

scMod/Sim.exe

For Supply Chain Modelling and Simulation

J. Manuel Feliz-Teixeira

March 2007

GEIN – Section of Management and Industrial Engineering
Faculty of Engineering of the University of Porto, PORTUGAL

Url: <http://www.fe.up.pt/~feliz>

Email: feliz@fe.up.pt

KEYWORDS: supply chain, modelling, simulation, flexibility, event scheduling, computer application.

ABSTRACT

In this paper we present an application dedicated to modelling and simulating Supply Chain systems (*scMod/Sim.exe*), developed with C++, with which a wide number of structures and policies can be analysed. The model construction is based on primitives resembling the real objects involved in these systems, like trucks, transport paths, retailers, factories, warehouses, stocks, etc. The result is a very simple environment for representing the Supply Chain in a wide range of situations. The simulator application, with its various tools for automatically representing the relevant metrics, gives the analyst an excellent opportunity for studying the behaviour of his/her Supply Chain, from the operational basis to the level of strategic decisions. In addition, the simulator includes a tool for helping to create the *rigidity matrix* of the Supply Chain (Feliz-Teixeira & Brito, 2004), with which the analyst can get an idea about how much his/her system will be *flexible* to demand variations.

used in *Petri nets*, *System Dynamics*, UML, etc., or even the wide spread *state transitions diagrams*. Building a model therefore implies the appropriate ability to construct based on such blocks. Of course, today it is easy to agglomerate a certain number of these blocks and assign them the “self” of an “entity”, using the properties of inheritance offered by the modern object-oriented languages, for example, but even so the modeller must have some knowledge about the basis behind these approaches, since it is frequently necessary to readjust such “entities” to the specific contours of a project. A substantial number of companies purchase generalist modelling tools with the intent of addressing specific problems, but later realise that it would also be necessary to contract a specialist on modelling, since the system to model reveals itself to be considerably more complex than expected. The usual destiny for such modelling systems is therefore the wastebasket. Exceptions exist, of course, among companies with a high capability for management and contracting, where groups of engineers are specifically dedicated to modelling and simulating the most relevant processes. This emphasis on the simulation as a matter for specialists seems an interesting policy, and for sure the results emerge accordingly.

1. Introduction

Developing a model commonly implies the representation of a system with the help of general purpose primitives, like the basic blocks of logic

2. About *scMod/Sim.exe*

The double application named *scMod/Sim.exe*, comprised of *scMod.exe* (modelling) and *scSim.exe* (simulation), aims to dissolve the need for special skills in building the Supply Chain model, while at

the same time maintaining the need for reasonable skills in handling the simulation process. The entities of the Supply Chain are represented as much as possible as they appear in “reality”, thus, no special knowledge is needed to construct a model other than that of creating a replica of the system based on “scaled elements of the reality”. This could perhaps be interesting for Supply Chain specialists who have few skills in modelling, but a good background in simulation.

In effect, each object is intended to be seen by the user as an “imitation” of an object existing in reality, with a certain functionality encapsulated, instead of exposed to the modeller. Thus, the level of abstraction required in the process of modelling comes closer to that of a planner. At this level, one does not need to define or represent the dynamics of the internal processes of the Supply Chain elements, since they are pre-defined in the respective objects. The user only has to tailor these generalised objects – named *Customer Supplier Units* (CSU) – in order to represent the particular Supply Chain with the accuracy needed. The next figure shows various CSUs prepared to be linked in a future Supply Chain network.

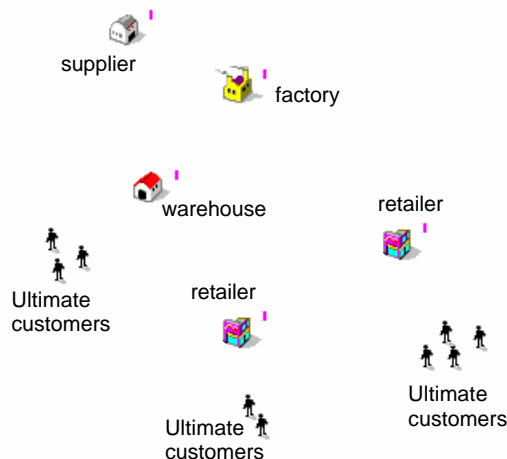


Fig. 1 Elements for creating the Supply Chain model.

This approach may also be considered a “data-driven” approach, since there is no need for programming in order to model or to simulate the system. The proposal also reflects the tendency of providing the industry with simulators for specific areas of knowledge, trying to imitate the natural specialisation observed in the business world. As there are companies specializing in airport projecting, urban traffic design, nuclear central building, etc., our point of view is that it would be

advantageous to reproduce the same tendency in industrial simulation tools, instead of systematically developing models from the basis. This would provide the modelling community with an extensive and very useful “data base” of specific simulators, for specialist planning. A huge data base of highly reliable specific simulators is what we modestly intended to contribute to. More detail on the processes involved in this application can be found in Feliz-Teixeira & Brito (2003) and in Feliz-Teixeira (2006). A visit to the author’s website could probably be of interest too: the site can be accessed at <http://www.fe.up.pt/~feliz>.

3. Modelling with *scMod.exe*

Here we will try to give an idea of how easy it is to create a model of a Supply Chain, even when it is to represent a complex network. This will be done with the aid of some images taken from *scMod.exe*, the application for modelling (Fig.2).

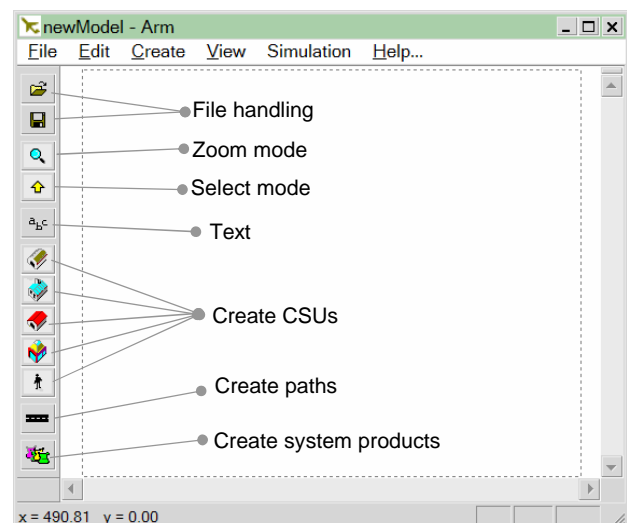


Fig. 2 A general view of the *scMod.exe* application.

The Supply Chain model will be represented in the application’s window with the aid of the buttons of the main *toolbar*, which allows not only the creation of diverse Supply Chain facilities (CSUs), but also the products, sub-products and raw materials available to the system, and the transport paths for connecting these CSUs.

One can basically say that to build the Supply Chain network will be as simple as to place into the application’s window the various Supply Chain facilities, that is, factories, warehouses, retailers, etc., and then connect them in the appropriate way. Notice that the dimensions of the “*operations theatre*”, the space where the system lies, will be

represented in scale to the reality. The user establishes these dimensions when a new model is created. From then on, when the user moves the mouse around this area, he/she will see the dimensions (x, y) in kilometres displayed down at the application's *status bar*.

Next figure (Fig. 3) shows a new Supply Chain represented in the *scMod.exe* window, with 2 suppliers, 2 factories, 2 warehouses and 6 retailers. The retailers supply several customers, and these will be responsible for injecting the demand into the system. Notice that the facilities are connected via *transport paths*, with distances, maximum speeds, road charges, etc., previously configured by means of a property dialog box.

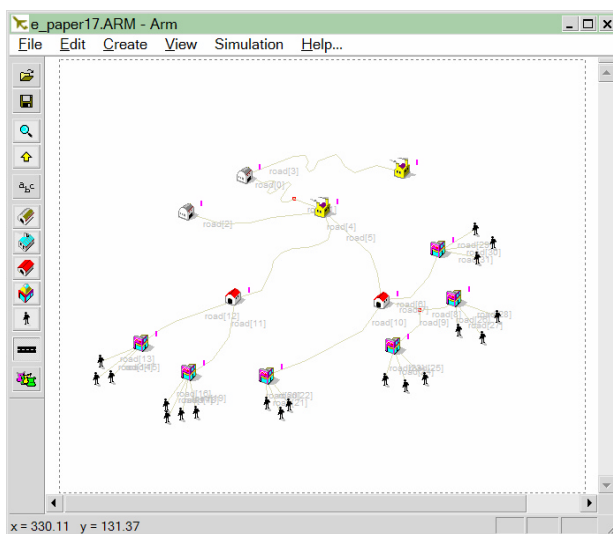


Fig. 3 A new Supply Chain represented in *scMod.exe*.

Once the Supply Chain network has been represented in the application's window, one must prepare for the implementation of the following steps: (1) define the products available to the system; (2) establish the type of transport paths between each pair of facilities; (3) Create the network nodes; (4) configure each facility to handle its particular products; (5) define the demand at each customer. In the next sections, we will try to explain how these steps can be completed.

As we see, after representing the network much more time will be consumed in carefully and properly configuring the various elements. And this task will get harder with the increase of the Supply Chain complexity and the number of products considered. In each facility (CSU), at least the *ordering policy* specification and the *supplier* specification are needed for each product handled

in the facility.

Concerning the *transport paths*, they may also be configured for a specific distance, giving the modeller the ability to “force” certain paths to a convenient distance of travel. This is very useful when certain facilities are located close to each other, but others are not, as for instance, in certain complex Supply Chains including regional and trans-national links, as in the case of ZARA fashion clothes, for example, where some suppliers are located in China, while the main factory and a great number of retailers are located in the Iberian Peninsula.

3.1 Products available to the system

The products available to the system, or *system products*, can be created and accessed using the button at the bottom of the main toolbar:

This option, which must be the very first option to use when configuring the Supply Chain, gives access to a *dialog box* (Fig. 4) where the user may name, characterise and link products, sub-products and raw materials, as well as establish among them a kind of *Materials Resource Planning* (MRP) tree, if necessary. The example shown in the figure is purely didactic.

ref	type	name
P[0]	P	water
P[1]	P	air
P[2]	P	plants
P[3]	P+	life

subproducts:

qt	subproduct
= 5	water
= 10	air
= 100	plants
	life

Name: life P[3] Quantity: 0 update

☐ raw material (R) Typical Load/Unload time per unit (min): 1 ± 0.2
☐ basic product (P)
☒ sec. product (nP+)

other data:

Base price/unit (\$): 1000 ± 0

Life Cycle (days): 10000 ± 0

Next Gen. (days): 5000 ± 0

Product units per container: 1

add update delete Close

Fig. 4 Dialog box for creating system products.

Notice that these products may or may not be used by the CSUs, depending on if they are handled or not handled by its inventory. Notice also that in the example of the figure, the secondary product named “life”, is defined as 5 “water” + 10 “air” + 100 “plants”. During the simulation process, this

information will be considered invaluable to good stock management.

3.2 Type of transport paths

After connecting the appropriate CSUs with *transport paths*, these may be configured in order to represent *roadways*, *railways*, *airways*, *seaways*, *information paths* (used to order by catalogue, by telephone, or by Internet), or *xTunnels*, a special kind of path for modelling pipeline-like flows. In addition, some other properties can be configured, like the exact length of the path, the maximum speed allowed, charges, type of charges, etc. Figure 5 shows the *dialog box* where all this can be done.

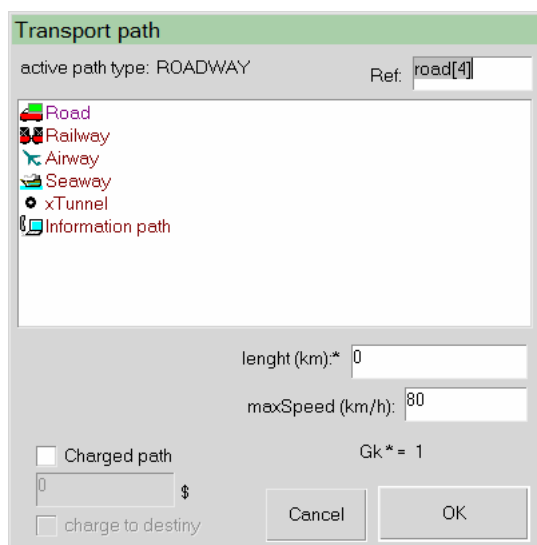


Fig. 5 Dialog box for creating system products.

3.3 Creating the network nodes

The network nodes, very important elements for the process of simulation, are automatically created in the application in response to the menu option "*Create->SupplyChainNodes*". This creates the nodes based on the proximity of the endpoints of the various paths, associates paths with nodes and nodes with paths, and establishes links between nodes and CSUs. This ensures that all the elements in the system will be correctly interconnected. Some nodes can then be configured to act as points of pause in the transport network, or even points where the material is transferred from a certain type of vehicle to a vehicle of another kind. This is very useful when modelling interfaces between normal roads and the railway, for example.

3.4 Configure each facility to handle products

Figure 6 shows the *dialog box* for configuring the products that the CSU will handle in its stock (or inventory), the maximum available space for those items, and their initial stocks. Notice that on the left list of this *dialog box* all the previously created system products are shown, while on the right side list appear only those particular products the CSU handles.

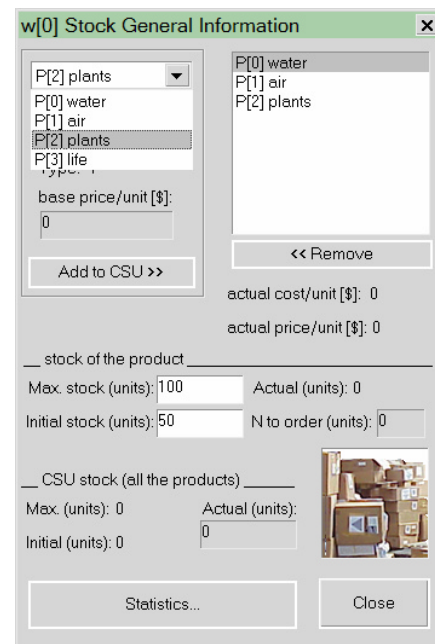


Fig. 6 Dialog box for configuring the CSU's stock.

The second most important task in the configuration of the CSU products has to do with the choice of the *ordering policy* to be used in each product, and the choice of its *supplier*. This is perhaps the part of the configuration process where the modeller must be more cautious, since small deviations from the appropriate representation of such policies will possibly lead the outputs to significantly diverge from reality. In effect, the choice and the configuration of the inventory model will definitely affect not only the internal dynamics of the CSU, but also the traffic on the transport paths and the pressure on suppliers. In a complex structure of CSUs, such mistakes of modelling may well produce chain reactions where from no reliable conclusions can be obtained. So, it is principally through this *dialog box* (Fig. 7) that each CSU is customized in order to behave the best possible in accordance to the reality.

Notice that for each product handled, there are basically six fields of data to be defined: (1) *when to order*, which identifies the method used in the

time domain to define the moment for ordering from the supplier; (2) *How much to order*, which identifies the method used in the volume domain to compute the amount to order; (3) *how to order*, specifying if the order is to be sent to the supplier *now* or if it can go at the end of the day with other orders; (4) *who to order from*, choosing a supplier for the product; (5) *cool time*, the lead time from the supplier that would be considered 100% appropriate; (6) *one order cost*, as the cost of ordering from the supplier.

Fig. 7 Configuring the order policies, the supplier, etc.

A similar *dialog box* is used for *orders of production*, if the facility is to include a production process. Notice, however, that most of these parameters will not be taken into account when using historical data, imported from a text file.

3.5 Defining de demand at each customer

Since the customer (ultimate customer) is also a CSU, it will be configured in the same manner as any other CSU. Nevertheless, it will not handle any stock resources or production facilities, and it is only allowed either a cyclical ordering of a certain amount of product (see again figure 7, and the *by cycle* option), with the frequency given in days, or,

instead, ordering based on historical data. The ultimate customer is, however, an element with which the demand at a certain facility can be made as complex as needed, using several customers to demand different amounts of product at different moments. Each *customer* is also able to generate *demand steps*, that may be used to help construct the *rigidity matrix* of the Supply Chain, a way for estimating the *flexibility* of the overall structure (see Feliz-Teixeira & Brito, 2004).

4. Simulating with *scSim.exe*

The simulator application (*scSim.exe*) looks very similar to the modeller application, apart from the main *toolbar* which now only gives access to some file handling options and to the configuration of the system products. In the menu of the application, the most important option is the “Simulation” option, which leads to the simulation start. Notice that the simulation may be set to be *deterministic*, instead of *stochastic*, if the option “Simulation->UseRandom” will not be checked. Notice also that a model must have been created previously with *scMod.exe* in order to be opened and simulated with *scSim.exe*, represented in figure 8.

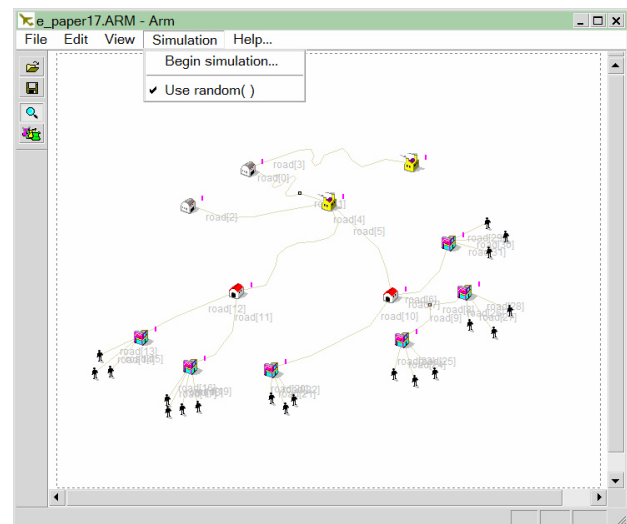


Fig. 8 A model in the *scSim.exe* application.

When the user chooses the option “Simulation->BeginSimulation”, another toolbar appears at the bottom of the application’s window, from where the simulation process will be controlled (Fig. 9). Most of the options in this toolbar require no explanation, since it resembles a TV controller. That is, once the “start” button is pressed, a sequence of actions is started, first testing the consistency of the model, and then connecting and preparing its objects for

the simulation process. In the end, the process starts and movement can be seen in the application's window.

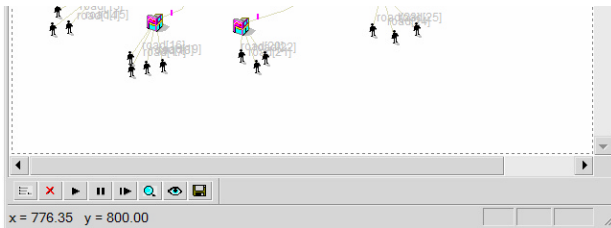


Fig. 9 The simulation toolbar.

During the simulation process, it is possible to inspect how the important Supply Chain indexes are performing, for example, by simply clicking on a CSU with the *right-mouse-button* and choosing the appropriate option from the menu list that would meanwhile appear (Fig. 10).

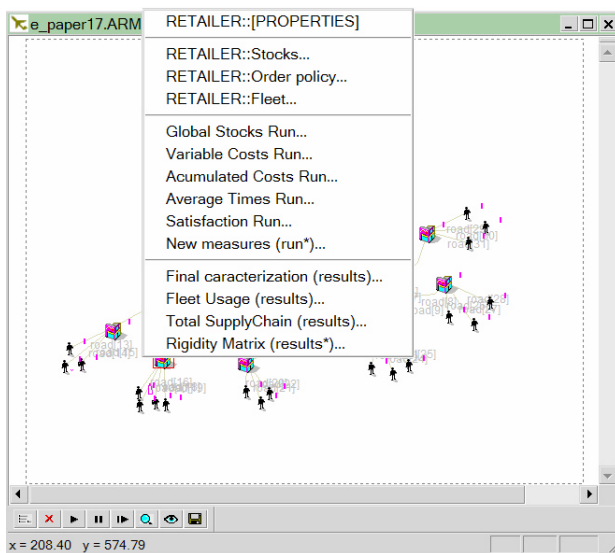


Fig. 10 Diverse parameters can be inspected during the simulation process, and the conditions reconfigured.

The next figure shows as an example on how the global inventory is performing at the facility $w[0]$.

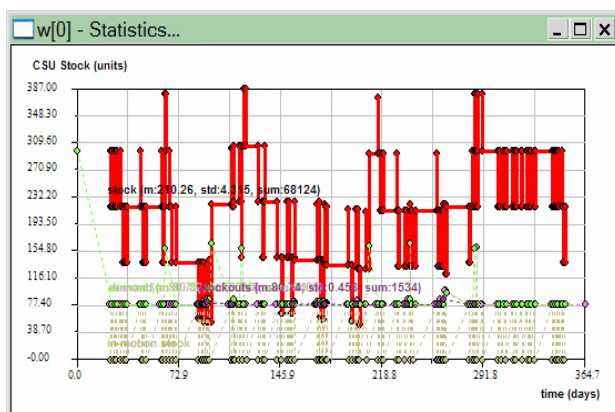


Fig. 11 A dynamic view of the inventory at $w[0]$.

In addition to these, several other graphics can be visualized even while the simulation is running, providing the analyst with a powerful tool for inspecting the behaviour of its system.

5. Metrics and outputs

The outputs of this simulator are firstly focused in a group of relevant operational measures, like the behaviour of the *global inventory* (all items), *variable costs*, *accumulated costs*, *average times*, *satisfaction with suppliers*, etc., with which the analyst may easily estimate the performance of each facility. Then, there is a second group of metrics which are computed from these, raising the analysis to a certain *operational-tactical* level, like the standard measures of *turnover*, *service level*, *stockout ratio*, among others (see Fig. 12). Finally, a tool for helping to measure the *flexibility* of the overall Supply Chain is included, allowing the studying of the network structure as a whole, an important instrument for strategists.

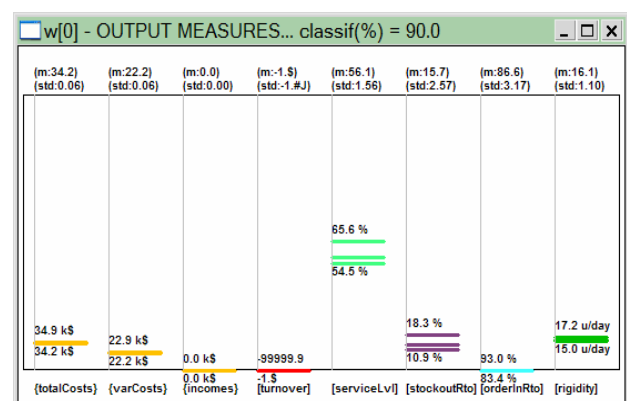


Fig. 12 Some standard measures¹ related to $w[0]$.

6. The rigidity/flexibility matrix

As told before, the simulator includes the ability of helping to compute the *rigidity* of the entire Supply network in the form of a matrix (Fig.13).

¹ In this case the *turnover* was not computed, since no base prices were given to the products.

	<NO STEPS>	r[0]	r[1]	r[2]	r[3]	r[4]	r[5]
s[0]	10.9 ± 4.3	f 00	f 01	f 02	f 03	f 04	f 05
s[1]	16.9 ± 7.1	f 10	f 11	f 12	f 13	f 14	f 15
f[2]	85.9 ± 29.5	f 20	f 21	f 22	f 23	f 24	f 25
w[1]	45.0 ± 6.6	f 30	f 31	f 32	f 33	f 34	f 35
w[0]	16.1 ± 2.2	f 40	f 41	f 42	f 43	f 44	f 45
r[0]	7.5 ± 0.5	f 50	f 51	f 52	f 53	f 54	f 55
r[1]	9.0 ± 3.0	f 60	f 61	f 62	f 63	f 64	f 65
r[2]	7.2 ± 0.8	f 70	f 71	f 72	f 73	f 74	f 75
r[3]	8.9 ± 1.4	f 80	f 81	f 82	f 83	f 84	f 85
r[4]	7.7 ± 0.4	f 90	f 91	f 92	f 93	f 94	f 95
r[5]	8.9 ± 1.4	f 100	f 101	f 102	f 103	f 104	f 105

Fig. 13 Matrix of the “residual” *rigidity* of the system.

This matrix is computed by means of generating *steps of demand* at each customer, which introduce a perturbation in the system, and then this perturbation is measured along all the network nodes. At the end of this process, a matrix can be drawn with the results. Notice, however, that only the “residual” rigidity of the system has been measured in the present case, that is, the rigidity exhibited by the facilities without any abnormal perturbation (no-steps). Anyway, from this matrix we may already conclude that the worst rigidity observed was at the factory *f[2]*, exhibiting around 85.9 units/day of stock imbalance.

7. Conclusions

The modelling/simulation application presented in this article, specifically devoted to Supply Chain, resulted in a powerful tool for analysis based on an intuitive ambience of interaction with the user. Several unusual tools are included, like the *demand steps* for helping computing the *flexibility* of the overall structure in terms of a *rigidity matrix*, transport paths for modelling *pipeline* type flow of materials, facility related metrics, etc.

The application, made of a modeller *scMod.exe* and a simulator *scSim.exe*, was developed by the author in C++, and is considered a *beta* version academic software. The simulator *scSim.exe*, however, is available as free software, and can be used for directly running and testing models previously created with *scMod.exe*. We invite interested people to try to simulate their own Supply Chain models in this way.

Authors Biography:

J. Manuel Feliz-Teixeira graduated in Physics in the Faculty of Sciences of University of Porto, Portugal, and received an MSc and PhD from the Faculty of Engineering of the same university. His work has been related to various matters, from optical communications, solar energy and seismology to, more recently, the simulation of complex systems in management science, like warehouse and supply chain. His PhD thesis is on “Flexible Supply Chain Simulation”.

References:

- Feliz-Teixeira, J. M. (2006). *Flexible Supply Chain Simulation (published thesis)*. Porto, Portugal: Publindustria Publisher.
- Feliz-Teixeira, J. M., & Brito, A. E. S. C. (2003). *An Approach for Dynamic Supply Chain Modelling*. Paper presented at the 2003 European Simulation and Modelling Conference, Naples, Italy.
- Feliz-Teixeira, J. M., & Brito, A. E. S. C. (2004). *On Measuring the Supply Chain Flexibility*. Paper presented at the 2004 European Simulation and Modelling Conference, UNESCO, Paris, France.