

These examples validate the ability of EmSim to represent dynamical systems since we found the same results as those given in the booklet. See *Figures 5a and 5b* for instance. As a result of teaching and research efforts a new set of examples is under construction and they will be available online at the Sourceforge website.

5. EMSIM PROPAGATES EMERGY ACROSS LARGE AND COMPLEX NETWORKS

5.1 Statement of the problem

Another very important issue when dealing with sustainable development is whether or not an alternative production process requires less energy. The emergy methodology is a great modeling tool to study this problem. In this perspective the EmSim modeling convention is:

- An arrow pathway stands for: "that the source component is required to produce the target component, at least within the considered system" (causality).
- Flows of resources can split to support various processes. In this case the required emergy for each process is split in the same proportion as the energy (or often matter) flow. An example would be a model where the oil production is partitioned to support various activities.
- Some reactions are structurally co-productive. An example is a process breaking a molecule in two parts. In this case, once you determine the emergy (by properly summing all the available energy inputs) needed to provide one by-product, then the other is an automatic consequence: it would be an error (of double accounting) to say that this other by-product requires some more emergy to be provided. Thus, whenever the production of a resource requires several products that are in turn by-products of the same process sometime before, then only the largest emergy contribution should be counted (see [2] and [8] for more details). This issue explains why a computer aided monitoring is absolutely essential.

Considering the rules for emergy accounting, EmSim builds up the formal expressions by propagating the emergy across very complex networks. It then asks the user to fill in the minimum required inputs (independent emergy sources and splits ratios). Moreover it highlights the different pathway contributions for each target component of the graph!

Compared to the matrix approach [13], EmSim produces the same results but is much more user friendly and information system oriented.

5.2 The emergy track summing algorithm

Once the graph topology and energy inputs are defined (e.g. available energy intensity at flow splits and independent emergy inputs), then EmSim formally estimates the emergy (also called "source requirements" or "accumulated exergy" in this formal perspective):

For each target component we want to compute the emergy {
  For each independent emergy source {
    A tree of all pathways that bring some emergy to the target component is built by iterating the graph: we follow the directed pathways starting from the source and

keeping only the pathways that actually reach the target component. When a loop is encountered, the pathway ends when at the component where the crossing occurs. At the same time, the emergy brought by the source is propagated across the energy pathways according to the following allocation rule: the required emergy is always conserved along a pathway except when an energy split is encountered. In this case, the emergy flows in the various branches according to the energy proportions getting in all branches. This includes the co-production case where the emergy of each by-product is equal. Now for our target component, we have got a set of pathways bringing some emergy from the considered source. But NOT ALL those emergy components should be added to determine the required emergy to support the resource under analysis because that could DOUBLE COUNT the emergy of inputs that were co-produced! To avoid this we need to filter out the double counted emergy.

So for each pathway bringing some emergy to the target component {

We look at the pathway tree and sort the pathways in sets that have been co-produced. For each set of by-products {

We only keep the maximum of the emergy components. Indeed, by-products have the same emergy after exiting the co-production process but after that, only a portion of the co-produced flow might actually support our target component; this occurs when the flow of by-product splits before reaching our target component.

}

}

Then all those independent emergy components are summed and stand for the emergy the current source brings to the target component.

}

Then those emergy components of the various sources are summed and stand for the emergy required by the target component under analysis. Indeed, the emergy brought by the various source can be summed without concern because according to the modeling convention those sources are considered *de facto* independent.

}

Finally, the transformity of each resource that flows inside the system can be computed by dividing the emergy of the product exported via the considered pathway by the exergy intensity of that pathway. Thus computing, the transformities calculation will require much more data (exergies) from classical thermodynamics but the transformities might be used to obtain emergy indicators do discuss sustainability of systems with more confidence.

5.3 Examples

Again, to demonstrate the ability of EmSim to analyze the propagation of emergy through a network, an example is provided as a demo file. We validated our algorithm on a famous case study. *Figure 65* is a screenshot from H.T. Odum ([5], page 100), also mentioned by C. Giannantoni ((14] page 30).

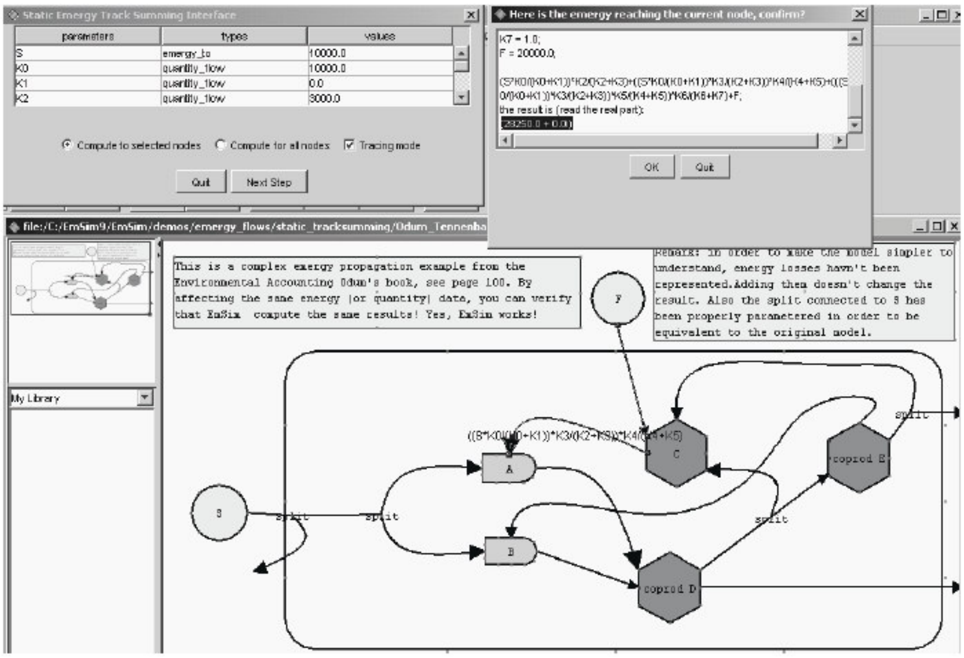

Figure 6. EmSim successfully computes the emergy on example from Odum [5], page 100

5.4 EmSim experiments could be used to check the consistency of the maximum empower theory.

The work of C. Giannantoni [14], mentions that empower studies could overcome the limitations of current emergy based indices. Indeed emergy indicators often suffer the followings limitations:

(a) Low credibility because inputs from many sources are *de facto* assumed independent and because emergy values simply come from pre-computed transformities available in tables without special investigations to see if those transformities can indeed be used in the case under study and also without serious care about confidence intervals.
(b) Source independence verification (but information systems like EmSim could pay off!). Long run positive or negative feedbacks of the investigated product upon their own production is barely modeled nor considered by those indicators.
(c) Huge sensitivity to the structure of the chosen model (see [7] and [15]). Especially, saying that products are fully co-produced is often a huge and common approximation tending to bring over estimated emergy costs, while at a more detailed scale, only small parts of those products are really by-products. Not to mention, this distinction is also affected by our knowledge of the process.
(d) Narrow validity scope. Indeed, emergy indices are relative measures. They aren't absolute because they are issued from computations performed on conceptual models and are impossible to verify most of the time. So before comparing emergy based indicators for two alternatives, one should ensure that the two models are comparable (it should even be the same model but connected differently in the ideal case).
(e) A theory of value based on emergy is a possibility proposed by Odum and others. This opens a very necessary discussion. Of course ecosystems faced an optimization through evolution that lead to the highest efficiencies at maximum emergy flows and therefore the emergy of any resource directly reflects the controlling power it has on maintaining the whole environment (see Odum [3]

page 253 and [7]). But at the same time, such a theory would have to explain the present crisis. Somehow, good ecological explanations would have to make sense in a human society with limited values that proceeds in an irrational way in the short run making us waste emergy, by destroying biodiversity and menacing mankind's existence!

While it could inherit some of those limitations, the empower of a system seems to be a more objective measure. Nevertheless, it is still dependent on the basic structure used for modeling.

For further investigations, EmSim has the potential to be an adequate platform to realize empower computations according to the formulation made by C. Giannantoni in [14]. Moreover, we made EmSim capable of propagating emergy in a network under dynamical conditions, *i.e.*, where energy or matter flows are driven by the differential system associated to the causal energy diagram. Giannantoni suggested that results could be wrong and it would be necessary to introduce emergy accumulation terms in the equations as he did in [14]. We don't really know too much about this scientific issue but we guess that the research could be undertaken using EmSim.

Finally, Giannantoni also suggested that dynamic empower studies should involve fractional calculus and "incipient derivative operators". EmSim could build fractional derivative equations by reading graphs using similar recursive programming to that used for generating integer-order, derivative, differential equations. However, the numerical simulation of those equations would also require the implementation of a fractional calculus solver. But so far, there are no applications of this method in empower studies.

6. EMSIM IS FULLY OPEN FOR FURTHER DEVELOPMENT BUT REQUIRES NEW RESEARCH PROJECTS

Commercial simulators are not always a suitable alternative, because their code is not available for modification by the emergy community and in many countries, the scarce economic resources could be better allocated using free alternatives. In contrast, EmSim has many advantages, foremost its code is open and freely available at: http://emsim.sourceforge.net. This site also provides a discussion forum and lots of tools for team development: CVS, bug database, feature trackers and so on. Many people can collaborate improving a part of EmSim, correcting a bug, providing a new translation or more examples.

Desired tasks are listed at: http://sourceforge.net/pm/?group_id=102093. Even when a code design doesn't seem suitable, we should consider "refactoring" the code which allows simply correcting or extending huge amounts of code. For instance, the JGraphpad library, on which EmSim is based, wasn't very suitable for sub-classing, but after entering the JGraphpad project forum and committing to decisive changes, every potential sub-classing application (like EmSim) can now benefit from the extended perspectives.

And finally, when emergy research matures, EmSim could become the perfect user interface to collect and share systems ecology data. Taking this in mind, systems ecology researchers would have to publish their models as EmSim files and an appropriate server could collect them before data could be handled and transformed into knowledge though data mining techniques (see [16]). Currently those files can be sent to the authors, but in the future they would be put on line as demonstration files.

References

[01] Sun Microsystems. Java, a platform independent and object oriented programming technology. http://java.sun.com.

[02] Raphaël Valyi. The emergy simulator open source code project. http://emsim.sourceforge.net, 2004.

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

[04] Howard T. Odum Elisabeth C.Odum. *Computers Minimodels and Simulations. Exercises for Science and Social Science*. Center for Environmental Policy - University of Florida, 1994, extracts provided as samples file with EmSim at. http://emsim.sourceforge.net,

[05] 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
http://www.unicamp.br/fea/ortega/agroecol/emergy.htm.

[06] The jgraph team. the JGraph and JGraphpad open source projects. http://www.sourceforge.net/projects/jgraph, 2004.

[07] Jorge L. Hau Bhavik R. Bakshi. Promises and problems of emergy analysis. *Ecological Modelling*, 2004.

[08] Jorge L. Hau Bhavik R. Bakshi. Expanding Exergy Analysis to Account for Ecosystem Products and Services. *Ecological Modelling*, 2004.

[09] XML.org. XML formalism and tools. http://www.xml.org.

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

[11] William H. Press Saul A. Teukolsky William T. Vetterling Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[12] Ivaylo I. Iliev. Fractal aid, a java language to process and plot mathematical data. http://www.fraid.org, 2004.

[13] Dennis Collins Howard T. Odum. Calculating transformities with an eigenvector method, 2000.

[14] Corrado Giannantoni. *The Maximum Em-Power Principle as a basis for Thermodynamics of quality*. Servizi Grafici Editoriali snc, 2002.

[15] Robert. A. Herendeen. Energy analysis and emergy analysis: a comparison. *Ecological Economics*, 2004.

[16] Weka: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/