Table of Contents Go Back

Why and How Should We Replace the Tank-Pipe Analogy of our Stock-Flow Models by a Chemical Process Metaphor

JACQUES LEFEVRE [1,2,3]

[1] IDEA.SIM LTD, Harrow, UK
[2] Centre for Information and Measurement in Medicine, City University, London, UK
[3] Equipe de Recherche en Génie Industriel, Ecole Centrale, Lille, France

A paper devoted to the memory of B. Richmond: On the shoulders of a giant!

Abstract: At the 2002 Congress of the System Dynamics Society, the author presented a new stock-flow diagramming language replacing the tank-pipe analogy classical in System Dynamics (SD) by a generalised chemical reaction. He argued that its use results often in better, more intuitive and less spaghetti-like stock-Flow diagrams (SFD). The paper was very well received by its reviewers. However, its presentation at the congress was disappointing since, despite its confrontational character, it elicited almost no reactions. With hindsight, it became evident that the author had focused too much on the new language and not enough on a detailed analysis of why the current method may be unsatisfactory. Henceforth, the present paper presents this detailed study of the SFD language used in SD. We restrict our attention to the parts of a SFD which represents flows of matter-like entities (tangibles). Focusing on models in population dynamics we begin by presenting their basic problem: visual complexity due to a spaghetti-like aspect. We then argue that, in addition, these models do not match well the structure of our mental models and thus lack intuitiveness. We then show that a new metaphor, replacing the tank Pipe systems of our classical SFD by generalised pluri-molecular chemical reactions called (N,M)-processes eliminates all the problems discussed. We present a way to combine a set of (N,M)-processes in a map displaying their interactions intuitively and parsimoniously. Finally, we argue that the use of these maps is not restricted to chemical or population dynamics models but may be used very fruitfully in other applications like for instance in Business Dynamics. We end up by referring the reader to our previous Palermo paper for a discussion of the translation of these maps in fully fledged computational models called "Kinetic Process Graphs or KPG" which, in our opinion, should supersede the stock-flow diagramming method currently used.

Key Words— *Stocks and Flows, Visual complexity, Intuitiveness, Tanks-Pipes and Reaction Analogy, Kinetic and Bond Graphs*

1. Introduction: Stock Flow Diagramming and Its Problems

> *Perhaps our ultimate understanding of scientific topics is measured in terms of our ability to generate metaphorical pictures of what is going on. Maybe understanding is coming up with metaphors.*
>
> *P. Bak; How Nature Work: the science of self-organized criticality (p.50)*

In an influential paper, B. Richmond states forcefully what building a System Dynamics (SD) model is: *«Operational thinking represents the essence of System Dynamics. One does it when one gets at the core stock-flow infrastructure of a system »* [1]. The graphical language used to specify Stock-Flow Diagrams (SF-diagrams) is thus a key component of System Dynamics. Invented by Forrester [2] and based on a metaphor of tanks and pipes, it has known many extensions. For instance, discrete elements (queues and ovens) have been introduced to extend its modelling range. To decrease the complexity of large models, today's excellent modelling packages use features like sectors, bundled flows and connectors, sub-models and array variables. Finally, they use also many tricks like space compression and ghosts to make the life of their users easier. Modern SF-diagrams are thus very sophisticated and allow their users to develop a wide range of applications. However, by far and large, we still draw our SF-diagrams as Forrester did in [2]. At this time, SF-diagrams were the first graphical language for the class of complex socio-economic systems which was and remains the main application domain of SD. Its creation was thus an epoch-making event. Since then, it has become clear that its concepts are simple to learn. It has also been proven that its basic concepts (stocks, flows, converters, feedback loops) are fundamental to any modelling enterprise devoted to the range of applications considered in SD. The high value of SF-diagrams is thus indisputable.

Nevertheless, as tools to express our mental models, SF-diagrams suffer from a severe problem. Every SD practitioner has indeed learned the hard way that they become easily spaghetti-like as soon as their sizes increase. Again, this problem is expressed quite pungently by B. Richmond (who knew better than anybody): *"The second problem (of SD) is the visual complexity of SF-diagrams which often look like a spaghetti of stocks, flows, converters and connectors. We must find ways of parsing them into more bite-sized chunks"*[1]. This problem limits severely the usefulness of SF-diagrams in communicating with users or even between modellers. It also makes our models difficult to modify and to maintain. Using modular and hierarchical diagrams clearly improves greatly the situation but does not change the basic method itself which, to the best of my knowledge and despite the importance of this problem, has been left unchanged since the early days.

During the 2002 meeting of the SD Society, I presented a new SF-language called *"Kinetic Process Graphs or KPGs"* encompassing our current SF-diagrams as special cases but providing in many cases SF-diagrams with much less connectors and more intuitive stock-flow networks [4]. KPGs are not based on the classical tank-pipe analogy of SD, but on a less familiar but more powerful metaphor using generalised chemical reactions. Since the nature of our language shapes the way we express things, new modelling ideas emerge also quite often from its use. The reviewers of the paper were quite positive. Using their own words, they judged the paper as *"providing an innovative and very cohesive representational alternative to SF-diagrams"* and they found it *"challenging and interesting"*. However, during the congress, it was quite clear that the message was not getting through very well. With hindsight, I realised that I was focusing too much on the arcane details of my new language and not enough on a detailed analysis of why I think that the current method is incomplete. Trying to change the metaphor on which a whole community thrives is a difficult and dangerous job. In order to avoid instant rejection, it has to be carefully motivated and well thought off and I was trying to achieve too much in a few pages.

The goal of the current paper is thus to serve as an introduction to the previous one by presenting in more depth why I believe that our current SF-diagramming method is sometimes insufficient and how we may improve it. As explained before our major difficulty is the spaghetti-aspect referred to in the above quotation by Richmond. In some application domains, it is largely an unavoidable consequence of the complexity of the system modelled and, except for introducing modules, hierarchical representations, arrays and similar constructs; it seems that nothing much can be done about it. For instance, let us consider the Business Dynamics SF-Diagram of Figure 1.

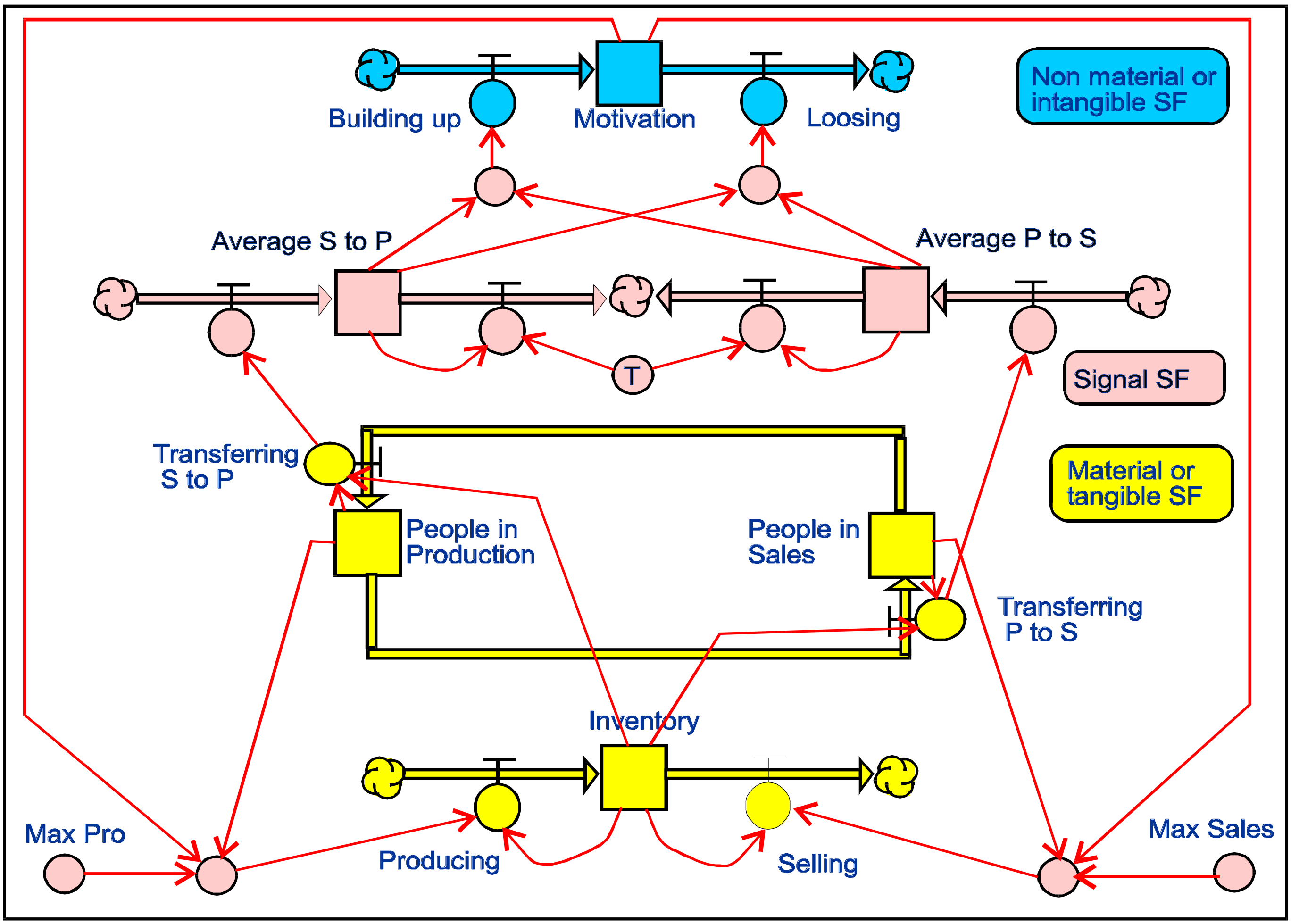

Figure 1: Initial sketch of a simple "Business Dynamics" model. The products in the Inventory stock are produced and sold by the two flows "Producing" and "Selling". The rates of these flows depend respectively on the number of people in the stocks "People in production" and "People in sales". They also depend on a maximum factor and on the motivation of the workers computed in the stock "Motivation". The total work force is constant (no hiring or firing) and divided in two stocks representing the number of people affected to production and those working in sales. People may be transferred from one of these stocks to the other. Each of these two transfer rates depends on the number of people in the stock which is left and on the history of the inventory. There should be an averaging process included here but we neglected it since it was not contributing to our argument. The transfer rates from sales to production and from production to sales are averaged by two averaging processes with a time-constant T and the signals representing the averaged transfer rates control the motivation of the workers who lose motivation when transferred too frequently and builds it up when stable. Although disguised by intermediate nodes, the basic SD oscillatory structure where two stocks serve each as a producing resource for generating the other stock's flow ([3], p.6-8] is recognisable between the stocks "Motivation" and "Inventory". The equations of the various flows and connectors are not given here.

It is not really a fully developed model but just an initial sketch of the main ideas underlying what would become a model after further development. Its structure, explained in detail in the legend, is inspired by the Figure 3.21 of the Stella manual "Introduction to Systems Thinking" [3] modified to introduce simply and explicitly a distinction between some parts of a SF-diagram which we need to distinguish in the rest of this paper:

- The *"Stocks and Flows Network"* contains all the parts which are not connectors or converters, it is divided in three different sub-networks:
    - A *"Material part"* in yellow (people and inventory) manipulating physical or tangible quantities,
    - A *"Non-material part"* in blue (motivation) dealing with intangible, non-physical quantities
    - A *"Signal part"* (averaging) in pink which, although made also of stocks and flows, belongs naturally to the converter network described hereafter to which it gives a dynamics.
- The *"Converter network"* made of converters and connectors of the signal part indicated in pink like the signal stocks and flows and which controls the various flows.

Despite its simplicity, this model is typical of the way many SD models start their life. Its structure is intuitive and only a few explanations are needed. The SF-diagramming language is well suited to its specification. We know that, if we try to make it more realistic, we will soon generate the kind of spaghetti mentioned before but this seems an unavoidable consequence of the use of a graphical language to express complex multi-factorial systems. At this stage and for this kind of applications, a basic reformulation of the SF-language appears thus unwarranted.

In the following, we will have no specific criticism to make against the non material and signal parts of SF-diagrams. However, we will argue later that the adequacy of the SF-language to represent the material parts of SF-diagrams is questionable. We will try to show that the structure shown for that part in many models and in Figure 1 depends on two factors: the first is obviously the nature of the problem under study and the second is the almost unconscious intervention of simplification hypotheses made to adapt the model to the peculiarities of the SF-language used. The first factor is a positive one; it gives its intuitiveness to the diagram. However, we will argue that the second limits and may hinder our thinking. We will then show that the elimination of these simplifying hypotheses necessitates a new SF-diagramming method or language encompassing the current one as a special case and that this new language provides a way to develop more complex models with a better representation for the material parts (more intuitiveness) while also decreasing the complexity of the signal part (much less connectors)

We begin in section 2 by restricting our attention to SF-diagrams representing models in population dynamics, a class of applications which presents an acute form of the problems which we want to address. We will show that the SF-diagrams of these models are extremely spaghetti-like and intuitively awkward. In section 3 we will argue that these problems are due to the fact that, in complete opposition with what most SD practitioners would say, a SF-diagram is not close to our intuition or mental model but to the mathematics of the system described. We will indeed show that, for systems which can be described by differential equations, SF-diagrams are in a one to one correspondence with the block diagram representations used in mathematically-oriented simulation packages like SIMULINK™ of MATLAB™. We will argue that SF-diagrams must therefore be, like Richmond claimed (see before) parsed into more bite-sized chunks, less close to the mathematics and closer to our mental models.

In section 4, we will claim that the root of the lack of intuitiveness and complication of SF-diagrams in population dynamics is that these models are isomorphic to sets of generalised chemical reactions which we call sets of *"kinetic processes"*. We will show that, when applied to a set of kinetic processes, our current SF-diagramming method produces necessarily the spaghetti aspect and the lack of intuitiveness discussed before. Our conclusion will thus be that we need a new SF-diagramming language well adapted to the representation of these sets of kinetic processes. However, if this new method is only useful in chemistry and population dynamics, it will not be very worthwhile since, in SD, these applications are a bit marginal. In section 5 we will thus go beyond our initial restriction to population dynamics and suggest that kinetic processes may be used very fruitfully in many SD applications like for instance in business dynamics which is certainly the mainstream SD application. We will then argue that it is the awkwardness of the representation of kinetic processes in the current SF-diagramming method which leads many people working in Business dynamics to avoid them, unconsciously or not, and leads to the simplification hypotheses mentioned before. We will thus propose to revisit many fields of applications of SD with a method well adapted to kinetic processes to see if introducing them explicitly leads to new insights. Having thus justified fully the use of our new method, we will refer the reader to our previous paper presented at the Palermo congress for a complete presentation of our method to transform a set of kinetic processes in a fully defined computational model called a *"Kinetic Process Graphs (KPG)"*. We claim that our KPGs, which are currently implemented in a commercial package called 20-SIM™ and include the current tank-pipe SF-diagrams of SD as a special case, supersede our current SF-diagramming method and should be used with profit in many applications.

2. The Material Parts of SF-Diagrams in Population Dynamics Are Spaghetti-like and Awkward from an Intuitive Point of View.

The best way to start thinking about the *"spaghetti syndrome"* is to analyse a well chosen example presenting an acute case in order to infer some of its general characteristics. We must therefore choose a spaghetti model which will serve as an appropriate test bed. To be representative, it should belong to a domain studied classically in most SD textbooks. In order to focus on the material part of the SF-diagram, it should have an extensive material part. Finally, as can be inferred from section 1, its structure should be almost imposed by the general knowledge of the field and leave only a very limited freedom for unconscious simplifying hypotheses about the selection of stocks, flows and connectors to include in the model.

The SF-diagram of Figure 2 satisfies all this requirements. Its domain, ecological population dynamics, is studied in many SD textbooks (see for instance [6], [7],[8]) and has no non-material part. Moreover, it is the domain used by one of the main SD-package provider to introduce SD to beginners [9]. It may thus be considered as a paradigmatic example of the application of SD and we would expect it to present examples of good SF-diagrams.

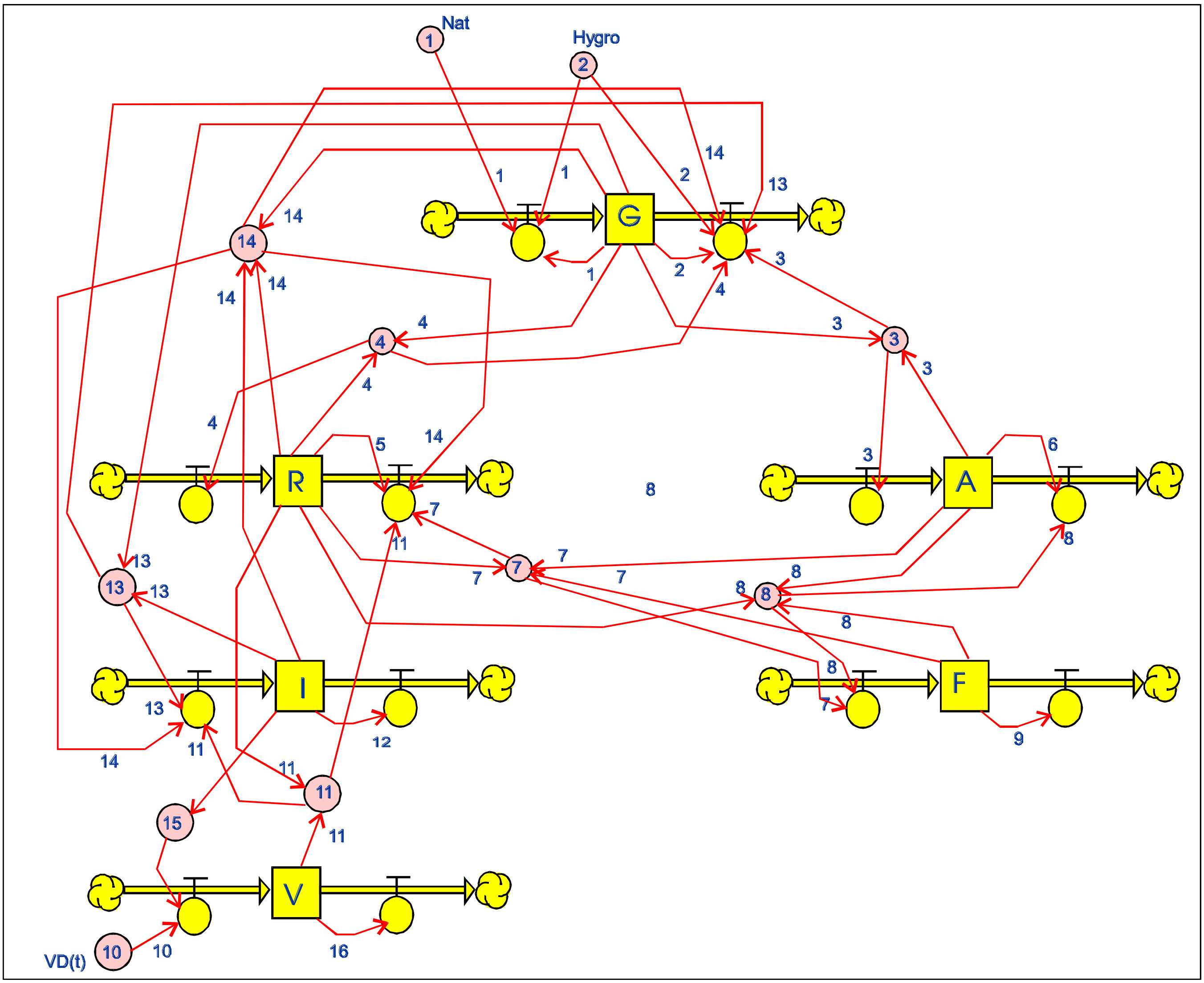

Figure 2: A simplified model of a part of the attempts by Australian authorities to control the explosion in the population of rabbits by man-introduced species (foxes) and viruses (e.g. myxomatosis) (See [10]). The stocks are G (grass), R (rabbits), A (small Australian herbivores), I (infected rabbits), F (foxes), V (virus in a vector population). Each stock has input and output flows controlled by mechanisms represented by the connector part (in red): 1= growth of G, 2= natural decay of G, 3= grazing of G by A and related birth of A, 4 = grazing of G by R and related birth of R, 5 = natural death of R, 6 = natural death of A, 7 = predation of R by F (depending on A) and related death of R and birth of F, 8 =predation of A by F (depending on R) and related death of A and birth of F, 9 = natural death of F; 10 = Human-controlled virus dispersion as a function of time, 11 = Infection of R by V (epidemiological mechanism of appearance of I), 12 = death of I, 13 = reproduction of I with I giving birth to I, 14 = grazing of G by I and subsequent birth of I, 15 = addition to V of viruses from I (vectors biting the I population), 16 = decay of V and death of vectors. To simplify the diagram, many connectors present several outputs (scaled differently).The equations are not presented here but they are classical and impose the "spaghetti" structure of connectors displayed in red. See [10] for further details and development of a more realistic model.

To tell the truth, the SF-diagrams presented in the above references do not really suffer from the spaghetti syndrome. However, all we have done to get Figure 2 is just to go a bit further in the application of the same modelling concepts by including a few more stocks and flows than customary. The result is very obvious; despite its small size (only six stocks), the SF-diagram obtained suffers from a severe spaghetti syndrome. Remark that the functional dependencies of its flows and thus the detailed structure of the diagram do not depend on some choices left to the modeller but are imposed by the classical hypotheses of population dynamics and epidemiology. These hypotheses cannot be changed and force us therefore to confront the problems of SF-diagrams without any possibility to simplify them by ignoring some stocks or flows in order to digest more easily the spaghetti.

It is a simple didactical model used in a study of the attempts by the Australian authorities to control their rabbit population by infecting rabbits with specific insect-borne viruses (e.g. myxomatosis and now calicivirus). If, for the moment, we ignore the SF-diagram, the basic ecological facts underlying the model are easy to explain in words. We consider a fixed territory in which grass (G) growths by using mineral natural resources (Nat) and Water (hygrometry Hygro). Grass is grazed by small Australian mammals (A) and by rabbits (R) imported originally from abroad but proliferating wildly due to the absence of their natural predators in Australia. The population A tends to disappear due to the depletion of the reserve of grass by the rabbits. Initially classical means of control (not modelled) have been used (hunters, fences, destroying burrows) but they were not sufficient. In a second attempt at controlling the rabbit population, foxes (F) have been introduced in the system in an attempt to provide a predator for the rabbits but progressively, foxes have shifted their feeding priority to the small Australian mammals. This is thus a typical “fix that fails” solution since now the system contains two proliferating foreign species (rabbits and foxes) out of control and eliminating the natural species. In another attempt at controlling rabbits, viruses infecting the rabbit population have been introduced. They cause a high death rate and a loss of reproductive abilities in rabbits which never recover from infection. Obviously, this is also somewhat of a “fix that fails” since in a few years, the viruses and the attacked rabbit population have evolved to become respectively less virulent and more resistant. However, these evolutionary aspects are neglected in the model discussed here.

All this is very simple and can be told in a few words. However, if we try to explain this story to non-specialists by using the SF-diagram and pointing at its various flows and connectors, we soon loose our audience which suffers from a form of spaghetti-induced “cognitive overload”. Yet this is only a very simplified model, a fact which illustrates quite vividly our point: when some “not very high” complexity threshold is crossed, FSD diagrams become very difficult to use for model description. As a consequence, modellers do not describe them explicitly. This is certainly a very bad practice since then, model verification and development by other people become very difficult, to say nothing about the use of these models in teaching and educating the general public.

We might perhaps make Fig.1 more palatable by re-organising it a little bit but finding a good way to draw the various connectors is far from trivial and generally such a re-organisation presents diminishing returns for increased system complexity. Remark also that each part of the model is quite simple and that encapsulating several parts in a sector or in a sub-model would just decrease superficially the complication and, by hiding some details, this modification would just create another problem. It is the very way in which SF-diagrams are formulated which creates the problem. In fact, Figure 2 presents an aspect which is characteristic of many SD models:

- The material SF-network (in yellow) is very simple and made of disconnected sub-networks with very simple (source-stock-sink) structures, one sub-network for each species.
- The spaghetti aspect is due to the fact that many of the mechanisms to be taken into account are represented by converters (i.e. 1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15) with several inputs and sending signals to several flows.

The stock-flow structure is thus very (perhaps too…?) simple and the network of connectors and converters network is very (perhaps too…?) complex. These characteristics are found in most population dynamics models but, as we have said, this field, although important, does not belong to the mainstream SD applications. This problem might thus be minor. However, we will suggest later that, although in an indirect form, it affect also some mainstream applications like Business Dynamics. Therefore, we cannot ignore it. Indeed, why do we use a graphical language? Obviously, because a picture is worth a thousand words or equations. SF-diagrams are thus intended to be communication tools, with ourselves, with other modellers and with model users. However, if what we have said is true, many SF-diagrams hinder understanding and communication instead of improving them.

Moreover, the spaghetti syndrome is not the only problem presented by the SFD of Fig.1. I will now try to show that it also suffers from something I call *“semantic awkwardness”*. Obviously, this is somewhat concealed in the maze of Fig.1. To see it more clearly, let us analyse the simpler model of Figure 3. It shows the same kind of structure than in Figure 1: two source-stock-sink structures linked by a network of connectors and converters.

However, this time, the spaghetti syndrome does not strike much and the converter network is sufficiently simple to be presented and understood by everybody. It is in fact a modified version of the classical prey-predator « Pop Dynam » model presented in page 31 of the « Getting Started » booklet of Stella™ [4]. Its simplified version has probably been chosen by HPS as an introductory example due to its intuitiveness, its good compromise between simplicity and complexity and its intrinsic interest. I tried therefore to use the model of Figure 2 as my main communication tool to explain the underlying ecological facts (exponential and limited population growth; dynamic carrying capacity; links between nutrition, metabolism and reproduction; prey-predators interactions) to various newcomers to both SD and ecology chosen to have very different backgrounds and capacity of abstraction (students in secondary education, medical students, engineering students, average citizens) I then asked them to comment on their feelings about the diagrams.

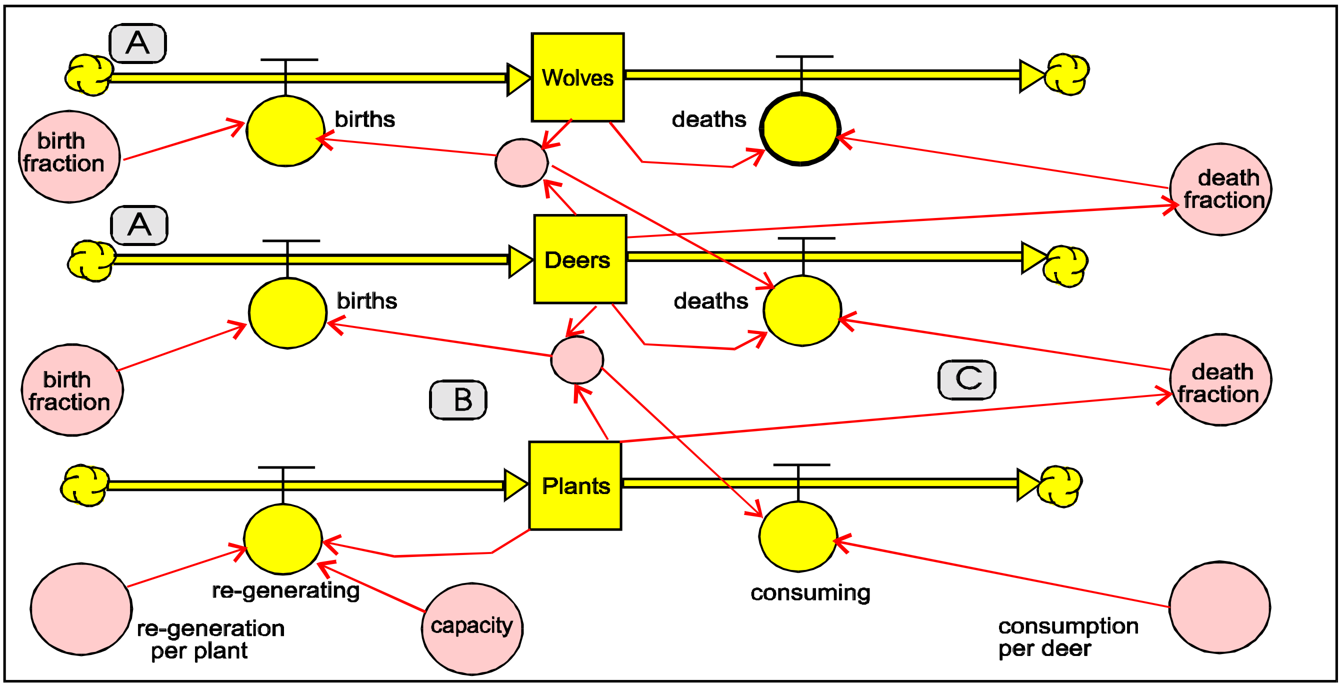

Figure 3: A SF diagram modified from an introductory SD text [9].

The results of this test have been discussed in [4] but, they are so central to our argument that we will briefly restate them here. First let us remark that this SF-diagram is really very simple and indeed, most people found him acceptable but many were nevertheless quite puzzled by it.

Here is a typical sample of their remarks:

- A 13 year old student pointing to source A (Fig.3): « What is the source of deer? Where do they come from? »

- A student in a BSc in biology, showing the signal B (Fig.3): « You told us that red lines are signals and do not transport materials. But then look: “eating plants” is a signal. This is strange. Wolves eat deer. This is not really a signal.

- An engineer and researcher in biology: « Ok, I can use your diagrams, this is no big deal but to me, they are just another way to describe the equations. ?They do not fully capture what I mean when I speak about material flows from plants to deer in metabolism. It is not a signal but an exchange of materials...»

Over the years, I have discussed informally these problems with many people and I have heard this kind of remarks quite frequently. They suggest a couple of general remarks. Firstly, in SD, we are always told to look at endogenous causes for the observed behaviours. It is thus interesting to note that the first comment given above, although somewhat naïve, concerns an exogenous source representing, not a true boundary or environmental condition, but what is in fact an endogenous growth process. This is in fact a “dummy” source. Such sources are endemic in SD models representing growth processes (see for instance the sources of people or capital investment in the World 1 model [11] or chapter 16 and 18 in [12]). Of course, for people used to SF-diagramming, these dummies do not present any problem. We all know that they are just a notational trick but it is indeed this tricky and artificial nature which creates a problem for the students. It seems that they are sources… of confusion to be eliminated when and if possible. Obviously, this cannot be done without basic changes to our SF-diagramming method.

This first problem concerns only endogenous growth. The second remark, triggered by the two other « gut reactions » given above, is more general. These people express what they perceive as non-natural in some SF-diagrams: the representation of some "material" flows (e.g. plants enter into deer) by "immaterial" signals linking material entities of different natures. In the following, I will try to demonstrate that these remarks point at a very basic deficiency of SF-diagramming closely linked with the first one and creating for many people a grey area of confusion which makes them feel uneasy when confronted to a SF-diagram. We will show later that, in order to avoid this confusion, we need to use a SF-diagram with material entities of different natures interacting directly without necessarily exchanging any signals. Of course, this is directly in conflict with a basic tenet of SD which we call the *"principle of material consistency"* (see Forrester in [2], p.50):

> "It should be noted that flow rates transport the content of one level to another. Therefore the levels within one (SF)-network must all have the same kind of content. Inflows and outflows connecting to a level must transport the same kind of items that are stored in that level. Items of one type must not flow into levels that store another type."

This principle seems sound and indisputable. To challenge it, we will thus need a lot of justification and very deep changes in the structure of SF-Diagrams. However, if we take the above comments seriously, then the conclusion cannot be avoided: it is not only the spaghetti syndrome but also a *"lack of intuitiveness"* or a *"semantic awkwardness"* which cuts deep into the core of SD. We have thus to change deeply our SF-diagramming method to make it more parsimonious (resulting in diagrams with much less arrows) and much closer to our intuition.

3. Most of the Complication and Lack of Intuitiveness of the Material SF-Diagrams Used in Population Dynamics Comes from Their Use of Low Level Graphical Primitives Which Are not Close Enough to Our Mental Models.

If it exists really, the *"semantic awkwardness"* just described shows that the current material SF-diagrams do not match the kind of structures which we use or would like to see or recognise consciously or not in our mental models of material flows. This is obviously impossible to prove since, to say the least, the kind of generic structures (if any…?) which we use in our mental models is not very well known. However, since we deal here with material SF-diagrams, we are in an area of modelling quite close to physics and chemistry. As we will see, we can thus learn a lot from the representations used in these fields. We will start with a quick tour of some basic notions in electricity. At first sight, this visit may seem removed from our main line of enquiry. However, we will see later that the notions developed will directly inspire and have close parallels in our new SF-diagramming language.

Let us consider the circuit of Figure 4-A. Together with the equations of its elements (E, R, L, C) and Kirchoff's laws, it gives us an intuitive model of the real system modelled. Indeed, professionals, teachers and students use this representation to discuss informally the behaviour of the real system. Used in conjunction with specialised simulators (e.g. SPICE), the circuit representation is also a fully-fledged computational model. Its suitability for both informal and computational uses scales up well to bigger, more complex circuits. The language of electrical networks satisfies therefore all the criteria to be a good diagramming method for electricity

Figure 4-B shows another representation of the same system, a *"block diagram (BD)"* used by packages like SIMULINK™. The arrows indicate signals travelling between the nodes in the direction pointed at by the arrows. Each node of a block diagram describes a mathematical operation acting on its inputs (arrows received by the node), to produce its outputs (arrows issued from the node). To be able to describe all the equations used in modelling, the BD language uses many kind of nodes but, for the simple system represented here, we will need only five of them: a "sum block" (a rectangle with a symbol Σ) doing the algebraic sum of its signed inputs; a gain (a triangle with a constant indicated inside) multiplying its input by the constant; a function generator (a rectangle containing the symbol of a time function) having no input and generating the indicated function of time; an integrator (a box with an integral sign) computing the definite integral of its input; a connector (a circular dot) receiving a signal and distributing it to several locations.

To draw the BD of an electrical circuit, one starts from the circuit and, using the equations of the components and the Kirchoff laws, one generate a set of differential equations called the state equations expressing the time derivatives of the energy-storing or state variables i.e. the charges in the capacitors (here Q_{C1} and Q_{C2}) and the magnetic fluxes in the inductors (here Φ_L) as functions of some time inputs (here $E(t)$) and of these charges and fluxes. Then from the state equations, one generates the block diagrams by first drawing integrators for the derivatives (here dQ_{C1}/dt, dQ_{C2}/dt, $d\Phi_L/dt$) of the state variables and then using their outputs (the state variables) to generate progressively the three inputs of the integrators (the state derivatives) by realising the right members of the equations with the various blocks.

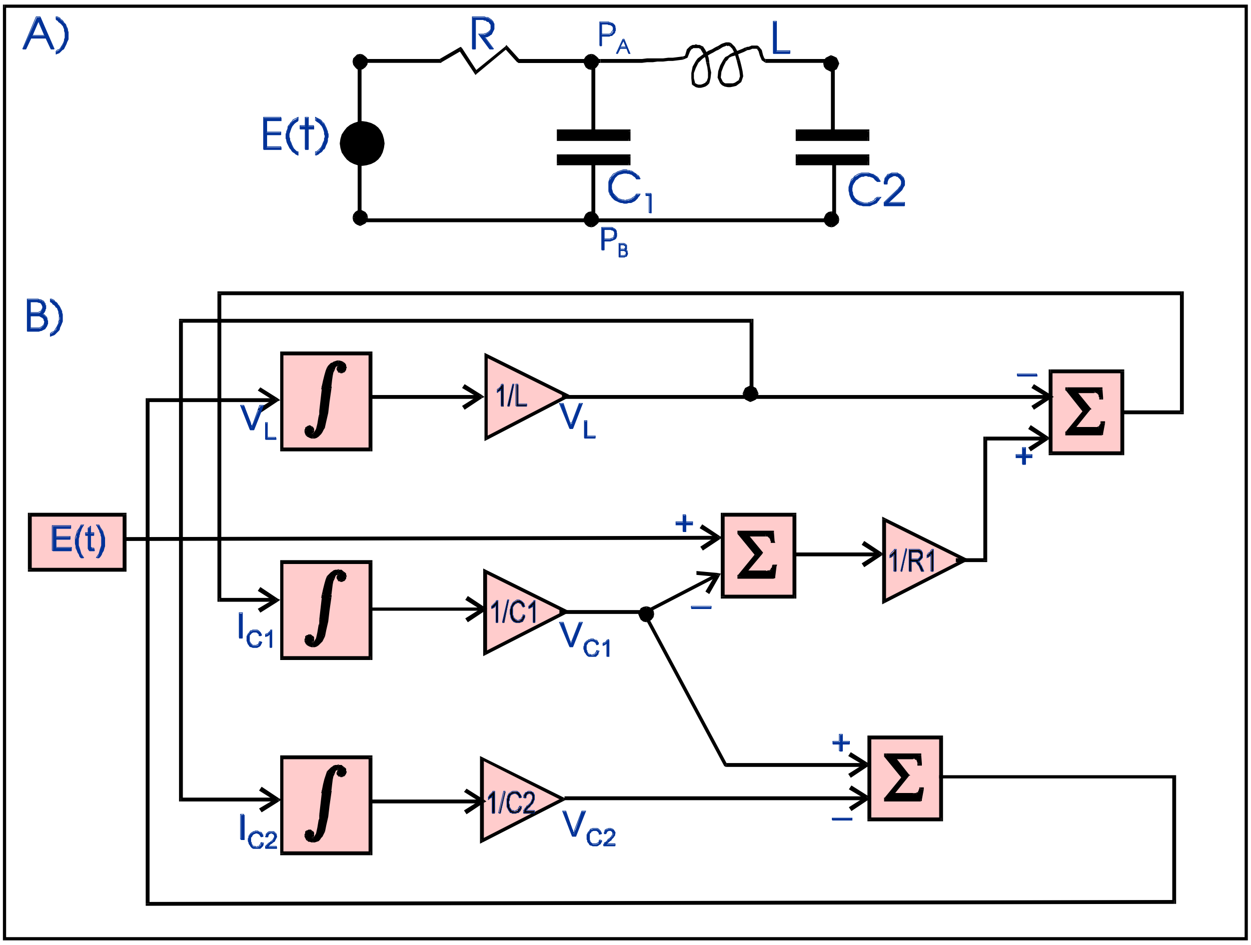

Figure 4: Panel A, a simple electrical network giving us an example of intuitive diagram,
Panel B: a mathematically useful but non intuitive representation of the circuit of panel A.

In our case, the reader will easily verify that the equations are:

$$dQ_{C1}/dt = (E(t) - (Q_{C1}/C_1))/R_1 - \Phi_L/L$$
$$dQ_{C2}/dt = \Phi_L/L$$
$$d\Phi_L/dt \ = Q_{C1}/C_1 - Q_{C2}/C_2$$

They can be read on the block diagram by starting at the three inputs of the integrators and following the arrows in the reverse directions until coming back at E(t) or at an integrator output.

A block diagram is thus a direct graphical representation of the equations of a system. As such, it may be used fruitfully by people who, like for instance control engineers, need to be close to the mathematics of their systems. However, it is a representation unsuitable for any sort of qualitative, intuitive physics-based discussion about a system. Let us remark that, in the form given above, it is impossible to see, without writing the equations, that the circuit and the block diagram describe the same system.

However, the block diagram may be redrawn to show directly by its global shape that it represents the circuit. To see this, let us start by remarking that, according to network theory, each element E, R, L, C_1 and C_2 of the circuit operates on two variables, a voltage V and a current I. This is shown in Figure 5-A for the four components E, R, L, C and for a parallel connection P like those seen at the points P_A and P_B of the circuit. Each of these blocks exchange four signals with the rest of the system (two voltages V_1 and V_2 and two currents I_1 and I_2). These signals may be grouped in two sets called *"ports"*. An element exchanges two signals at each port with the external world. In port 1, these signals are I_1 and V_1 and in port 2, they are I_2 and V_2. Remark that, in each case, the two signals of a port have always different orientations, one toward the block and the other away from it.

In Figure 5-B, we place the elementary blocks just described in the same respective positions than the one they occupy in the circuit and we connect them together according to the equations of the circuit and in a way compatible with the orientations of the signals at the various ports. The block diagram of Figure 5-B is entirely equivalent to the one given previously in Figure 4. However, now, it is very easy to recognize the shape of the circuit. The new block diagram possesses two loops of signals, one for voltages and one for currents. Remark that the orientations of the signals in the various parts of a loop do not necessarily run in the same direction. Indeed, these orientations correspond to the computational requirements imposed by the orientations of the signals in the ports (this is called the "causality" of the ports in block diagram parlance) and not to physical orientations of currents and flow. Remark also that the various boxes are separated by pairs (I, V) of signals running in opposite directions. If we denote such a pair by a single wire; we just redraw the circuit.

Now; the equivalence between the circuit and the block diagram is visually obvious. If we were starting the study of this system by studying its block diagram as given in Figure 4-B, we might even say that Figure 5 shows that the circuit representation of Figure 4-A is just a modular representation of this initial block diagram which separates it in high-level modules representing not the individual mathematical operations but the components having a physical meaning in our mental model of the circuit.

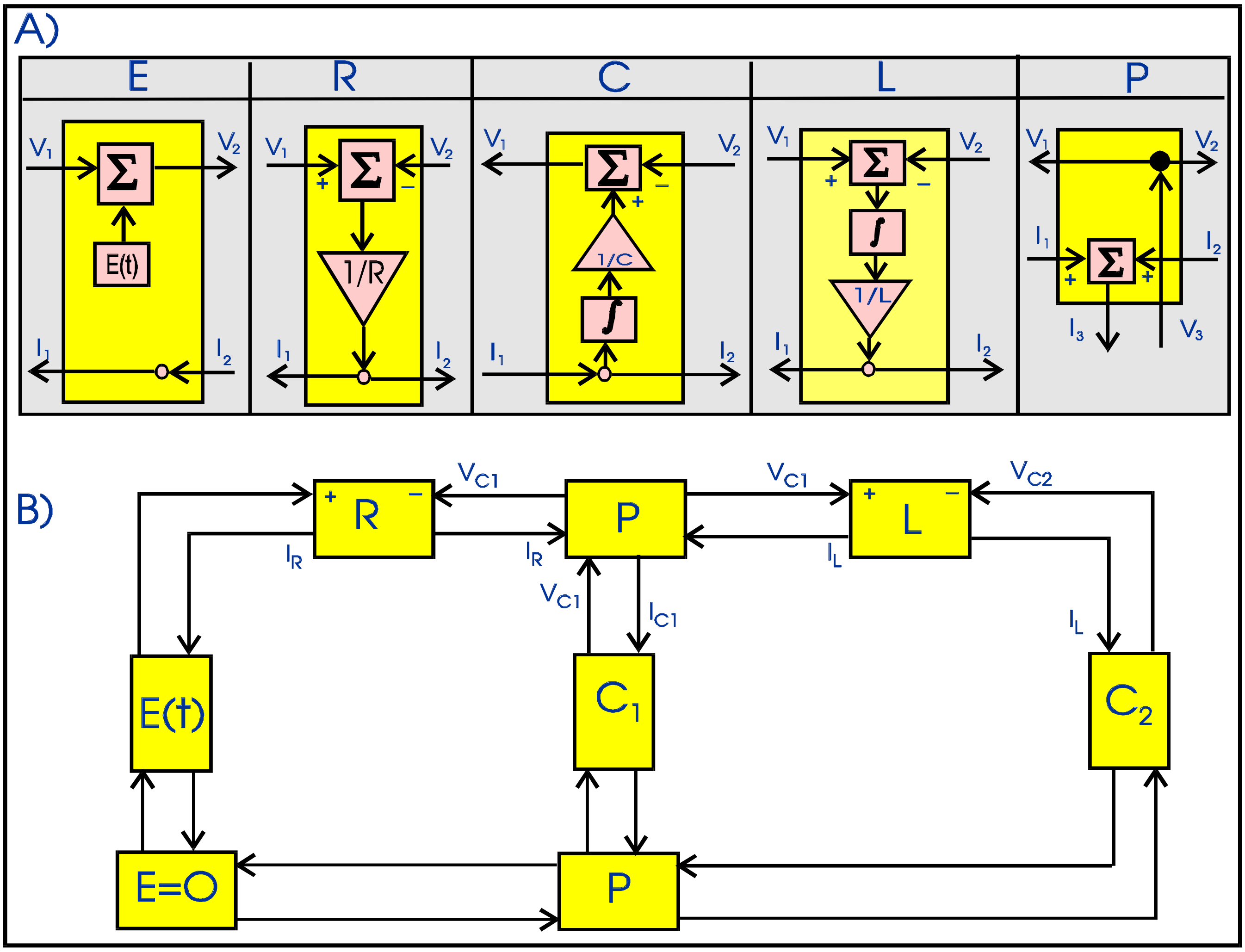

Figure 5: Representation of the circuit of Figure 4-A by a block diagram showing explicitly the circuit topology. Panel A gives the block diagrams equivalent to the four circuit elements E, R, L and C. It gives also the block diagram of a parallel connection P. Panel B uses the element of Panel A to represent the block diagram in a one to one correspondence with the circuit elements.

As we said before, the equivalence between a circuit and its block diagram will have a close parallel in our method. The nodes of our new SF-diagramming language are indeed decomposed in elementary block diagrams very similar to those given above in Figure 5-A and having also ports exchanging two different signals with computational requirements going in opposite directions (see [4] for details). Trying to elucidate the relations between a SF-diagram and the block diagram realizing its mathematical operations will thus be an essential component of our approach.

As a first step in that direction, let us remark that we can give a very simple and generic procedure to obtain the block-diagram representation of a SF-diagram. This is illustrated in Figure 6 for an elementary SF-diagram made of a source, a stock and a sink linked by two flows. To facilitate the comparison with the block diagram, we represent the SF-diagrams very sketchily as a background form and we superpose on it the various elements of the equivalent block diagram.

We start by drawing an integrator for the stock and we connect it to a sum block with a plus and a minus sign. Then starting from the output of the integrator (stock content) and from auxiliary blocks representing constants or input signals, we generate progressively the signals representing the input and output flows of the stock by using the mathematical blocks of the block diagram notation in place of the converters of the SF-notation. We end up by connecting these input and output flow signals respectively to the plus and minus inputs of the sum block.

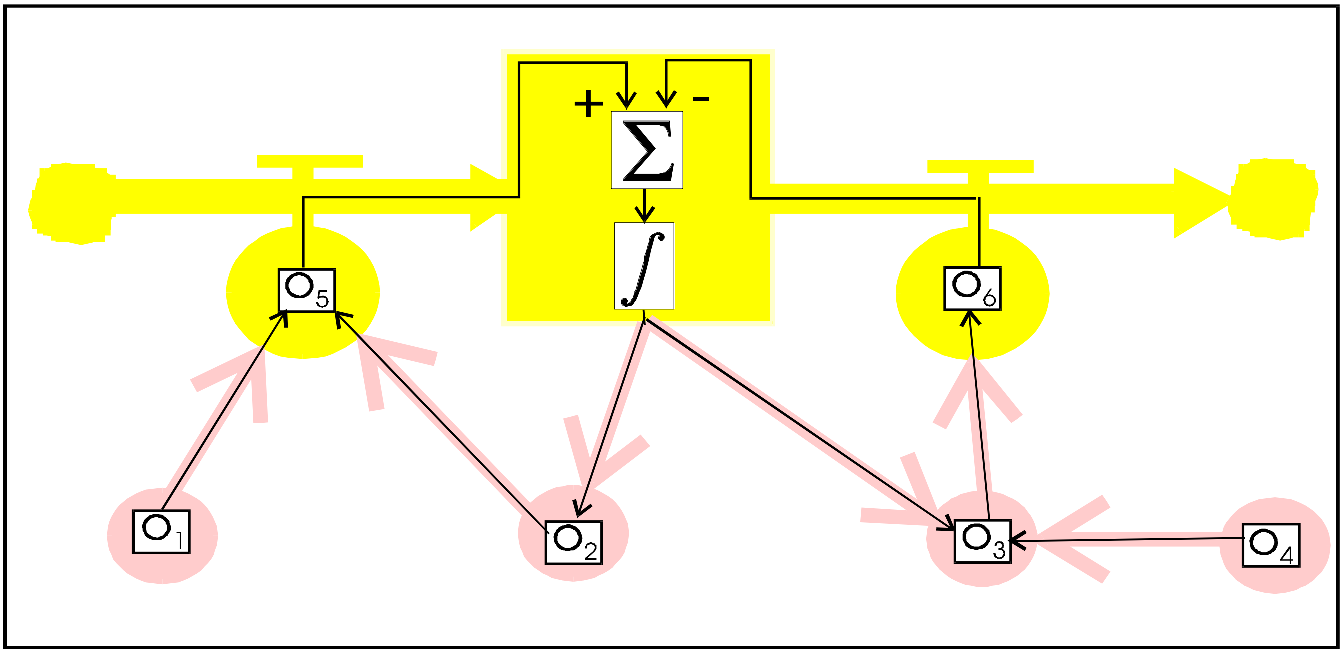

Figure 6: Equivalence between a SF-diagram (represented as a background shape) and a block diagram superposed on the SF-diagram. Each node indicated "o$_i$" denotes a mathematical operation identical to the set of operations executed by the corresponding element of the SF-diagram. There is a one to one correspondence between the elements of the SF-diagram and those of the block diagram.

Using this procedure, we can obtain the BD equivalent of any SF-diagram. Let us note a few points about this procedure. First the only generic part of the SF-diagram is the interconnection of the stock and the flows. Indeed, we find stocks and flows in every SF-diagram and they are always characterised by the same equations, a stock integrating the balance of its input and output flows. The blocks denoted i, Σ, O_1 and O_2 and their specific interconnections exist therefore for each stock in every SF-diagram although the mathematical definition of O_5 and O_6 might differ from case to case. This part of the block diagram corresponds well to the notion of generic, high level component called elementary components in the circuit. It is in fact close to an electrical C component which integrates the sum of its input and output currents to compute its charge

On the other side, although the notion of converter is generic, the converters of Figure 6 and their interconnection have no genericity. They might correspond to different operations, have a different semantics and be connected differently. We observe that there is a one to one correspondence between these converters and the blocks O_1 to O_4 of the block diagram. Thus, except for the stock and the flows, going from the block diagram to the SF-diagram does not involve any grouping of computational blocks in higher level generic elements like it was the case in electricity. In figure 6, except for the stock and flows, we do not constitute groups of blocks with a generic semantics corresponding to some general concept used in many of our mental models.

If the only high level concept involved in the differentiation between a SF-diagram and its equivalent block diagram is the stock and flow concept and if we cannot group the converters in generic components, we can hardly say that the semantic structure of SF-diagrams is rich. The only high level generic concept is the important but admittedly very simple and elementary idea of a tank which may be filled and emptied. This may hardly be compared with the richness of the semantics of the generic concepts E, R, L and P in circuits. Consequently, for material flows, SF-diagrams are not more and not less intuitive than block diagrams. They have just the same content of generic intuition and semantics and this content is very low.

Our conclusion is that, SF-diagrams are not a high level, generic representation (i.e. one close to our mental models but a very low level one (close to mathematics). We are thus left with the problem of finding a higher level representation of material SF-diagrams: we must invent a diagramming method less influenced by the mathematical structure of the model considered and closer to our mental models of material stock-flow systems.

This goal is specified more accurately in Figure 7. The upper part shows the relation which we have discussed above between circuits (class B) and their equivalent block diagrams (class A) in the modular, topologically equivalent form (Figure 5). The block diagrams are low level representations, close to the equations. From them, one obtains the circuits by defining higher level modules (E, R…) putting together some computational blocks in a way which is physically meaningful and generic for every circuit. These modules are the basic building bricks of every circuit and are close to the physics of the situation. A model using them is a high level model close to our intuition.

The lower part of the figure illustrates our goal which is to apply the same concepts in System Dynamics. On the left, we see our current models expressed either as SF-diagrams or as the block diagrams obtained by the procedure of Figure 6. We put them in the same class (C) since we have just demonstrated that they have a node to node equivalence and are thus both at the same low level of generic intuition.

Looking at the blocks A, B and C of Figure 7 suggests an obvious question: can we find, for material stock-flow systems, the equivalent of the circuits in electricity, i.e. a way of creating a class D, a modularisation of the SF-diagrams which groups together the elements of the block diagram of class C in a way which, in addition to the stock and flow concepts brings out generic building bricks closer to the physics of the situation or to our mental models of material SF-diagrams than the symbols of our current method.

To put it differently, we have shown that the stocks and flows of the tank and pipe analogy define a kind of loose analogue to a C element in electricity. Can we find a new SF-diagramming method which defines additional *"high level and generic concepts"* loosely analogue to the other circuit elements? To do that, we will obviously need to go beyond the tank pipe analogy, we will thus need a new metaphor for stock-flow systems and it is towards finding it that we turn now our attention (see the quotation of P. Bak at the beginning of section 1)

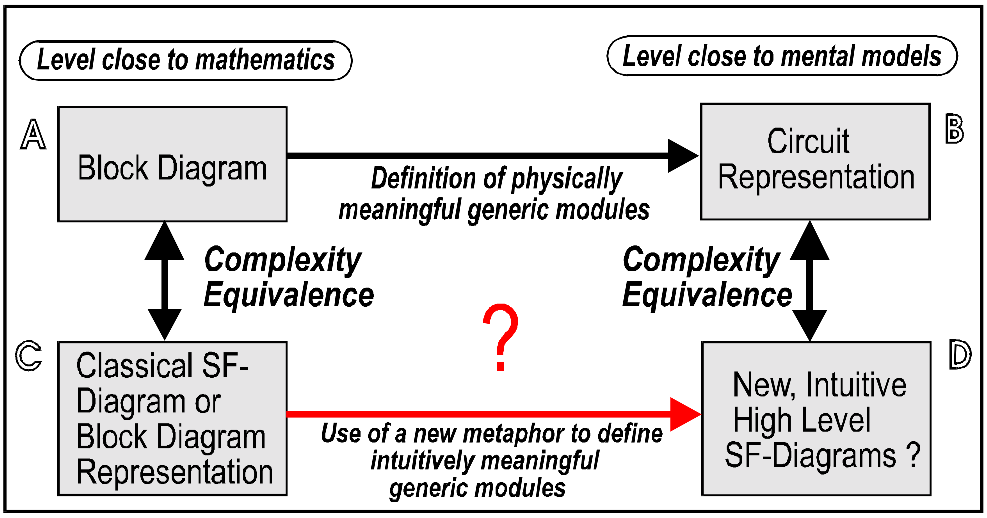
Level close to mathematics
Level close to mental models
A
Block Diagram
Definition of physically meaningful generic modules
Circuit Representation
B
Complexity Equivalence
Complexity Equivalence
?
C
Classical SF-Diagram or Block Diagram Representation
Use of a new metaphor to define intuitively meaningful generic modules
New, Intuitive High Level SF-Diagrams ?
D

Figure 7: A graphical representation of the need to find for System Dynamics a set of high level (i.e. intuitively rich) generic components giving a representation of material stock-flow systems which is for System Dynamics what Circuits are for electricity. This involves the need to define loose analogues of the classical E, R, L, C components and junction structures (series, parallel) in electricity.

Changing the tank-pipe metaphor will obviously entail more than giving a face lift to SF-diagrams, some more heavy surgery will be needed. Currently, the primitive concepts (in the sense of language theory) of these diagrams are only a few: stocks, flows, taps, converters and connectors, sources and sinks. With these 7 classes, you can build every SF-diagram. This simplicity is very attractive since the elements of SF-diagramming may be learned in a few hours. However, it might also be a weakness. If you use a simple language, it becomes very intricate to express subtle thoughts and our discussion implies that the current SF language, despite its well proven value, suffers just from that problem. We will thus need to keep its seven primitive concepts but perhaps modify their semantics and almost certainly add a few more building blocks of our own.

4. The Root of the Problems of SF-Diagrams in Population Dynamics is That They Are Equivalent to Chemical Reactions for Which SF-Diagrams Are Unsuitable; Metabolic Maps, a Representation Used in Biochemistry Might Be an Useful Way Forward.

In the preceding sections, we have presented a series of arguments supporting our view that the current SF-diagramming method is not well adapted to population dynamics. In this section, we will show that the root of the difficulty is that population dynamics systems are closely related to kinetic models in biochemistry and that our current SF-diagramming method is inadequate to represent the kinetics of systems of chemical reactions. We will also see that, to deal with the complexity of their systems of reactions, biochemists use in their informal discussions, a kind of back of the envelope diagramming method called *"Metabolic Maps"* which, after some formalization and generalisation might give us a good starting point in our search for a new SF method.

Let us start by recalling some notions of chemistry. The basic object of study in chemical kinetics is the unidirectional reaction which in general may be described by a chemical equation:

$$N_1A_1 + N_2A_2 + \ldots + N_NA_N \rightarrow M_1B_1 + M_2B_2 + \ldots + M_MB_M$$

Expressing that N_1 units of A_1, N_2 units of A_2,… and N_N units of A_N react together to give M_1 units of B_1, M_2 units of B_2 … and M_M units of B_M. The A_n are called reactants and the B_m products. The numbers N_n and M_m are called the stoechiometric coefficients. In order for a reaction to take place, reactants have first to collide but a simultaneous collision of more than two molecules is highly unlikely. Thus true or "elementary" reactions can have only two reactants. They are the following:

1) $A \rightarrow B$ which is called *"unimolecular"* since it has only one reactant and one product;
2) $A+B \rightarrow C$, $A \rightarrow B+C$ and $A+B \rightarrow C+D$ which are called *"bimolecular"* since at least one of their sides (reactants or products) contains two species;
3) $A+B \rightarrow C+D+E+F+\ldots$ which *is "pluri-molecular"* (several products) but has only two reactants.

Consider now a more complex equation like $3A+2B \rightarrow 2C$. It cannot be elementary since an individual reaction step would require the simultaneous collision of 3 *As* and 2 *Bs*. We call it a combined reaction and it expresses the global balance of several elementary reactions like for instance $A+A \rightarrow D$, $D+A \rightarrow D1$; $D1+B \rightarrow D2$ $D2+B \rightarrow 2C$ where D, D_1 and D_2 are intermediates which, being produced in a reaction and consumed in another, do not appear in the balance. Chemists and most notably, biochemists need very often to study very large sets of reactions made of many simultaneous elementary or combined reactions. They are almost always interested in the amounts of intermediates produced and thus cannot be satisfied with a global balance. A simple example is the Michaelis - Menten mechanism which is used as a paradigm for all the enzymatic reactions. It is given by the following set of two reactions:

$$S+E \rightarrow C;\ C \rightarrow P+E$$

in which a reactant *S* (the substrate) combines first with an enzyme *E* to produce a complex *C* which then reacts to form a product *P* and to regenerate the enzyme *E*. The balance of the two reactions is $S \rightarrow P$ and does not show the enzyme or the complex explicitly. However, these two species cannot be ignored since *E* accelerates the reaction by a factor which may be as great as 10^6. The set of the two equations needs thus to be taken into account.

Another more complex example is the following system: $S_1+E \rightarrow C_1$; $C_1+S_1 \rightarrow C_2$; $C_2 \rightarrow P+E$; $P+E \rightarrow X$ in which an enzyme *E* links first to a first molecule of substrate S_1 to form a complex C1 which then links to a second molecule of the same substrate S_1 to form a second complex C_2 which dissociates to produce a product *P* and regenerate the enzyme *E*. When *P* is in large amount, it may also combine with *E* to form a modified form of the enzyme called *X* which is inactive. The balance is simply $2\ S1 \rightarrow X$ which does not represent at all what happens in the mechanism, i.e. the production of *P*. A deep understanding of the behaviour of a chemical system needs thus the specification and study of its detailed set of reactions.

To be able to compute the time histories of the amount of the various species present in a system of reactions, one must specify not only the chemical equations of its reaction set but also its kinetics, i.e. a set of algebraic equations giving the instantaneous rate of each reaction as a function of the amounts of the various chemicals considered. In simple reactions, the rate of a given reaction is a function of its reactants only and, in the simplest cases, this function is given by the *"Law of Mass Action"* which states that the rate of a unidirectional reaction is a function of the product of the amounts of its various reactants. For instance, the mass action law of the elementary reaction $A+B \rightarrow C+D$ is simply $r= kAB$ where k is the speed of reaction.

However, in more complex cases like for instance in chemical equations expressing balances, a rate equation may become quite complex since it averages in fact the rates of several elementary reaction steps. The global rate may then be a function of all the various chemicals present in the system as well as of other given time functions like for instance the temperature. In general therefore, the rate of a reaction, also called the kinetics of the reaction, is a function:

$$r = R(A, \ldots, A_N, B_1, \ldots, B_m, C_1, \ldots, C_K, y_1, \ldots y_P)$$

with A_n the reactants, B_m the products, C_k some other chemicals present in the system but not intervening directly in the reaction as products or reactants and y_p given time variables. In general, the functions R giving the kinetics of a reaction set cannot be deduced from first principles and are obtained empirically.

The last point to specify in order to be able to compute the time derivatives of the amounts of chemicals is how to deal with the stoechiometry. For instance, for a combined reaction $2A+3B \rightarrow 4C$ with a rate defined by $r=R(A,B)$, one has:

$$dA/dt = -2R;\ dB/dt = -3R;\ dC/dt = +4R$$

Indeed, each time the reaction occurs, two *As* and three *Bs* are combined to form four *Cs* and since the rate of occurrence of these events is *R*. One may write the same kind of differential equations for any reaction with a given arte equation and a known stoechiometry.

We are now ready to see how we can use kinetics idea in our population dynamics models. Indeed, in the models of Figures 2 and 3, we suppose that all the individuals of a given species are equivalent. Moreover their interactions within their own species (reproduction, natural death) and with other species (grazing, predation, infection…) occur with a probability dependent, not of the individuals considered, but only on the numbers of individuals in the various species. Finally, no other interaction than those considered is supposed to occur and no other species is present. A bit of thought shows then that each of these models may be described by an equivalent set of *"generalized reactions*" which we call *"kinetic processes or processes".* For instance, the interactions between the species considered in the simple "Resources, Plants, Deer, Wolves" model of Figure 3 is represented by the following processes:

$$\begin{array}{lll} \text{R1:} & R + P \rightarrow n_P P & ;\ r_1 = R_1(R,P),\ R=Constant \\ \text{R2:} & P + D \rightarrow n_D D & ;\ r_2 = R_2(D,P) \\ \text{R3:} & D + W \rightarrow n_W W & ;\ r_3 = R_3(D,W) \\ \text{R4:} & W \rightarrow X & ;\ r_4 = R_4(W) \end{array}$$

With P = plants, D= deer, R= natural resources (constant), W = wolves and X = dead wolves.

Each process arrow describes flows from the stocks on its left called the *"donor side"* to the stocks on its right called *"the acceptor side".* Like in a chemical reaction, these flows are determined by the kinetics of the process (here *r1,…, r4*) and the stoechiometry. In the above example, all the stoechiometric coefficients on the donor side are equal to one. The stoechiometric coefficients on the acceptor side are n_P, n_D, n_W (all three bigger than one) and *1*. The kinetics $r_1, \ldots r_4$ express the hypotheses on the functional dependencies which were made to draw the converters and connectors of Figure 3. The algebraic form of these functions is not specified here.

The equations of a process are simple balances. For instance, for the process R1 occurring at a rate r_1, we have

$$dR/dt = -r_1 \text{ and } dP/dt = -r_1 + n_P\, r1$$

Indeed, *R* disappears at a rate r_1 and *P* enters into the process (in some sense, it disappears into it) at a rate r_1 to re-appear on the other side at a rate $n_P\, r_1$. Similar equations may be written for the other processes and the reader will check easily that the whole model is given by:

$$\begin{array}{lll} dR/dt & = - & r_1(R,P) \\ dP/dt & = & (n_p - 1)\, r_1(R,P) - r_2(D,P) \\ dD/dt & = & (n_D - 1)\, r_2(D,P) - r_3(D,W) \\ dW/dt & = & (n_W - 1)\, r_3(D,W) - r_4(W) \end{array}$$

In words, these processes may be read as follows:

R1: Plants *P* and natural resources *R* interact in a process of plant growth which consumes resources and, per unit of plants entering and thus counted negatively, produces n_p units of plants (n_P>1)

R2: Deer and plants interact in a process of grazing and reproduction which consumes plants and, per unit of deer entering and counted negatively, produces n_D nits of deer ($n_D > 1$)

R3: Wolves and deer interact in a process of predation and reproduction which consumes deer and, per unit of deer entering and counted negatively, produce n_W units of wolves ($n_W > 1$)

R4: Wolves enter into a process of natural death which kills them

As a side comment, let us remark that representing this model by processes makes its naivety quite glaring…, however, it is not the process representation which is naïve but the original model and we might easily add other processes making it more realistic. The process view just makes the naivety more apparent. This is certainly a quality of this way of presenting a model and not a drawback. Let us insist that the above process view is entirely equivalent to the SF-diagram explained before and represented in Figure 3 which was, in a way, hiding the naivety.

For a simple model like the one described here, the description in processes appears thus at first sight as a credible alternative to the classical SF-diagram and we might use it as a tool for diagramming. However, we will see that it becomes very confusing when the number of processes increases and when these processes are strongly coupled, i.e. when several stocks appear on the donor side in some processes and on the acceptor side on other. We will therefore propose later another representation called a *"kinetic process graph (KPG)"* directly inspired by the notion of process described above but which scales up well for large sets of coupled processes. As an exercise to convince him/her that a set of processes may become confusing, the reader may try to make the above model more realistic by adding for instance other species (hares, foxes), intermediate age classes (babies and young rabbits and wolves) and sexes (males and females). He/she may also wish to write down the list of processes describing the Australian ecology model of Figure 3… and, why not…, make it more realistic.

From the above example, it is easy to infer that every population dynamics model may be reformulated as a set of processes. We need thus to see how well SF-diagrams fare in expressing processes or more simply sets of chemical reactions. Let us first consider the elementary unimolecular reaction A→B. In simple cases, its rate (number of molecules of A transformed per unit time) is simply given by $R=kA$ (law of mass action) but in more complex cases, it may be given by more complicated formulas as a function of the amount of A and B present and eventually of other time-dependent quantities $Y_1,\ldots,Y_n$ like PH or Temperature or amount of other substances present in the reaction vessel. Since the stoechiometry coefficients are both equal to 1, the rate and species equations are:

$$R = r(A,B,Y_1,\ldots, Y_n)\ ;\ dA/dt= -r,\ dB/dt = r$$

Figure 8-A shows that this equation describes also what happens in a simple SF-diagram with two stocks (A and B) and a single flow defined by the formula of r given above. The stocks represent stores of the chemicals A and B and the flow is the reaction process transforming A into B at the rate given by the chosen kinetics *r(A,B,Y)*. The familiar tank-pipe analogy of System Dynamics and unimolecular reactions are thus completely equivalent. To illustrate this point, the panel B of Fig.8 gives the stocks and flows of the following system of 7 reactions:

$$A\rightarrow B;\quad A\rightarrow D;\quad D\rightarrow B;\quad B\rightarrow C;\quad C\rightarrow A,\quad E\rightarrow F;\quad F\rightarrow E$$

The network of converters and connectors has not been represented. The system has two inputs of chemicals *A* and *E* from outside (sources) and an output of *B* into the environment (sink). It is made of two disconnected components. The first is *(A, B, C, D)* and the second is *(E, F)*. They are linked only by connectors which are not represented but must exist to compute the various rates. This system of stocks and flows shows a structure familiar to every SD practitioner: A SF-diagram which, from the point of view of its stocks and flows is made of several very simple sub-networks. Each sub-network has stocks and flows disconnected from the stocks and flows of the other networks. Two disconnected sub-networks may only be connected by converters and connectors.

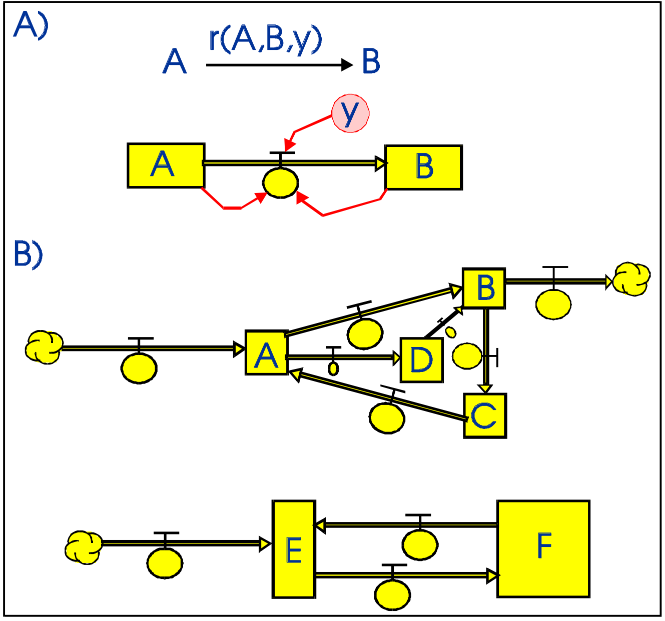

Figure 8: A) Equivalence between the tank-pipe analogy and unimolecular reactions. B) The stocks and flows of a system of unimolecular reactions.

At this stage, we have gained nothing by going from tank and pipes to sets of reactions. Except for the change of chemical nature between the donor and acceptor sides of a reaction, systems of unimolecular reactions and tank-pipe systems are completely equivalent and SF-diagrams are perfectly suitable for representing both. However, here comes the catch: *we have seen that the vast majority of chemical reactions are not unimolecular but pluri-molecular*. Therefore, if we want to apply System Dynamics to chemistry, we will not go very far with our unimolecular reactions. What happens when we consider more interesting reactions? Let us for instance study the combined but still simple reaction *nA + mB* → *pC + qD*. During each elementary reaction event (what we call "a step"), this reaction puts together *m* units of *A* and *n* units of *B* and re-arranges their atoms to produce *p* units of *C* and *q* units of *D* (*n, m, p, q* being the stoechiometry coefficients). As we will now see, our SF-diagramming method does not fare well in representing such a reaction.

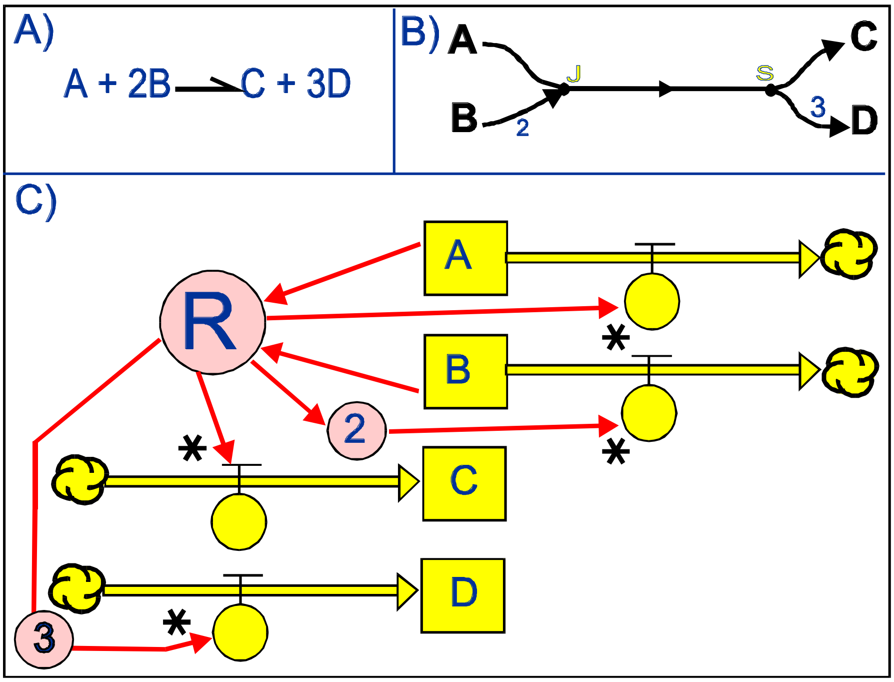

Fig.9: Insufficiency of SF-diagrams for pluri-molecular reactions: the origin of spaghetti and semantic awkwardness.

Figure 9 illustrates this point for the reaction *A + 2 B* → *C + 3 D*. Chemists represent it by the chemical formula used before and reproduced in panel 9-A. However, in biochemistry, they have to consider sets of simultaneous reactions with up to several tenths of reactions and they get lost in a long list of equations. In that case, they prefer to use the informal representation which, as we will see below, has many important advantages and which, suitably formalized and generalized will form the basis of our new language. They call it a *"metabolic map"*.

The basic element of a metabolic map is the representation of a reaction illustrated in Figure 9-B for the case of Figure 9-A. We can see four stocks of *A, B, C* and *D* joined by a multi-input, multi-output arrow having arbitrary numbers of inputs and outputs (in hypergraph theory, it is called a branch). Here our branch is connected on the input or donor side to the two stores *(A, B*) and on the output or donor side to the two stores (*C, D).* The flows (number of molecules consumed per unit time) of *A* and *B* get together at the point *J* (junction or join). Each time the reaction is occurring, this point accepts one *A* and two *Bs* from the stores of A and B. The molecules joined by J react together in a reaction which is represented by the part of the arrow between J and S. This reaction transforms the nature of the products between its input and its output. In J we have C A and B and in S, they are replaced by C and D. The flows of products *C* and *D* separates at point *S* to go into the stocks *C* and *D*. The coefficients *n=1, m=2, p=1 and q=3* indicates that per *A* used, the reaction uses 2 *Bs* for a single step producing 1 *C* and 3 *Ds.*

Now let us see what happens if we use a SF-diagram to represent this reaction. If it was unimolecular, everything would be perfect. As shown previously, the chemical equation A→B and the SF-diagram are equivalent. It is easy to see that the metabolic map is also given by A→B. The three representations are thus equivalent. However, for our pluri-molecular reaction, this equivalence disappears. The map of Figure 9-B puts together in J two flows of different natures, a flow of *A* and a flow of *B* and, in System Dynamics, this is forbidden by what we previously called the principle of material consistency. Similarly, the flow of products separates at point *S* in two flows of different nature (*C* and *D*). This is also forbidden by material consistency. Thus, to give a SF-diagram of this reaction, *we need to use a trick to represent by information links what happens at J and S, i.e. a transformation between flows of different natures.* This is exactly what our users reacted to with astonishment before: the representation of something which happens between flows as a signal. The result is the SF-diagram of panel C.

It has four stocks for *A, B, C* and *D*. Since *A* and *B* are only used in the reaction, the corresponding stocks are linked to sinks. Similarly, since *C* and *D* are only produced in the reaction, they are linked to sources. The reaction rate is computed in the block (R) which expresses the kinetics chosen *r = R(A,B)* and the value *r* of this rate is sent to the taps linked to the sources of *A* and *B* and to the sinks of *C* and *D*.

This SF-diagram implements the correct kinetic and stoechiometric equations. It is thus computationally correct. However, it is not difficult to see that it is unsatisfactory from the point of view of its intuitive content. Indeed, despite its simplicity, it shows already the emergence of the two symptoms of *"spaghetti-like complication"* and *"semantic awkwardness"* discussed previously:

> A spaghetti of non-local connectors (going from a stock-flow sub-network to another): they impose the constraints of matter conservation between the time variations of A, B, C, D (the four arrows denoted by an asterisk) and start to give to the diagram its messy look. For the single reaction represented here, this is still acceptable. However, if we model a network putting together many reactions, the problem is amplified and the result soon becomes an intractable mess.
>
> A semantically awkward stock and flow part: We are modelling a closed system: as it can be seen in its map, our reaction has no link to the environment. The four clouds introduced in the SFD are thus artefacts needed to create the necessary increases and decreases of the stocks. These clouds have no counterpart in the real system. To a seasoned SD modeller this is not a problem. He or she knows that this is only a notational device (although some confusion may creep in when the system becomes big and also includes true sources and sinks). However, as shown again by the reactions of the people we interviewed, this kind of trick creates a lot of semantic trouble for beginners. Haven't we told them to try looking at endogenous factors? Here we introduce completely artificial exogenous factors. This is hardly a textbook example of good practice but there is no way to avoid it if we have to use the SF language.

Let us try to summarise all this with a smile: as a result of these two drawbacks, we will certainly not convince a biochemist if we try to convince him or her that panel C is a simple and intuitive representation of the reaction for which he or she usually uses the notations of panel A or B. However, our example is still too simple to really make our point. To strengthen the argument, let us now compare a SF-diagram and a metabolic map for a slightly more complex set of reactions called the Michaelis-Menten enzymatic mechanism. To understand what this system does, let us look first at its map (Figure 10-A). A chemical called the substrate S is fed to the reaction vessel (input I) from the environment represented like in SD by a cloud Env$_1$. In a first reaction R$_1$, the chemical S associates with an enzyme E present in the system to form what biochemists call a complex C (i.e. E and S linked together). A second reaction R$_2$ works on C. It regenerates the enzyme E unchanged and re-arranges the atoms of S to produce a product P. E is thus re-cycled as shown by the cycling arrow called "Enzyme cycle", i.e. sent back to its stock E and made available to react again with another S. The product P is sent into the environment of this reaction (cloud Env$_2$, arrow O = output) to be used in other mechanisms not discussed here. At each cycle of a molecule of enzyme, a molecule of S is changed into a molecule of P and E is regenerated.

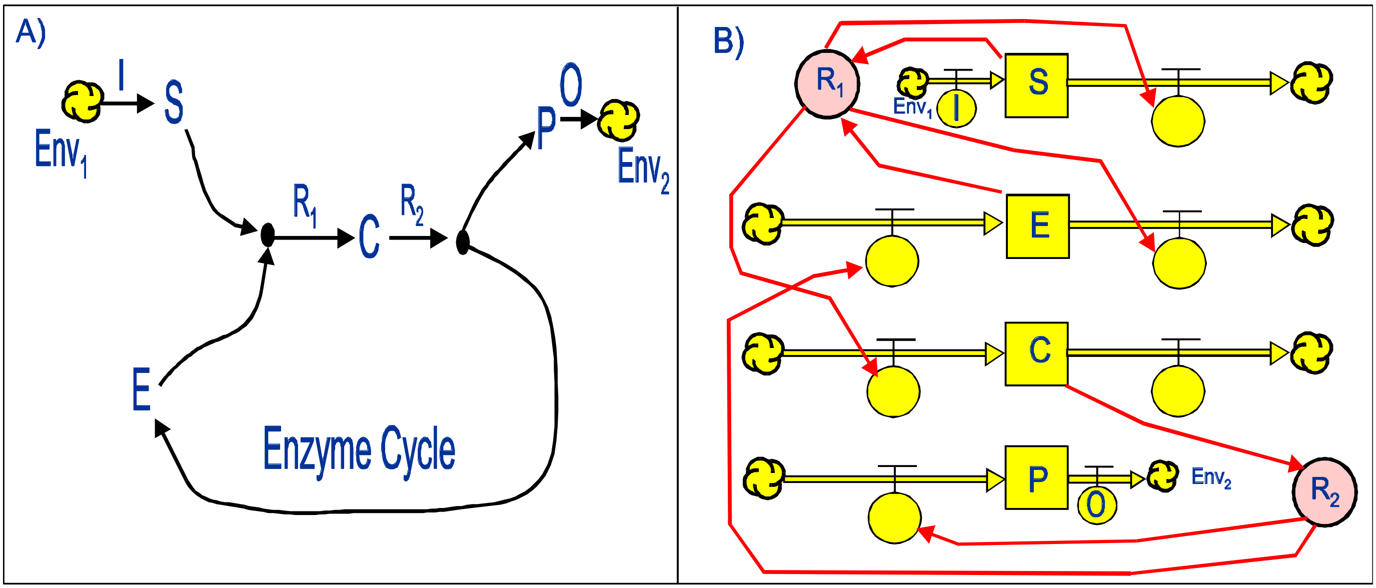

Fig.10: comparison of metabolic maps (A) and SF-diagrams (B) on a simple biochemical system (a Michaelis-Menten enzyme).

This short description of the Michaelis-Menten system shows that metabolic maps are very useful to explain a chemical system to non specialists. Remark that the various pathways do not exist physically in the reaction vessel which is a homogeneous solution; they just give a mental picture of reality. The fact that this picture is used by all biochemists and understood easily by everybody shows that it is a useful mental model. Remark also that the map has only the minimal number of arrows needed, that its two clouds are physically meaningful (they are true input and output) and that the paths followed by each chemical species are clearly indicated including the recycling of E.

In strong contrast, the panel B of Fig.10 gives the corresponding SF-diagram modified from [7]. Clearly the spaghetti syndrome and the semantic awkwardness begin to strike again. Using this SF-diagram to explain the system to beginners in biochemistry would be much less enlightening than using the map. Remark also that this is still a very simple system since it has only two reactions. If, like it is often the case in biochemistry, we consider a system with perhaps as much as fifty reactions, using the SF-diagram for communicating with beginners or even domain specialists becomes downright impossible. In contrast, the daily experience of biochemistry teachers shows that the map, although becoming obviously more complex, remains clear and useful. Its parsimony and semantic value scale up gracefully since the map remains understandable and enlightening instead of seeing, like in lists of process equations or SF-diagrams, its usefulness destroyed abruptly when the complexity of the system increases a little bit. This is indeed why biochemists invented these maps and use them everyday.

Obviously a map is not a fully computational model of a reaction set but just an informal “back of the envelope” diagram of the virtual paths followed by the various material transformation flows. Its property of *“graceful scaling up of parsimony and semantic value”* is just what we need in our SF-diagramming method. This point is so important that, in order to really convince the reader, we will now look at a few more complex examples.

Our first example is the following reaction set R_1: $A+B \rightarrow D$; R_2: $B+C \rightarrow E$; R_3: $D+E \rightarrow C+F$. With its three reactions, it may hardly be considered as complex. However the fact that the products *B, C* and *E* appear in more than one reaction has interesting functional consequences which do not pop directly to the eye when we are describing the system by this linear list of equations. The map given in Figure 11-A is much more useful? The catalytic cycling of *C* is indeed obvious and the two arrows going out from *B* to the two reactions R_1 and *R2* show clearly the competition since the product *B* used in one reaction cannot be used by the other. If we had not three reactions but fifty with many competitions and cycles, the list would be useless and the map would become our only way of representing the system.

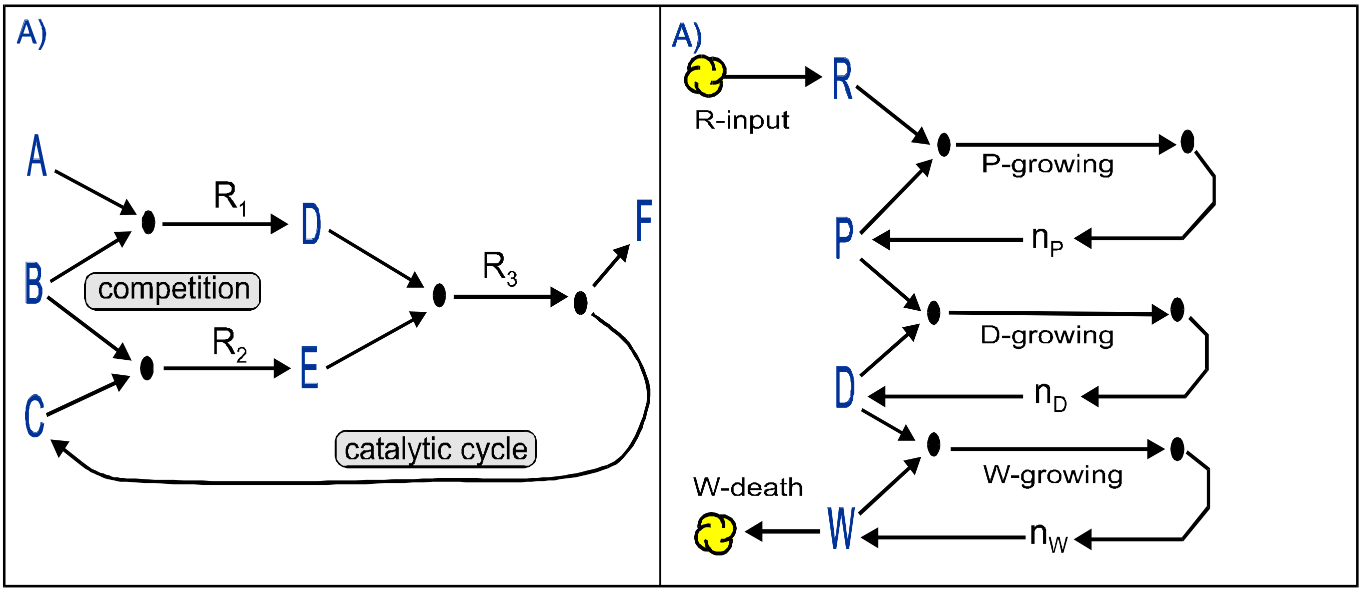

Figure 11: Panel A: A simple reaction set showing a competition and a cycle; Panel B: the map of the (resource, plants, deer, wolves) system of Figure 3.

Our second example is the system (resource, plant, deer, wolves) for which we have previously given the SF-diagram and the linear list of processes. We can translate this list almost trivially into the map of Figure11-B. We see that the natural resources are entering from the source in the top left corner. They get together with the plants in the plant growing process which per unit of plant entering produces n_P units of plants ($n_P > 1$). Two other growth processes are represented similarly: deer feed on plants to reproduce and growth; wolves feed on deer to give birth to more wolves. Finally the wolves may die from natural death and disappear in a sink in the bottom left corner.

When compared with the linear list of processes given previously, this representation is a bit more intuitive since it shows clearly the facts that each species participates in its own production forming thus a cycle and that the various species are in a food chain (the three cycles are chained vertically). However, the number of processes is still small and with some practice, it is possible to see the same points easily from the list of processes. We will have to show a bigger example to make our point. However, when compared with the SF-diagram of Figure 3, the advantages are impressive: the source and the sink do exist really in nature, they are not dummies. No dummy sources are needed. The number of arrows is minimal and each arrow shows a flow of species which corresponds to something existing really in our mental model. Finally, the explanation of the system follows naturally from its graph. Admittedly, the map is just an informal representation and not a complete computational model like the SF-diagram but clearly, we are well on our way toward better diagrams. All we need to do, at least for chemicals and population dynamics system is to transform the maps in complete computational models while preserving their topological qualities.

Before to do that, let us give a last example which is sufficiently complex to be incontrovertible. It is the Australian ecology problem discussed before. The reader has been invited before to transform its SF-map (Figure 2) in a linear list of processes. In Figure 12, we give the map corresponding to this list.

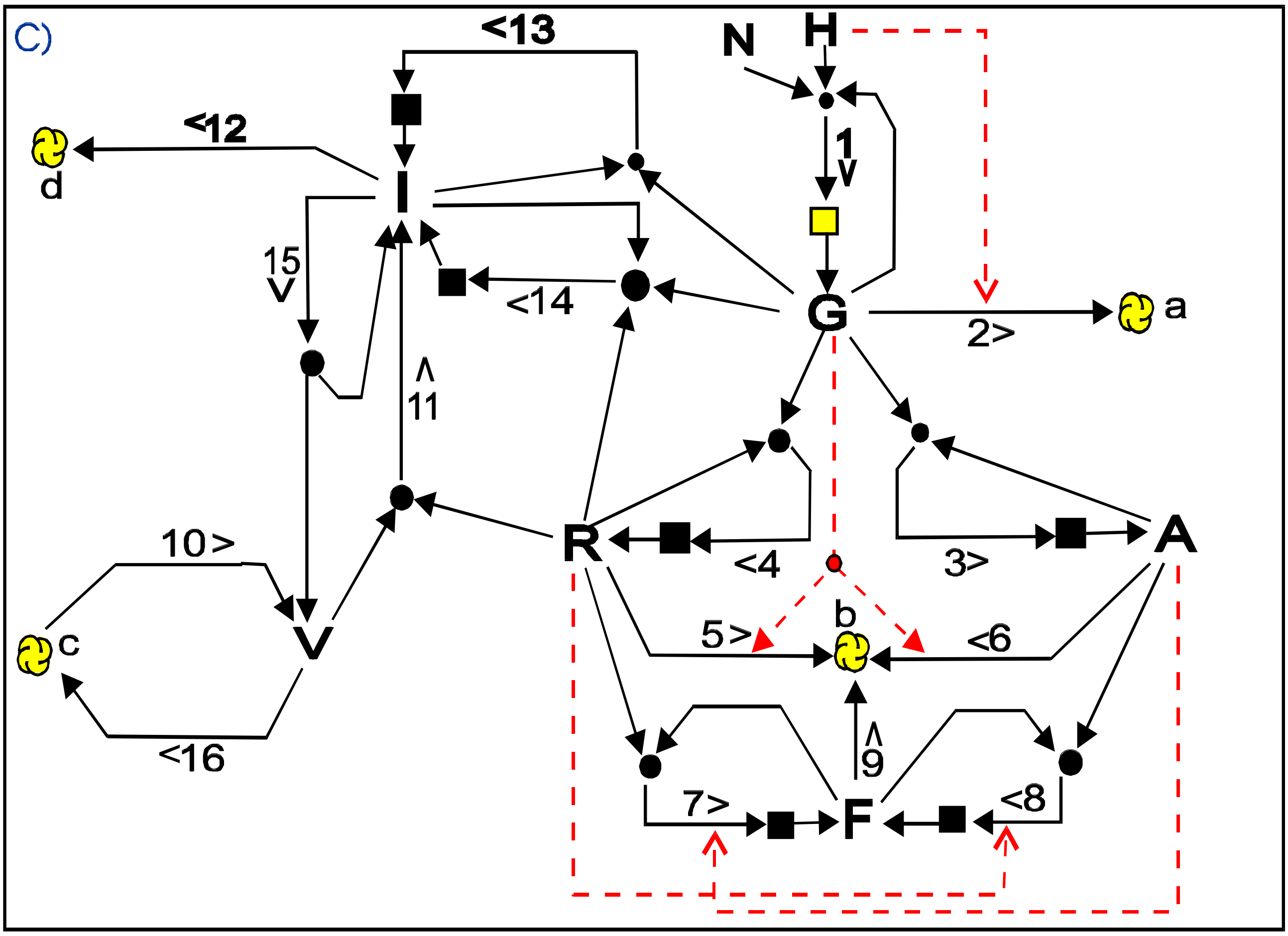

Fig.12: Map of the 16 processes necessary to model the Australian ecology problem of Figure 3.
G: grass, N: natural resources, H: hygrometry, R: rabbits, A: Australian herbivores, F: foxes, I: infected rabbits, V: free virus in vectors

1	Grass growth	6	Natural austr. mam. deaths	11	Infection of rabbits	16	Decay virus
2	Natural grass death (G,H)	7	Foxes predation on R	12	Death Infected		
3	Australian Mammals growth	8	Foxes predation on A	13	Reproduction between I		
4	Rabbits growth	9	Natural foxes death	14	Reproduction I, R		
5	Natural rabbits death	10	Introduction Virus	15	Re-injection virus from I		

This map uses slightly modified notations. The processes are indicated by numbers corresponding to the list given in the legend. Arrow symbols > give the process direction. The “get together” and “separation” junction points are indicated by black circles. Material flows are denoted by continuous arrows while interrupted arrows show signals from a species to a process. Finally, the stoechiometry, representing growth like in the preceding figure, is indicated by black squares inserted in the flows.
Starting from top right, G uses N and H to grow (1) in a process similar to those described previously. Grass may also decay and die (2) if too dry (signal from H). In the first stage of the food chain, the herbivores A and R compete for G (3 and 4) and grow. Both A and R may die naturally (5 and 6) if not enough G is available (signal from G to 5 and 6). In the second stage of the food chain, both A and R die from being predated upon by F (7 and 8). Each of the predations depend on the herbivore non predated upon in the process (signals on 7 from A and on 8 from R) in order to allow a prey shift when one prey becomes too low. Foxes may die their natural deaths (9). Then comes the epidemiological part. V is injected from outside in vector insects in (10) and V infects R producing infected rabbits I which die at a high rate (12), reproduce between themselves (13) or contaminate non infected rabbits (14). Of course in these processes, all animals involved consume G. Finally some V is re-injected in the free vector-borne form V from I by insects biting infected rabbits (15) and some virus V may decay (16).

At first sight, this map appears a bit daunting. However, in contrast with the SF-diagram of Fig.3 in which various connectors transmit the same information, each intervention of a species in a process is indicated by one and only one arrow. Our map has in fact all the qualities which we claimed to be missing and needed in SF-diagrams: no dummy sources, minimal number of arrows, no representation of a material flow by a signal, use of signals only to indicate influence of a species on a process without material transformation, strong intuitiveness due to a clear display of the various cycles, competitions, food chains and reproduction modes. The complexity of Figure 12 is not a consequence of the diagramming method but of the fact that we have 16 processes dependent each on several variables. Even when it is parsimonious and intuitive, the model of a complex system must be complex otherwise something is wrongly oversimplified. In fact, explaining this map is the best way we have found to go beyond the simple explanations given in page 5 in order to tell our students what happens really in the Australian model.

As far as chemistry and population dynamics are concerned, a comparison of Figures 2 and 12 leads us to think that all we still have to do is to formalize and generalise a little bit the informal metabolic maps which have just been discussed in order to transform them into fully detailed computational models while preserving their qualities: clear topology, parsimony and intuitiveness. This is what we will do in section 5 to obtain an elementary form of what we have called *"kinetic process graphs"*.

5. By Formalising and Generalizing the Maps of Section 4, We Obtain "Formal Process Maps" Which Are as Parsimonious and Intuitive as Their Informal Counterparts.

We have presented in our previous paper the transformation of our informal maps in fully fledged computational models. This was done in three steps. Firstly (step 1), we defined unambiguous notations for each part of the map. Secondly (step 2), we specified the flow-related semantics of all the elements of this notation i.e. the way in which each element operates on the flows of entities which go through it. We gave also an algorithm allowing us to compute at all times t the values of all the flows and stock contents of a formalized map if the rates of the various processes are known at all these times. Finally (step 3), we extended our notations to allow the computation of these rates at all times from the values of the contents of the stores and from auxiliary inputs. Taken together, the steps 2 and 3 gave us the complete definition of our "Kinetic Process Graphs", at least in their elementary form. We refer to our previous publication for complete details [4]; here we will only recap the first step, the definition of all the symbols needed to go from the informal maps drawn until here to standardized maps.

First let us define our symbols precisely (Figure 13-A):

- Like in System Dynamics, a *"cloud"* ☁ indicates either a source or a sink of material.
- A *"stock"* is denoted by square brackets surrounding the name of the material stored (e.g. [S]).
- A *"flow link"* is a semi arrow ⇀ which denotes a flow of material from its origin to its end.
- A *"junction node"* is a circle ○ , it is either a J or a S junction, indicating the joining together or the separation of several flows of different natures. The letters J or S may be skipped if the role of the junction is clear. Remark that the nature and thus the physical units of the flows are different in each arrow connected to a J or S node which therefore creates a change (putting them together or getting them separated) in the materials it transmits.
- A *"kinetic process"* or *"generalized reaction"* is a node denoted P> or R>, it represents the process of material transformation (reaction) itself and transforms thus the materials it receives in its acceptor side into those it sends to the acceptor side.
- A *"modifier"* is denoted by (n) and represents a stoechiometric coefficient as defined in the theory of chemical kinetics.

In the next page, we give a few examples of formalized flow maps using all these nodes. As an introduction to the notation, Figure 13 gives a simple map illustrating the use of all the symbols. It describes three coupled processes:

P0: SO→A; P1: 2A + B +C →3D + E + 2C and *P2: E→ S*

In Figure 14 we reformulate some of our previous examples: panel A shows the Michaelis-Menten mechanism of Figure 10, Panel B the competition and autocatalytic cycling mechanism of Figure 11-A and panel C the Australian Ecology problem of Figure 12. Since they have the same topology, these maps keep obviously all the advantages of their informal versions. These maps appear thus very useful and all the more so since I will now show that their application is not restricted to chemistry and population dynamics but may encompass fruitfully every field of application of System Dynamics.

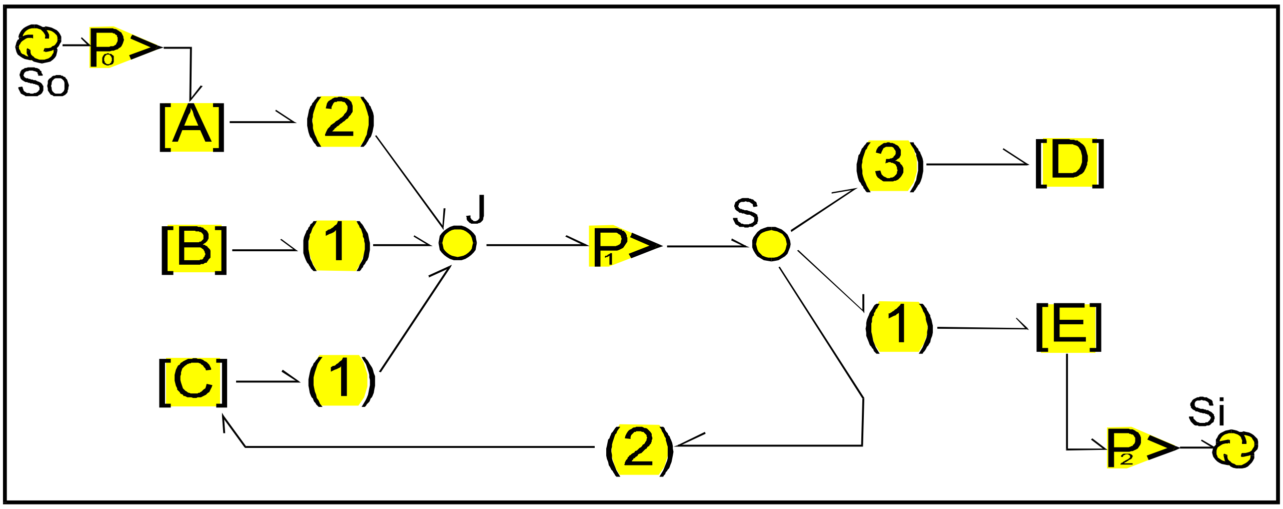

Figure 13: An example of Formal Flow Map showing the use of all the symbols defined until now.

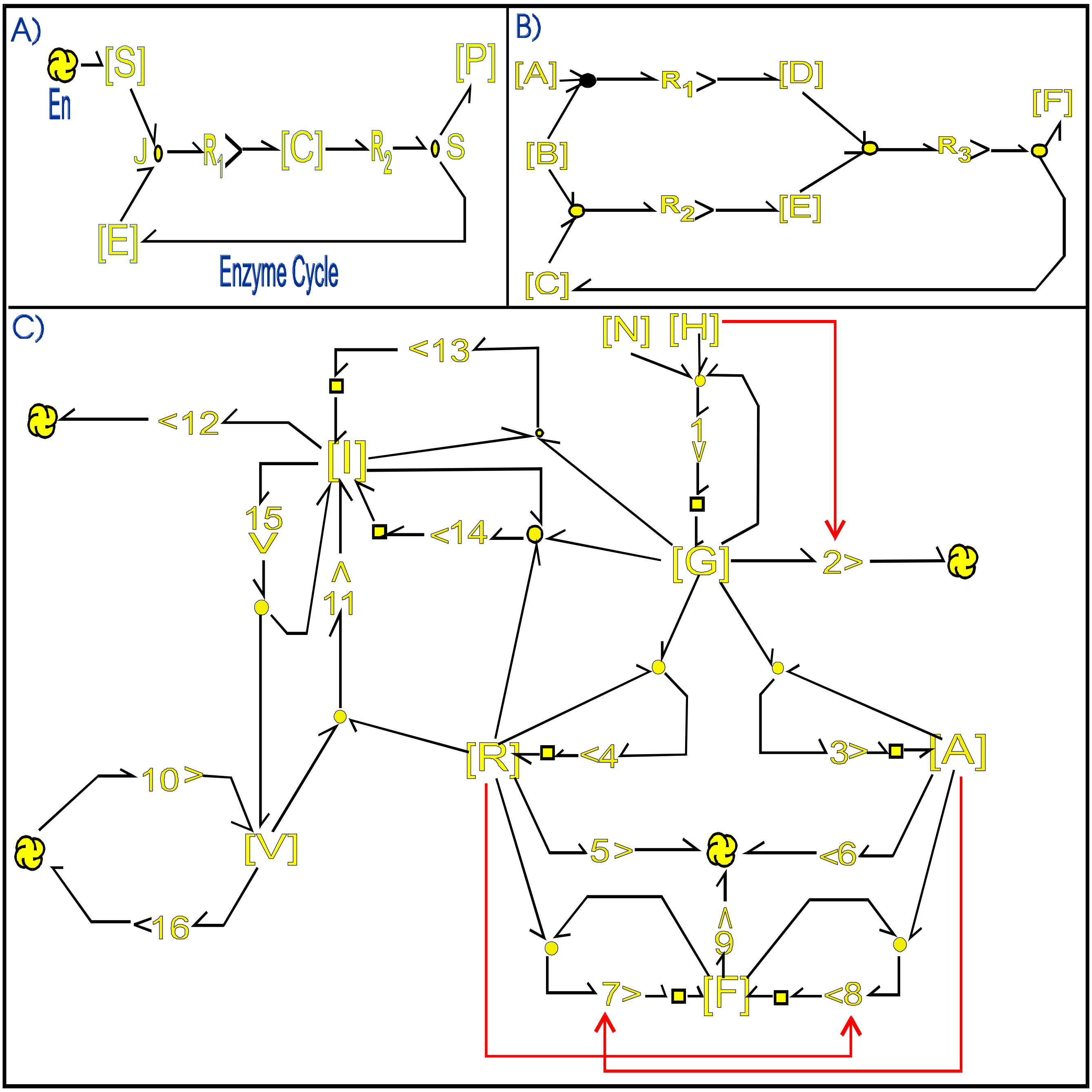

Figure 14: The formal maps of three of our previous examples. Panel A: the Michaelis - Menten Mechanism of Figure 10, Panel B: the competition mechanism of Figure 11-A and Panel C: the Australian ecology problem of Figure 12.

6. “Kinetic Processes” May Be Fruitfully Used Not Only in Biochemistry and Population Dynamics but Also in Other Applications of System Dynamics.

As we have seen above, once you know the list of the processes which you want to consider in a model, you may draw its map. However, until now, we have only used these processes to describe biochemical and population dynamics systems. Now, we will suggest that the same description may be used for many systems in most of the fields currently encountered in System Dynamics. First, le us specify the notion of process more precisely. In general a process will have N inputs $(A_1, .., A_N)$ and transform them in M outputs $(B_1,...,B_M)$ at a rate r (called its kinetics) and with a stoechiometry given by:

$$n_1 A_1 + n_2 A_2 + ... \qquad +n_N A_N \xrightarrow{\quad} \mathrm{r} \xrightarrow{\quad} m_1 B_1 + m_2 B_2 + ...+ m_M B_M$$

To emphasize the fact that it has N inputs and M outputs, we may call it a (N,M)-process. In the elementary (N,M)-processes which we consider in this paper, the rate is a function r = R($\underline{A}$, $\underline{B}$, $\underline{X}$,$\underline{U}$) of four vectors $\underline{A}$, $\underline{B}$, $\underline{X}$, $\underline{U}$ denoting respectively the donors $\underline{A} = (A_1, \ldots , A_N)$, the acceptors $\underline{B} = (B_1, \ldots , B_M)$, a vector of contents of other stocks participating to other processes in the model $\underline{X} =(X_1, \ldots ,X_K)$ and a vector of inputs which are known time functions $\underline{U} = (U_1,\ldots,U_P)$

In more sophisticated processes, the rate may be a Volterra functional of these same variables including thus a memory of the past. In general and in contrast with chemistry, the stoechiometric coefficients may also be functions of $\underline{A}$, $\underline{B}$, $\underline{X}$, $\underline{U}$, i.e.:

$$n_i = n_i(\underline{A}, \underline{B}, \underline{X}, \underline{U}) \quad \text{and} \quad m_i = M_i(\underline{A}, \underline{B}, \underline{X}, \underline{U})$$

We have thus to see if people may consider this notion of (N,M)-process as natural and intuitive, despite its abstract character. In other words, we need to know if, after a brief introduction to this notion, most people will be able to make a list of the processes relevant to a system in which they want to study the flows of *“generalized materials”* (i.e. people, tools, money, animals,…) and draw the map coupling all these processes together. Remark that, despite their importance in complete applications, we do not consider here intangible flows. We have thus to focus on applications in which material flows are important.

To investigate this point, I have again resorted to interviews. I have chosen a few typical application domains: the two already discussed (biochemistry and ecological population dynamics) and a few new ones (regional sustainable development, business dynamics, epidemiology). I have then selected a list of typical applications:

- The biochemical systems discussed previously to which I added cell calcium dynamics and glycolysis.
- A problem in bacteriology: chemostat growth with infection of bacteria by phages, both populations being able to evolve in a genetic arms race.
- The Australian ecology made more realistic by adding other species and evolution of species as well as of viruses (decrease of lethality in viruses and development of resistance in rabbits).
- The socio-ecological-economical development of a natural community emphasizing sustainable development and intending to test its feasibility.
- A model of human resource management modified from the CCRS model published in the System Dynamics literature by Andersen and Sturis [13].
- A model of supply chain dynamics modified from the classical Beer model of System Dynamics.
- A study of just in time production at the strategic level (effects of problems in staff motivation and delays in supply)
- A study of the effects on the employment market of governmental policies favouring continuous education in a country.

For each system, I interviewed a few specialists and students. An interview starts with a brief explanation of the notion of material stocks and flows, (N-M) processes and maps. Two or three typical examples are introduced (Lotka–Volterra system, Michaelis-Menten, logistic growth of a population). Then I ask the interviewee to think about the flows of generalized materials in his/her pet system and to work progressively toward a list of the stocks and processes they want to consider. I also ask them to describe each stock and each process by a short sentence. At several stages, I ask them to draw a progressively more complete map of their system and to comment on its usefulness.

Almost invariably, the interviewees adopted quite easily the process view. They were very frequently able to make a list of processes relevant to their problem and, with some help to transform it in a formal map. When asked to explain these maps, they were also able to give very understandable explanations. In fact, they used the maps to teach me about these systems. Hereafter I will discuss in detail a few examples of processes having emerged during that work. I will focus on examples showing more complex processes than those given before. For each example, I will give the sentence heard during the interview, a description in words of the process implicitly suggested by this sentence and its abstract representation:

A civil servant speaking about the Australian Ecology problem:

- Informal sentence: "In our country, foxes do not use rabbits as a major food source anymore, they have started to feed on small marsupials and they just thrive on them.

- Rephrasing as a process: "foxes, rabbits and small marsupials interact in a predation process which produces more foxes. More and more we observe that foxes shift from rabbits to foxes to find their catch.

- (N,M)-Process:

$$n_R\,(R,\,M)R + n_M\,(R\,,\,M)M + F \rightarrow n_F F + B;$$
$$n_R + n_M = 1;$$
R = rabbits, M = marsupials, F = foxes, B = biomass

note: the stoechiometric coefficients n_R and n_M express that the relative participation of R and M to the feeding process is variable and depend on the availability and ease of access of each prey. They are thus both functions of R and M, inferior to 1 and with a sum equal to 1. The coefficient n_F expresses the idea that each fox entering the process produces more than one fox at the output (itself + its children).

- An executive from a food company:

- Informal sentence: "We tried hard to automate in order to decrease the dependency of our production rate on our human resources but then quality went down since the remaining people were demotivated".

- Rephrasing as a process: "Human resources, equipment, and raw products interact in a food production process. Changing the ratio of automation is OK up to a point."

- (N,M)-process:

$$n_H\,HR + n_{NAE}\,NAE + n_{AE}\,AE + RM \rightarrow n_H\,HR + n_{NAE}\,NAE + n_{AE}\,AE + n_{HQP}\,HQP + n_{LQP}\,LQP$$

with HR = Human resources, NAE = non automated equipment, AE = automated equipment, HQP = high quality products, LQP = low quality products.

note: This process has more than two inputs and more than two outputs. The fact that HR, NAE and AE appear on both sides with equal stoechiometry means that they are not consumed in the process. At first sight, we might ignore them. However, putting them explicitly allows us to compute for instance their total flows or, in a more advanced model, to distinguish between their two states (before and after the process) by introducing different fates for each of them. Most stoechiometric coefficients may be functions of some variables (not indicated in the equation). For instance, n_H may decrease if we invest to increase automation. This also modifies the ratio of NAE to AE used at each step and thus their stoechiometry. Finally, if we add a firing process decreasing HR, we may compute a motivation signal (intangible and thus ignored in this paper) like in Figure 1. This signal may then affect the quality of the products by modifying n_{HQP} and n_{LQP}. The effect of automation on these stoechiometric coefficients should show a hump, it should increase quality up to a point. This example shows that the meaning of a kinetic process may become quite complex and express many subtle effects in a simple notation complemented with a few formulas.

- The CEO of a growing company (see example of Figure 8 in reference 4):

- Informal sentence: "Once we get the development of our Canadian subsidiary out of the way, we will be able to re-invest more in the training of our new sales force. This will also free some of our experts and they will lead our training effort;"

- rephrasing as a set of processes: Several processes are interacting here. In the first, money and human resources (expert) interact to develop a new plant in Canada. In the second, as the degree of completion of this new plant progresses, resources are liberated and the experts become available for training new sales people.

- (N,M)-Processes:

$$\$ + HE + P \rightarrow HE + P \text{ and } \$ + HE + NS \rightarrow TS + HE$$

With $\$$ = Money, HE = human experts, P = degree of completion of new plant (between 0 and 1), NS = new sales force, TS = trained sales force

note: HE and $\$$ are used in two processes competing for them, plant completion and training of sales people. As the first advances (P tending toward 1), its kinetics may be such that its rate decreases and then frees these resources for the second process.

We might continue giving a long list of similar examples. Indeed, during that work, I have encountered many (N,M)-processes in various fields. Here is a small list of examples in each of the domains listed above:

- Metabolism of glycolysis: phosphofructokinase + fructose 6-P + ATP → Fructose 2-P + ADP
- Bacteriology: bacteria+ glucose → more bacteria + useful metabolites + toxins
- Microbial ecosystems: amoeba + bacteria → more amoeba
- Bacteriology: bacteria + phages →genetically variant phages
- Ecology: see examples in previous sections
- Epidemiology: uninfected people + infected people → more infected people
- Management: equipment + materials + resources +people → products + equipment + people + pollution
- Management: people freshly hired + trainees + resources → more trained people
- Supply chain: products +orders + resources → more resources + satisfied customers.
- JIT: WIP stage N + kanbans stage N+1 + resources + materials → WIP stage N+1 + kanbans stage N
- Public health: doctors + patients + resources → outpatients + doctors

Admittedly, the mental models expressed in these examples are often too simple to be really useful. However, they should just be considered as starting points for a more complete mapping of the processes involved. Their meanings may be understood intuitively and, in each case, they may be modified, made more realistic, de-aggregated in several components and coupled to other processes. We have demonstrated in our Palermo paper that the result are usually far less naïve (see figure 8 in [4]) and that the maps obtained represent quite well our mental models.

The ease with which most people get familiar with the (N,M)-process description and the writing of process maps makes us believe that these structures correspond well to the way we frame our mental models of this kind of systems. It seems *that (N,M)-processes and process maps, with their dense network of material exchange, loops and flow junctions connect very naturally with our mental models and enrich them considerably.* Obviously, the PM of a complex system is complex. But in contrast with the SD flow map of the same system, it keeps together the flows which seem to be linked in our cognition. In these cases, our maps, despite their global complexity, retain a high level of local simplicity and intuitiveness and may still be used as good communication tools.

7. Conclusion: By Adding to Formal Process Maps the Information Transforming Them Into a Computational Model, We Obtain “Kinetic Process Graphs”, a New SF-Diagramming Method Superseding the Current One and Including It as a Special Case.

In the preceding sections, we have argued that our kinetic (N,M)-processes give natural representations of the way we perceive the global kinetic result of the dynamic interactions between large numbers of objects encountered in the real systems modelled by system Dynamics. We have also shown that putting a set of (N,M)-processes together in a formal kinetic map provide a very intuitive and parsimonious way of expressing our mental models of the material flows in such systems. Since a map is just a graphical representation, we still need to transform it into a KPG, i.e. into a fully defined computational model? This has already been done in our previous paper. The goal of the present paper was only to justify the need for a new language and to show that (N,M)-processes and kinetic process maps may be its basis. We feel that we have now reached this goal. We refer thus the reader to [4] for all the details on how to transform these process maps in KPGs.

Let us just recall here that this transformation is done in two steps.

In a first step of *“determination of the rates influence functions”,* we attach a computational function to each node of the map. This function computes the effect of the node on what we call *“influence factors”.* These factors allow us to compute the kinetics of each process. To do that, they transport to each process the information which it needs to compute its rate (value of its donor and acceptor stocks, values of other stocks and input functions from which it also depends).

The second step is the *“flow computation algorithm”* which defines operations on material flows. These operations are again attached to each node. Together, they allow us to compute the flows in all the flow links of the map once the rates of the various processes are known at all times from the first step.

Taken together, these two steps allow us to define completely the state equations attached to a kinetic process map. Graphically speaking, each of these steps adds to the map various graphical symbols representing the various operations of the two computational steps. The kinetic map complemented in this way is called a *“Kinetic Process Graph”* and provides a full graphical definition of the model while keeping the intuitive qualities of kinetic maps.

It is apparent even from the above summary that KPGs have two operations attached to each of their nodes, one for flows and one for what we have called influences. This is obviously reminiscent of the double role played by each box of Figure 5-B in which each port is connected to two kinds of signals V and I. In fact, we could show that each of the nodes of a KPG, when transformed into a block diagram possesses also four connecting wires, classified in pairs (flow, influence) and thus acting as ports. By re-examining our previous paper, the reader will see that these nodes are the loose analogues of the circuit elements which we asked for in the legend of Figure 7.

The set of kinetic process graphs includes the set of all SF-diagrams which are just the special case of (N,M)-processes for which N = 1, M = 1 and the stoechiometry coefficients are equal to 1. Moreover, each KPG may be translated algorithmically into an equivalent STELLA-like SF-diagram. However, this SF-diagram looses all the nice properties of the kinetic map. Indeed, by insisting on the structure of interaction between flows of different natures, KPGs force us to adopt modelling viewpoints which are much more integrated than those of System Dynamics. This results often in models which, although possessing also a SF-diagram representation, would never have been obtained using the current method since, in this method, they would have been too maze-like and thus would have been simplified consciously or not to ignore most of the flow couplings of different nature.

I suggest that, in many cases, these simplifications are just … oversimplifications and that many of the most usual fields of application of System Dynamics would gain much by being re-visited with our new method. To end up on a practical note, I want to insist on the fact that the method of KPGs is inspired freely from the method of Bond Graphs used in classical engineering. Instead of trying to develop a KPG package based on a modularisation of Stella-like diagrams, it was thus much more easy and efficient to implement it a Bond Graph package. I have chosen to use the commercial simulation package 20-SIM™ which is a top of the range package for mechatronics and multiphysics simulation based on bond graphs. 20-SIM is sufficiently open to allow the developer to define all its own nodes and thus it allowed me to implement all the KPG nodes defined in [4]. Using this package and our KPG library named FP4U™ (Fuzzy Processes For You), it is possible to define a KPG model graphically like it is done in usual System Dynamics packages.

The transformation in a computational model is then automatically done and the program does many checks insuring the syntactic correctness of the model. A FP4U model may contain an arbitrary number of processes. Each individual process may be defined by a continuous or discrete equation. It may be deterministic or stochastic and it may even be represented by a set of Fuzzy rules (Sugeno types) which are especially useful to represent flows of intangibles and the interface between management rules and material flows. A KPG model may be defined hierarchically and each level in the hierarchy may contain sub-models, coupled to KPGs and to block diagrams. The simulation program included in 20-SIM is more powerful and faster than most of the current System Dynamics packages since it uses a mix of advanced numerical method including for instance stiff methods, mixed discrete-continuous integration and mixed algebra-differential systems. A 20-SIM model may be translated automatically in a MATLAB model and the simulation results may be written in files compatible with MATLAB. All these features result in a modelling and simulation package which, in my opinion is a credible alternative to those currently used in System Dynamics if the goal of the modeller is to develop a large and sophisticated model.

Until now, our main commercial applications (IDEA.SIM LTD) have been the following: human resource management for the EU communities; Kanbans in retail for the KSRC store chain (London); sustainable development in Lubéron for the Regional Council of Vaucluse - France and in the Pévèle community of townships (Lille area, France); prospective in the employment market for the Belgian vice-Prime Minister; coupling between evolution and population dynamics in multi-bacterial populations. Admittedly this is only a small set when compared to the huge number of applications developed since 1960 in the SD community. However, they give us an encouraging basis to suggest that KPGs are generic enough to support the whole range of applications usually encountered in SD. KPGs have been taught to about 300 students in three EU-supported MSc degrees (Bioengineering in Lyon and London, Medical Informatics in Athens, General Engineering in Lille). They have also been presented at Keynote Lectures at the European Congress of simulation EUROSIM – Helsinki, 1999, at the 6th World Congress on Bond Graphs ICBGM 1997, at the US Congress of the Society for Computer Simulation, Phoenix, 1997. Finally, they were presented briefly, at the 2002 Congress of the System Dynamics Society (Palermo).

Trademarks mentioned:

STELLA: trademark of High Performance Systems, Hanover, New Hampshire, USA
20-SIM: trademark of Control Lab Products, Enschedde, the Netherlands
FP4U: trademark of IDEA.SIM, Harrow, United Kingdom
MATLAB and SIMULINK: trademarks of The Math Works, Natick, Massachusetts, USA

Bibliographical References

[1] Richmond B. : System Dynamics, let's get on with it, Syst. Dyn. Rev., 10, 135-157, 1994
[2] Forrester J.W.: Industrial Dynamics, MIT Press, Cambridge, Mass. 1961.
[3] Richmond B. : An introduction to System Thinking, HPS, 1997
[4] LeFèvre J. : Kinetic Process Graphs, Proc. of the XXth Int. Conf. of the Syst. Dyn. Soc., Palermo, 2002
[5] LeFèvre J. : Modelling the world with Kinetic Bond Graphs, Proc. Int. Conf. on Bond Graphs , Tucson, 1997, pp.3-9
[6] Roberts N., Andersen D. et al. : Computer simulation, a SD approach, Addison Wesley, Reading, Mass.,1983
[7] Hannon B. and Ruth M. : Modeling biological systems, Springer, NY, 1997
[8] Ford A. : Modeling the environment, Island Press, Washington, 1999
[9] Richmond B. : Stella « Getting Started Manual », HPS, 1997
[10] Pech R. : Foxes, rabbits, alternative prey and calicivirus, J. of Applied Ecol., 35, 434-453, 1998
[11] Forrester J.W.: World Dynamics, Wright Allen Press, Cambridge Mass., 1971
[12] Sterman J.D. : Business Dynamics, Mc Graw Hill, Boston, 2000
[13] Andersen D.F. and Sturis J. : Chaos in management models, Syst . Dyn. Rev ;, 4, 218-245, 1988

Back to the Top