At the proceedings of the Industrial Simulation Conference 2004, Malaga, Spain, Jun 07-09, 2004

Distributed Application for Supply Chain Management Training

J. Manuel Feliz Teixeira
António E. S. Carvalho Brito

GEIN – Section of Management and Industrial Engineering
Faculty of Engineering of the University of Porto,
PORTUGAL
emails: feliz@fe.up.pt; acbrito@fe.up.pt

Richard Saw

Director of Executive Development of
CLSCM, University of Cranfield
U. K.
email: r.j.saw@cranfield.ac.uk

KEYWORDS: Supply Chain, management games, "*Bear Game*", inventory policies, "*Bullwhip Effect*".

ABSTRACT

Here we present a computer application planned to be used as an interactive tool for Supply Chain Management Training. Implemented with Visual C++, this application embeds the "*Cranfield Blocks Game*" (Richard Saw[1], 2002) network structure, and uses precisely the same demand patterns as the manual version of the game. Anyhow, instead of reducing the play to 12 reorder cycles, as the manual version does, this application extends those patterns throughout the time till any number of reorder cycles, what let the results become more useful and interesting even for didactic purposes. At the same time, the present application substitutes the classroom table by the computer screen, and can be made to run in AUTOPLAY mode, meaning the game can also be played with only one player or even automatically, with no players at all. In a certain way, this comes closer to some kind of Distributed Supply Chain Simulation, apart from the fact each lead time is fixed, as the "*Cranfield Blocks Game*" states, and the AUTOPLAY stock policy is empirical.

As each application communicates with the SERVER using the TCP/IP protocol, the players can be spread by different computers and even placed at different geographic locations if connected to the INTERNET.

In the end of this paper, results achieved with an automatic running session and with the involvement of a group of students from the *Escola de Gestão do Porto* (EGP)[2] will be presented and compared.

1. Introduction

Interactive Games are nowadays being used as powerful tools to exercise the ability of people to manage a wide range of complex systems, from flight simulation to the control of the flow of water in dams under extreme conditions of raining, for example. Following such a tendency, Supply Chain Management also started to apply those techniques to improve the ability of the managers to resolve faster and more efficiently some problems they are expected to face in the real world. Training before to fit later is the philosophy of such an approach.

Related to the sciences of management one can already find games like the *International Logistics Management Game*[3] *(ILMG)*, the *Corporation: A Global Business Simulation*, the *BEEFEATER RESTAURANTS MICROWORLD*[4], as well as the *GOLDRATT'S GAME*[5] and many others, but it seems one of the oldest and most popular directed to Supply Chain Management is the "*Beer Game*", conceived at MIT in 1960 (Sterman1989), with which the *Bullwhip* effect (amplification of demand at the upper levels of a Supply Chain) have been observed, as well as the importance of communication between partners stated. Many of those games, however, are still being widely used as manual tools. The "*Beer Game*", for instance, can still be bought inside a box in which one will find written instructions, some other pieces of paper and some more little blocks representing the material units used for stocking. Nevertheless, at the moment it is also possible to play the "*Beer Game*" electronically, after David Simchi-Levi[6] and Phil Kaminsky[7] developed their *Web Based Beer Game*[8].

In fact, there are some inconvenient features on presenting management games in traditional manual form, being the most significant: (1) the space required to play the game; (2) the need of an extremely well organized *manual* communication scheme between partners, by which time synchronization will be ensured (usual the job of another person); (3) the usage of paper cards to communicate demands; (4) and the non existence of a common support where the overall supply chain behavior can be easily shown and seen by all the members, during the game or at its end. The record of stock data and the computation of other decision variables are also time consuming in the manual version, and can be highly improved using a computer application. The present application, focused on the the "*Cranfield Blocks Game*", and named "*SCGame(cranfield)*", has as the main objective to give the players such advantages.

Once it uses a CLIENT-SERVER technology, this application has been divided in the applications "*SCGame(cranfield)_server*" and "*SCGame(cranfield)_client*". We will present each of them in this paper.

2. The *"Cranfield Blocks Game"*

The original "*Cranfield Blocks Game*" uses a multi-level structure to represent the supply chain, thus being a more general game than its counterpart "*Beer Game*", which usually looks at the supply chain as a linear structure. Nevertheless, the primary idea of the "*Beer Game*" is maintained, and one could say the "*Cranfield Blocks Game*" can also be seen as a more complex version of the former.

The structure of the "*Cranfield Blocks Game*" (Fig. 1) is based on four *depots* (retailers) that are connected to two central *warehouses* and these warehouses connected to a single *factory*, which in turn is served by an ideal *supplier*. The intent of this configuration is to let the players have a better sense of some of the behaviour that stands for real supply chains, at least concerning the need to fulfil the demand of more than just one customers in each facility, and how such a need can already introduce a significant variability in the demand, turning more difficult to establish the optimal management policy, even when only one product is considered. That is the case of the game. As well, this configuration lets the results give an idea on how the response of the chain depends on the demand amplitudes, as different demand amplitudes are previously "established" at the last customers, directly connected to the depots.

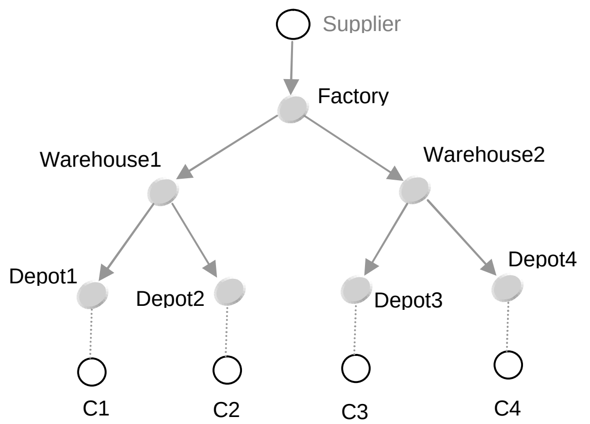

Fig. 1 Structure of the "*Cranfield Blocks Game*"

Usually, this game is played with seven groups of students, each group being responsible for the management of a single facility. The customers demand (*Cj*) are random generated, and the supplier (up on the figure) is considered an infinite source of materials. Thus, the student groups are only endorsed to manage from the *Depot1* to the *Factory*. No backorders are considered, and each facility must adjust its reorder quantity to a multiple of a certain *Base Quantity* (BQ) established at the beginning of the play and which increases from left to right and as the facility approaches the *Factory*. All the activity in the chain begin with the demand of products from customers (*Cj*), and the goal of each facility manager is to <u>fulfil demand keeping the *stock* as low as possible without incur in *stockouts*</u>. The average costumers demand amplitude also increases from left to right in the chain, meaning *C1* < *C2* < *C3* < *C4*.

The initial conditions of the game are the following:

Facility:	Demand	Initial Stock	Reorder (BQ)
Depot1	C1	40	20
Depot2	C2	60	50
Depot3	C3	40	40
Depot4	C4	100	80
Warehouse1	…	120	100
Warehouse2	…	200	120
Factory	…	200	200

Fig. 2 Initial conditions for the "*Cranfield Blocks Game*"

And the original customers demand patterns for the 12 cycles of the manual version is:

C1 = 20, 0, 10, 20, 20, 20, 10, 20, 0, 0, 20, 20.
C2 = 30, 40, 10, 20, 40, 50, 40, 10, 20, 50, 40, 30.
C3 = 30, 30, 30, 40, 20, 10, 10, 20, 30, 10, 30, 20.
C4 = 10, 50, 30, 50, 60, 50, 20, 30, 10, 20, 50, 40.

These patterns will be maintained in the computer version precisely till the 12th cycle, but then they will be replaced by patterns generated based on random normal distributed numbers. This aspect was strictly preserved, for that results achieved with the computer version could be compared with the results obtained with the manual version.

Finally, in order to be able to compare the management of different facilities, there was introduced a classification in the computer version that probably can diverge from the criteria used by the father of this game. For our own purpose we decided to classify the management of each facility by the following criteria:

CLASSIF = 100 x (*totalDemand* – 1.5 x *totalStockouts*) / *totalArrived*

Were *totalDemand* represents the amount of demand at the end of the game, *totalStockouts* represents the amount of *stockouts*, and *totalArrived* represents the amount of material arrived from the supplier. Thus, this classification pretends to give an idea or the efficiency of the management, assigning a negative weight factor of 1.5 to the *stockouts*.

3. The CLIENT-SERVER application

To implement this game we have developed a CLIENT-SERVER application communicating by TCP/IP, being the SERVER the responsible for the customers demand generation, the initialization of the game, the synchronization and control of time and the communications between partners, and the CLIENT being the interface each manager has to play and manage the stock resources of his facility.

Before the beginning of the game, the SERVER (Fig. 3) must be launched in the Operating System (WINDOWS) for that it will wait for the clients to ask permission to connect. As each client asks this permission, the SERVER will assign to it a free facility,

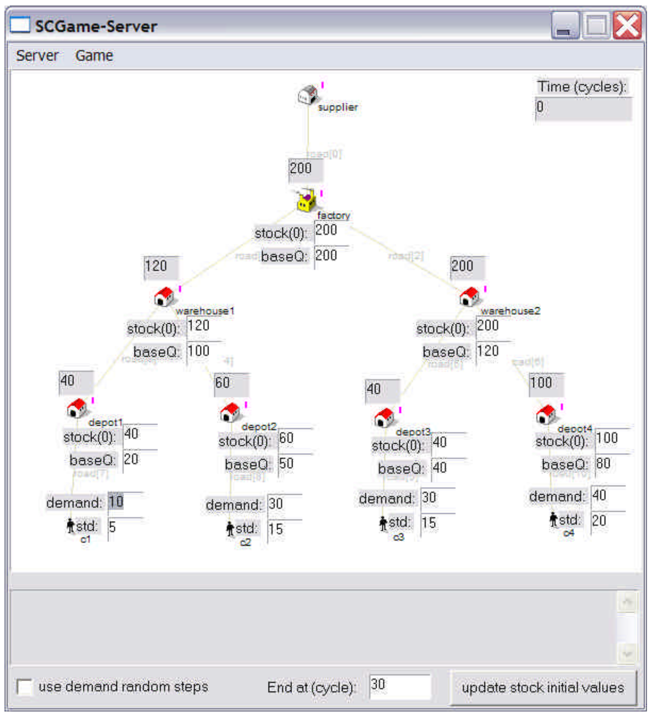

Fig. 3 Supply Chain SERVER "*SCGame(cranfield)_server*"

and will refuse any connections once the supply chain is complete. In that moment, the SERVER will then send a message <u>*msg::play*</u> to each CLIENT, to signal the game can be made to start. The time will only advance to the next cycle whenever all the CLIENTS have played the present cycle, thus ensuring the synchronism between them is maintained.

The SERVER is the centralised controller of the game. It not only establishes the access to the players and ensures the correct time holds for all of them, but also it controls the flow of demand between partners. This is achieved by the simple exchange of messages between the SERVER and the CLIENTS.

As it can also be inferred from the previous image, this version can be made to run for longer periods of time than the original 12 cycles, as well as it allows the initial stocks of the facilities, their *Base Quantities* (BQ) and the customers' demand to be modified, even if actually they will start with the values of the manual version. Finally, this computer version gives the manager the possibility to introduce "random steps" of demand at each customer's demand site, what probably can be used to train reactions to high instable markets, or even to make experiments concerning the flexibility or the agility concepts.

As the game runs, it will also be possible to make an idea of the overall supply chain behaviour, once the stock of the facilities can be shown graphically to the user, as the example of figure 4.

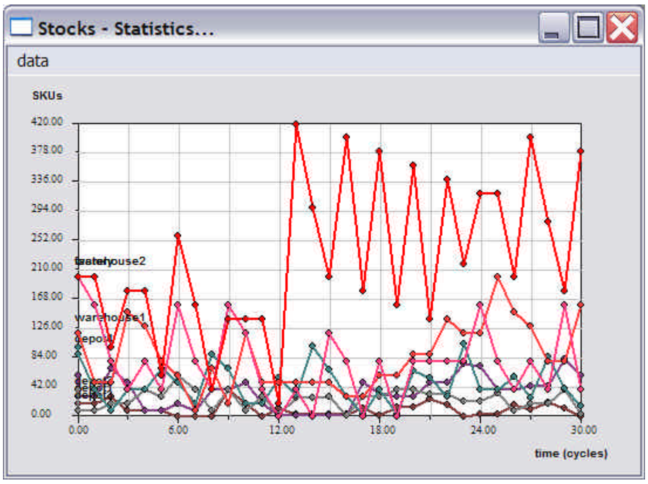

Fig. 4 Example of the stocks of facilities shown at the SERVER

As suggested in the next figure (Fig. 5), the game can be usually played in a room with each group of students around a computer, while the SERVER runs in another computer nearby, many times with its display projected in the wall as a way to let the groups access the evolution of the supply chain data, mainly

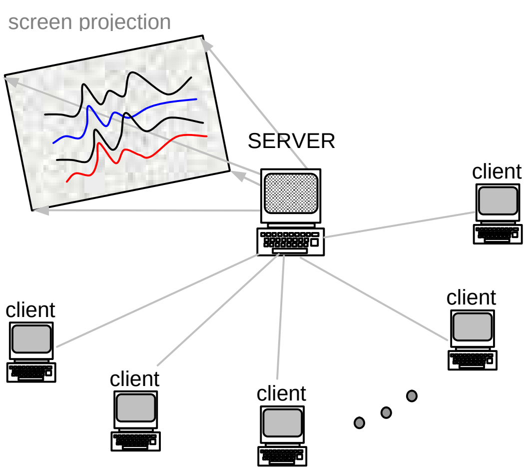

Fig. 5 Example of a simple apparatus to play the game

the stock levels of the partners. Anyhow, the game can also be played by internet if the SERVER is previously running in a reachable address.

On the other hand, the CLIENT application is the tool with which the groups manage their facilities, as well as they communicate the demand to their suppliers and serve their customers requirements. This application, less complex than the SERVER, is the real player's interface, and can be made to run in any location of the computer network as long as it will be able to reach the SERVER by TCP/IP.

All the CLIENTS have the configuration shown in the next figure (Fig. 6), and differ from each other by the natural amounts of stock and BQs, as well as by the facility name and the designation of their customers and suppliers. The figure shows a CLIENT assigned to the *Factory*.

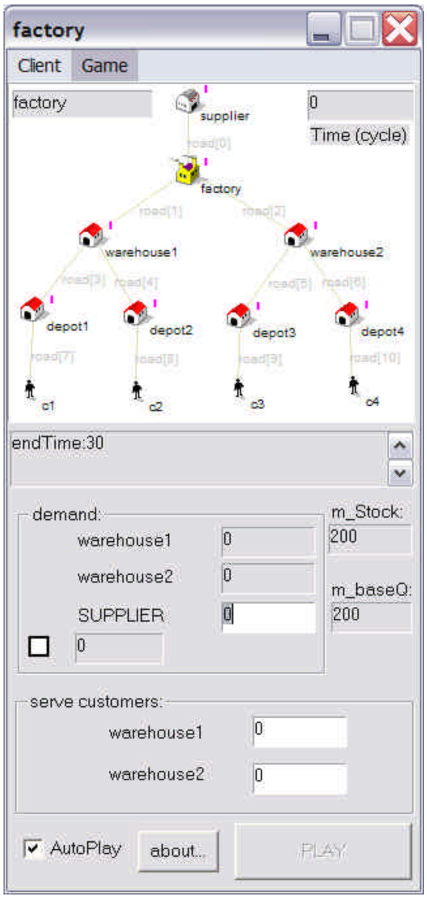

Fig. 6 The *SCGame(cranfield)_client* in this case assigned to the Factory

To help the player to locate better its facility in the supply chain, this application uses an image of the network where all the facilities are represented. Also there is a window where messages coming from the SERVER are displayed, and a check button of *AutoPlay* which lets the application react automatically to the customers demand and decide when to order new material. In any case, the criteria used on those decisions are empirical and do not follow any of the models known in practice for stock control, as the intention was to use this application to simulate the kind of "empirical" behaviour many times used in the practice for managing small stock resources.

At the same time, the CLIENT application has two other *zones* very important to the player, shown in figure 6: the *demand zone*, where demand from *customers* is continuously updated and the demand to the *supplier* affected, and a *serving zone*, where the user can fulfil the requirements of material to send to the customers. These operations will be executed when pressing the PLAY button, which

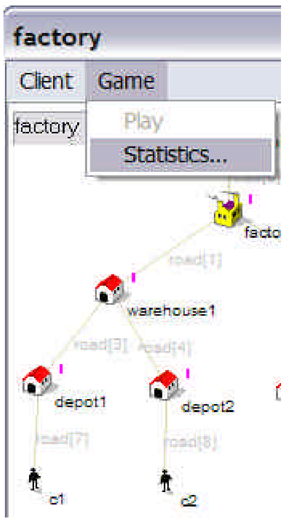

Fig. 7 Menu Statistics

then will automatically be disabled by the application and still in such state till the next time the SERVER will enable it, meaning this facility is again allowed to play.

Another feature concerning this CLIENT application is the possibility of tracking some important parameters during the game, as it is the case of the *stock*, the *demand*, the quantity *arrived* form the supplier, and the *stockout* level, what usually helps the player on making better decisions. This data will be presented in the form of a graph that will appear next to the CLIENT application as long as the menu option *Game->Statistics* is chosen (Fig. 7). A graph of this type, related in the case with *Depot1*, is shown in the next figure (Fig. 8).

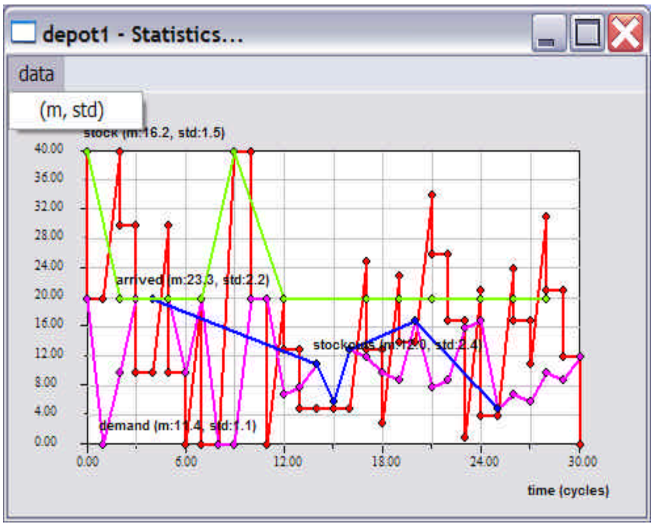

Fig. 8 Graph of *Stock*, *Demand*, *Arrived* and *Stockout* level

In addition to the visualization of these parameters, the player can use the menu option *data->(m, std)* of this window to automatically compute the <u>average</u> value and <u>standard deviation</u> of each of the data series represented.

4. RESULTS AND COMMENTS

As was said earlier, the results presented in this paper have been obtained in two sessions. In the first session the game was made to run automatically with all the facilities in AUTOPLAY mode, during 30 demand cycles. In the second session the game has been played by seven groups of students attending a Masters Course at the *Escola de Gestão do Porto* (EGP), and also run during the same 30 cycles.

In each case, data like the one showed in figure 8 was collected for each facility, and then its average values, standard deviations and accumulated values have been computed, as well as the final classification of the facility. These computed results were then recorded and handled in an EXCEL spreadsheet for the two separated sessions, in order to make the comparison between them easier. Notice that the unity of stock used here is the SKU, worldwide known as *Stock Keeping Unit*.

Different kind of calculations could be made based on the row output data retrieved from the game, but for our purposes we decided to present only some results based on averages and on standard deviations. Averages will give us the sense of how each facility was in general handling the interesting parameters, while the standard deviations will lets us have an idea about the variability observed on handling those parameters. Based on such simple results we expect already to be able to conclude about the interest of the present application tool.

In figure 9, for instance, the *average values* of those parameters are represented, computed after the game have been played the first time by the students. It shows that even if the final classification of the management of the facilities exhibit an almost flat behaviour, it was also observed an obvious tendency on the other parameters (except the demand, of course, due to be an input) to grow from the left to the right of the supply chain network, as well as when approaching the *Factory*. This is precisely the tendency induced in the system by the differences in the BQ quantities, as we have stated earlier. Thus, due to the fact these results tend to be proportional to the respective BQ of the facility, is seems all groups have used the same kind of empiric reorder policy, as well as it seems they have faced the same challenges on managing the facilities.

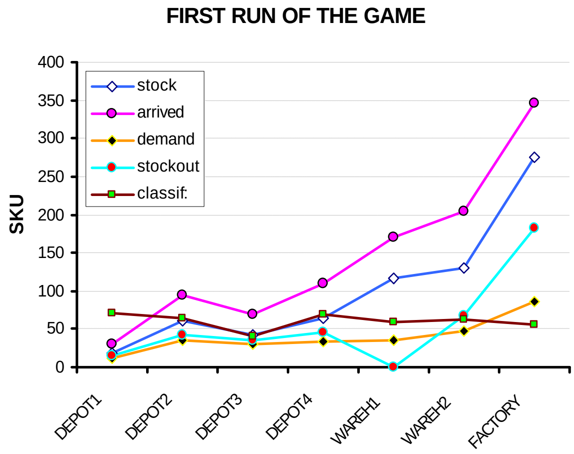

Fig. 9 Computed results obtained on the first Students Session

These results can be compared with the data obtained with the AUTOPLAY session, which is represented in the next figure (Fig. 10), where all the facilities were using precisely the same stock control empiric policy, previously established by us.

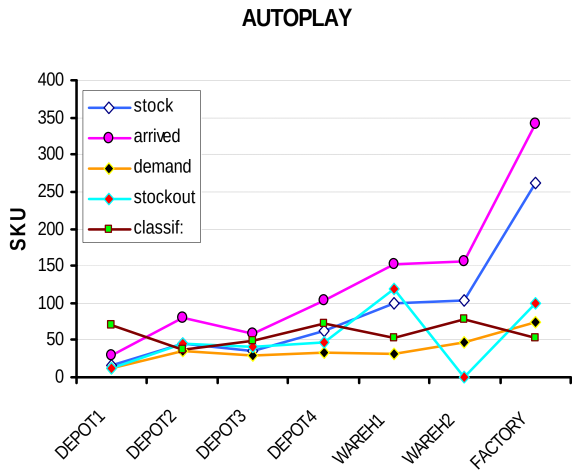

Fig. 10 Computed results obtained with AUTOPLAY Session

Although the various parameters seem to have improved slightly in the AUTOPLAY case, as well as the level of *stockouts* have been reduced, the two cases show a similar tendency to follow the induced BQ differences. Maybe this would mean that to manage this system in an empiric base few differences would arise due to the “experience” and the “quality” of the manager. In fact, in the two cases de classification parameter is maintained practically constant at all the facilities.

However, when we compare the variation observed in the two sessions by means of the standard deviations charts, shown in figure 11 and figure 12, a significant difference is already noticed. It seems obvious the "experienced" manager (AUTOPLAY) would be able to handle the facilities in a much more smooth way than a beginner. In fact, in the AUTOPLAY the variability of the *stockouts* was drastically reduced at the *warehouse1*, *warehouse2* and *factory*, as well as the amount of material ordered, even if not as drastically.

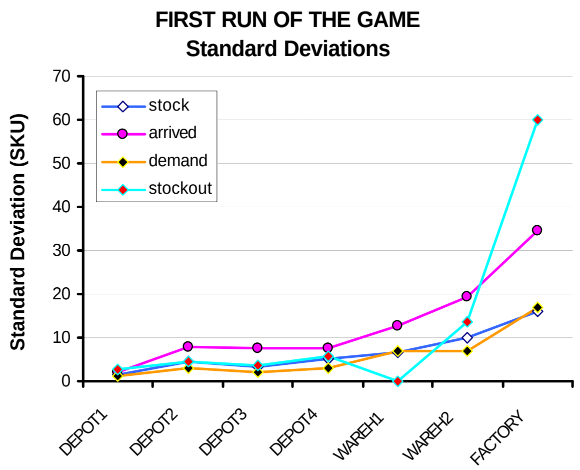

Fig. 11 Standard Deviations at the first Students Session

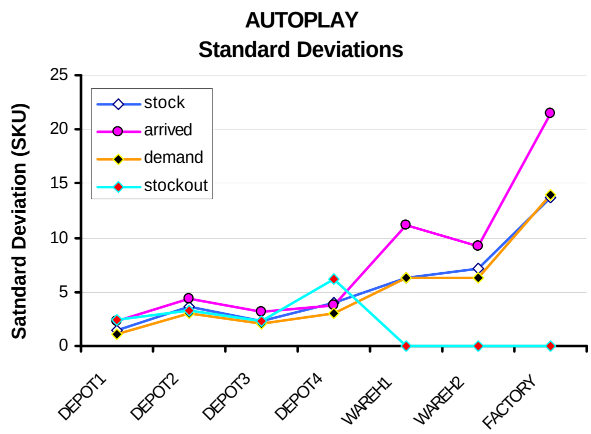

Fig. 12 Standard Deviations in the AUTOPLAY Session

Other measures could be achieved with this application, as it gives the user a wide range of untreated output values over time, so making the experiments of management dependent on the imagination of the game's supervisor. Also, as this game lets the players visualize and record the evolution of important parameters with the time, even testes concerning the *visibility* or *not-visibility* between partners can be made, depending whether the overall supply chain is projected or not-projected on a wall, where all the stock levels can be turned visible to every partners.

5. CONCLUSIONS

Based more on the potential of this application than on the results presented here (these are just an illustration of such a potential), we conclude this computer application represents an interesting tool for Supply Chain Management Training. In fact, the results achieved with the manual version can also be obtained using this tool, and in addition other interesting aspects are made available, like the portability and the flexibility of a distributed computer application, the capability of considering *visible* or *not-visible* scenarios, as well as the introduction of demand steps at the ultimate customers, which we imagine can be useful in future testes concerning studies of *Flexibility* or *Agility*, for example.

References:

[1] Richard Saw, "*Cranfield Blocks Game*", private communication, Centre for Logistics and Supply Chain Management (CSCM), University of Cranfield, 2002

[2] `http://www.egp.up.pt/`

[3] Robert W. Grubbström, "*International Logistics Management Game (ILMG)*", Linköping Institute of Technology, Sweden.

[4] Global Strategy Dynamics Limited, 2003.

[5] This game is described by Goldratt in his "novel", the Goal (North River Press, 1992, pp. 104-112.). It can be used to illustrate many concepts in Production Management.

[6] Professor at the Massachusetts Institute of Technology (MIT)

[7] Professor at the Industrial Engineering and Operations Research Depart., University of California

[8] `http://beergame.mit.edu/`