

Visual C++ software for warehouse simulation (an overview)

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

LASSIP – Simulation Laboratory for Production Systems
GEIN – Section of Management and Industrial Engineering
Department of Mechanic Engineering and Industrial Management.
Faculty of Engineering of University of Porto
PORTUGAL

February 1999

ABSTRACT

This paper presents an overview of a *visual and interactive warehouse simulator* developed with *Visual C++* for *Windows95/NT* operating system. This simulator have been designed under an *object oriented programming approach* and looks to the warehouse by an hierarchic decision level point of view, which allows an interesting separation of the responsibilities on the system. It is a modular approach where each entity is responsible for its own integrity and functionality, and where the decision rules are separated on three fundamental levels: *element level*, *control level* and *management level*. Nevertheless, the main interest of this paper is not to introduce any kind of code statements of the simulator, but instead to present the reader the structure of the more relevant elements and the way they were modelled. At the same time it will be shown some of the user interface facilities and also some references to studies made using this warehouse simulator.

1. Introduction

Warehouse automated systems are nowadays fundamental equipment to ensure the modernisation of production and distribution centres, to improve their flexibility and also to rise their processing capabilities in order to answer with more efficiency to the actual market demands.

The importance of simulation modelling in this field of industry is from all recognised. Anyhow, commercial software modellers appear to be or too simple to permit a reasonable modelling of the real system or too complex to consent to build it in a proper time, and for these two main reasons only few Portuguese enterprises stand making some efforts in the field of simulationⁱⁱ.

Nevertheless, with the evolution of programming languages towards *Object Oriented Paradigm*, given its decrease of price and its improving flexibility and also due to the excellent tools to build the user interface, simulation is becoming practicable, even in those cases implying specific development solutions. This simulator is in fact a case of a modeller directed for a particular area of interest, the automated warehouse systems.

2. Objectives

This *interactive warehouse simulator* was developed due to two main reasons: first, the tradition of knowledge in this laboratory on such matters, deep connected to the early work of António E. S. Carvalho Britoⁱⁱⁱ. Second, the obvious interest showed by EFACEC Portuguese company on our project of developing such a simulator.

Therefore, the main objective of this work was to project and develop a *visual interactive warehouse simulator* that could be used by that Portuguese company on its warehouse studies and building. The work have been started in 1996 and lasted for two years on conception and implementation of the elements and decision rules widely used on that company. Nevertheless, the final conception has been completely new, mainly due to the introduction of the *three levels decision logic* mentioned earlier.

The simulator flexibility and the possibility of configuring each element on the warehouse in a simple manner, as well as the capacity of simulating practical warehouse systems, were the main intents on the beginning of the project. Such demands implied a deep study of each of the elements to simulate, a carefully choice on the program structure and the development and implementation of certain algorithms for material moving inside the warehouse.

This led to a final *b-version* that is being used by that company since 1998 on its most important warehouse projects, two of them related with a study of a warehouse to Brasil^{iv} and a study for building a warehouse in Iran^v.

3. The simulator's base application

The base application, developed under Microsoft Visual C++, is a standard single document *Windows95/NT* application on which some useful tools were added. Such tools, however, were mainly separated on "drawing tools", by which the user is expected to draw the layout on the application's view, and "simulation tools" which are related with the creation and configuration of the warehouse elements, such as *conveyors*, *racking vehicles*, *transfers*, *transfer tables*, etc...

Next figure (fig.1) shows the main view of the simulator application.

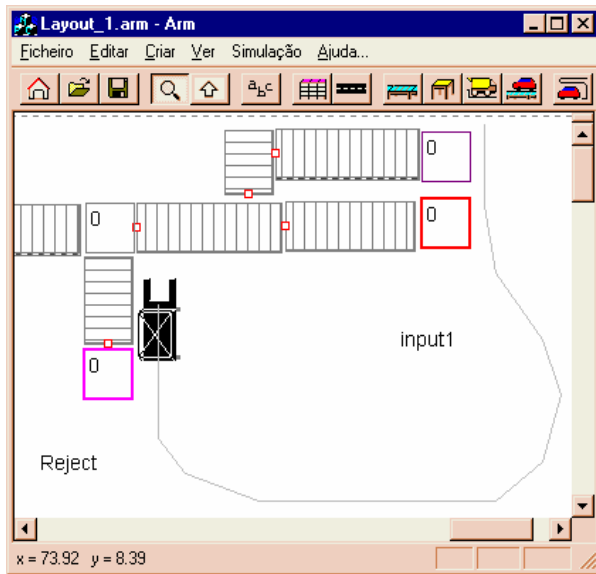


Fig.1

In the “drawing tools” were considered the following facilities:

- Interactive polyline draw with the mouse.
- Runtime creation of drawing objects.
- Scale representation in meters.
- Text inclusion in the interactive view.
- Facility of selection objects with the mouse.
- Facility of moving objects in the screen with the mouse.
- Facility of deleting and duplicating objects with the mouse.
- Facility of window zoom.
- Filter for element intersection.
- Continuous display of real coordinates.
- Floating menu to access the properties of each object.

These background tools let the user actions be very simple when drawing the scaled warehouse layout on the simulator’s interactive view.

In the “simulation tools” there were included the creation and configuration of the warehouse layout elements (conveyors, racking vehicles, etc.), the definition of the paths where vehicles are due to move during the simulation run, and the functions to analyse the overall layout’s data consistence. These tools also include some items to manage the simulation process.

Most of the “Simulation tools” can be accessed both from the application’s menu and from the top *toolbar* shown in the Figure 1.

4. Creating the warehouse elements

After creating a new simulator document on this application, where the user have previously defined its rectangular dimensions (in meters), there will be presented to the user an empty view where he is expected to draw the new layout with the help of the application’s tools. In this way, he will be responsible for the method or sequence in which he will draw the layout. As in the most cases there is no need to follow a specific order to create the objects, the user can do this task the way he prefers. In fact, he can first create the conveyors and then the racking systems, and then the racking vehicles, or to choose any other method of his own sympathy.

During this process the layout is being represented on the simulator’s view, staying all the elements accessible to the user to be changed, deleted or moved within the layout’s predefined space. Each warehouse element has associated a property box on which its characteristics can be entered or modified later.

4.1 Racking storage elements



Racking elements for palette storage can easily be created with the aid of one of the *toolbar* buttons and subsequent use of the mouse to define the racking dimensions.

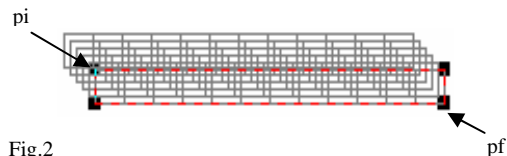


Fig.2

These dimensions are established by choosing the points pi and pf of the racking as represented in Figure 2 by means of a simple mouse action, which define the initial and final points of the racking. In this Figure it is also represented the vertical dimension, what lets the user easily visualise the racking overall capacity of storage. Accessing the property box of this kind of elements it is possible to modify the default parameters for racking storage creation. The overall *racking storage system* of the warehouse is then the group of all the *racking elements* created by the user.

4.2 Vehicle path network



Vehicle *paths* are polylines defined by the user along which *racking vehicles* are due to move while handling the jobs during the simulation run. As in the previous case, *paths* are easily created by interactive use of the mouse, what speed up de process of entering the layout’s data into the simulator’s window. The user draws each single *path* on simulator’s view, and the group of all the *paths* defines what we called the *vehicle path network*. *Paths* are also the elements through which the simulator associates *racking vehicles* to racking storage elements, and other vehicles to tables and input or output points.

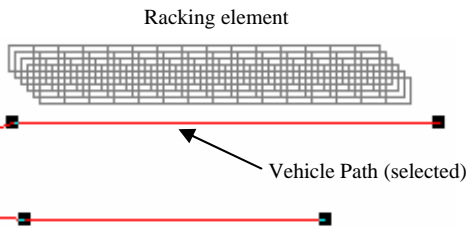


Fig.3

The association of *paths* to *racking elements* is an automatic process that runs before the simulation start, and the method of association is related with the distance from the *path* to the *racking* (fig.3). The same is true for transfer tables and input or output points.

4.3 Conveyors



Conveyors are the dynamic elements used to transport palettes through the warehouse and usually are used to connect the peripheral input and output points to the storage zones. As in the previous cases, *conveyors* are created interactively by the definition of only two points: the initial point (*pi*) and the final point (*pf*). The default direction of movement is from *pi* to *pf*, so defined by the user during the creation of the element.

Next figure shows a *conveyor* prepared to transport palettes from *left* to *right*. The small square in one of the conveyor's endpoints defines the default conveyor end, and then its default direction of transport.

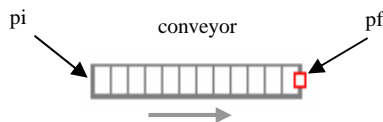


Fig.4

Other parameters of the *conveyor*, like its width or its speed or the unload time, are due to be modified on the element's property box accessed by a simple mouse DOUBLE-CLICK over the element.

Being a dynamic element, the *conveyor* is related with certain events that define its behaviour, which lead to the implementation of the related event handlers on this *class* of objects. This led to the possibility of observing the movement of the palettes on top of the conveyor during the simulation run.

4.4 Transfer tables



In general, *transfer tables* are the interface elements to be used between different kind of material handlers and in the case of bifurcation of the paths where the material flows. Material handlers are considered in this simulator to be *conveyors* and *vehicles*, and then, to ensure the transfer of palettes between them one must use *transfer table* elements. In this perspective, the *transfer table* is seen as a cell for intermediate storage of palettes, and so the

warehouse a site of a certain number of cells connected by *handler elements*. In this software version *transfer tables* are also considered static objects (with no events associated) which activity depends directly on the *dynamic elements* activity (*conveyors* and *vehicles*). Nevertheless, there is one kind of tables, named *Inspection Table*, that is considered closer to a dynamic element due to its active role on the simulation process. In fact, *Inspection Tables* are used to inspect the material and then to decide for its rejection or not.

Transfer tables are represented as rectangles whose dimensions are first specified by the user in the element's property box.

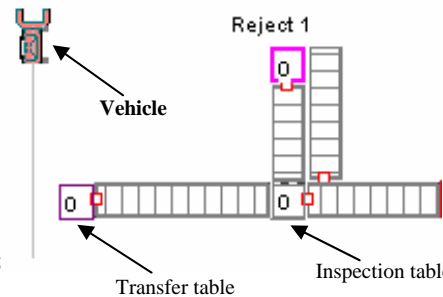


Fig.5

Figure 5 shows the two kinds of tables within a *conveyor* network interfacing with a distribution *vehicle*. It is important to notice that each *table* is previously configured to support certain directions for material flow (this is also defined by the user).

4.5 Input and Output points



Input and Output points are the elements by which the warehouse interfaces with the outside world. The material enters the warehouse through the *Input points* and leaves it through the *Output points*. Mainly these points are derived by the previous *table* concept, however they are active elements on the system due to its capacity of generating new events during the simulation process. Each of them is previous configured by the user by accessing its property box, where one can establish the palettes input or output rate as well as other parameters related with its physical dimensions and material flow distribution.

Although these elements were first created to play the role of inputting and outputting material, later they have been generalised to include two new cases: the *picking point* and the *rejection point*. The *picking point* is a special point where the material demands are generated manually by an operator, while the *rejection point* is a point to where the material will be directed in the case of rejection by an *Inspection Table*. Each of these special points can be seen as a mixture of input and output points, as the material can be redirected to the system after a certain time of processing inside that point.

As in the case of *Transfer tables*, each of the *points* considered here is represented on the simulator view as a rectangle which dimensions can be defined by the user accessing the correspondent property box.

Next figure (fig.6) shows a part of a layout that includes these kind of points.

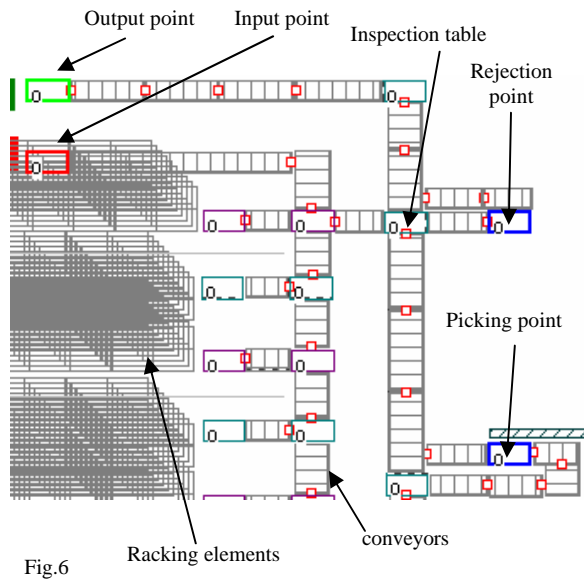


Fig.6

4.6 Vehicles



Vehicles are the dynamic elements that use *paths* to move through the warehouse while handling certain jobs. As in practical warehouse systems there is some different kinds of vehicles, in this simulator it was included five different types that were considered more relevant by the company EFACEC to ensure that this simulator could be able to model a wide range of layouts. All of these *vehicles* play the role of *material handlers*, although some are more adapted to certain jobs than others. For instance, there are *vehicles* with the capacity of accessing *racking elements*, while others only are able to move the material between *tables* or input/output *points*. Next figure shows the five types of *vehicles* included in this simulation software.

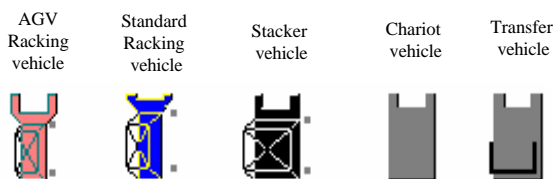


Fig.7

AGV Racking vehicles are *Automated Guided Vehicles* (AGV) with the ability to access *racking elements*, meaning that they are mainly used inside the storage zones of the warehouse.

Standard Racking vehicles are *vehicles* that move over static predefined aisles and also have the ability to access *racking elements*, also used inside warehouse storage zones.

Stacker vehicles are *vehicles* that usually have a relative freedom of movements, guided by an operator, and with the ability to execute stacking actions between different parts of

the warehouse, mainly between *tables* and input/output *points*. In this simulator this kind of *vehicles* is not considered to be able to access *racking elements*.

Chariot vehicles are *vehicles* that usually move over static inline predefined aisles and are mainly used as palette distribution elements, so acting between *tables* and input/output *points*.

Transfer vehicles are *vehicles* that move over static predefined aisles which main purpose is to transfer other vehicles from one place to another. These *vehicles* are usually associated with *Standard Racking vehicles* when it is necessary to transport these elements to different aisles. However, *Transfer vehicles* are also due to handle palettes in certain particular cases.

The user creates each of these *vehicles* with a simple mouse action choosing its type from a vehicle list included in the simulator. Once created the *vehicle* it is possible to configure its parameters using its property box, in order to adapt them to the desired values if different from the default values.

To each *vehicle* must be associated a group of *path elements* that will define the area within the *vehicle* can move later during the simulation run. In this way, it is possible to define different areas on the warehouse that will be served by different *vehicles*.

In the particular case of the *Transfer vehicle* the user needs also to associate it to the *Racking vehicles* that it will be responsible for.

4.7 Vehicle paths

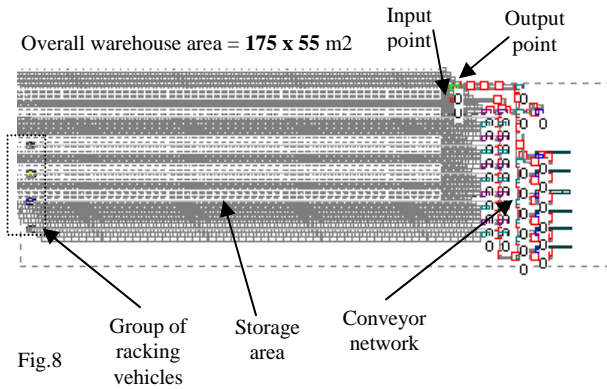


Vehicles need that the user associates them to a certain group of *path elements* in order to define their area of action within the warehouse. Therefore, for each *vehicle* the user must choose and select with the mouse a certain number of consecutive *elementary paths* from the overall *path network* previously created on the warehouse. Once this is done for all the *vehicles* the simulator can be considered prepared to start to simulate the movement of those *vehicles* along those *paths*. Any *vehicle* with no associated *path* will be considered inactive, remaining stopped during the simulation process.

5. An example of usage

This simulator was conceived to help the user creating a wide range of warehouse models without the need of programming new code. This facility arises from the fact that this simulator already includes in its simulation tools the most important elements used on common warehouses, through which the user can represent a wide range of layouts. Also, the inclusion of *Computer Aid Design* (CAD) drawing tools lets the user quickly represent and configure such elements. The functions that analyse the overall layout's data consistence are another important mechanism to speed up the modelling process.

In this section we present an example of a practical layout modelled with this simulator where there were included most of the elements presented earlier on this paper. The top view of this layout is shown in the figure 8 in a military perspective, where one can see the real dimensions of the warehouse, the racking storage area, the distribution conveyor network and other kind of elements.



This layout uses 5 racking vehicles working each on an independent raking path and serving the entire storage zone. The input and output points are the interface points with an external production zone, and the material flows in the conveyor zone passing through 6 picking points. The storage zone is a group of 10 double deep racking elements distributed by the 5 different raking paths and has an overall storage capacity of 11600 palletes.

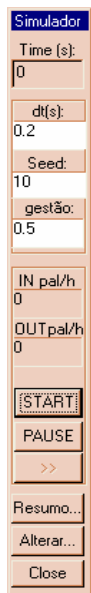


Fig.9

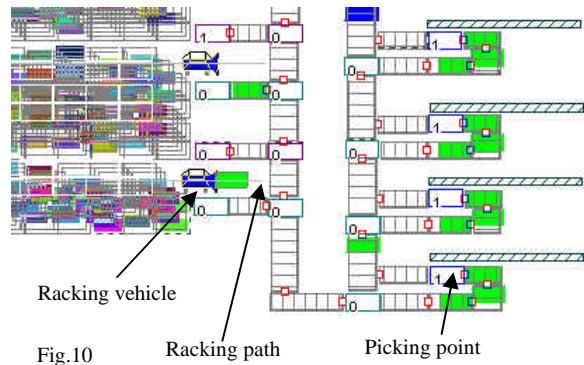
After the layout's configuration, the simulator fills the racking storage elements with a random distribution of palletes and then the simulation is ready to start. The start command consists of a simple pushing of one of the simulation control buttons (START) of the simulation control window that appears meanwhile (fig.9).

In this window the user is able to configure some static parameters of the simulation process, like the time interval of sampling for visual movement effects [dt(s)], the seed for random numbers generation [seed], and a management index [gestão] related with the method of choosing the position of the palletes within the warehouse storage.

This window will also keep a continuous display of the simulation time [Time(s)] as well as the value of the actual input and output pallette rates [IN pal/h] and [Out pal/h]. The other buttons are used to manage the simulation run and to let the user observe the evolution of the simulation output data [resumo].

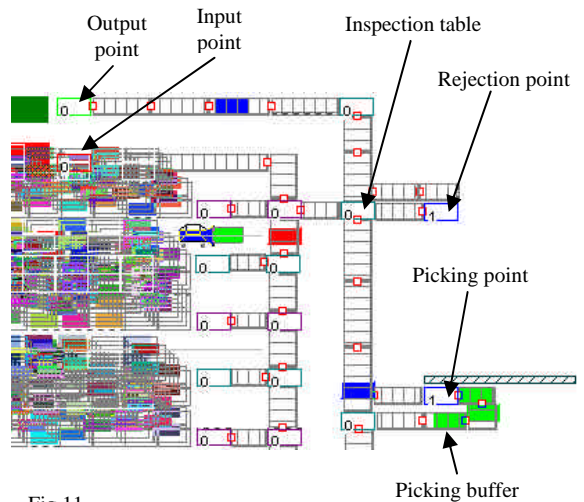
As this is an example related with a practical layout, all the elements have been configured according to the client requirements, what means that each value imposed on each element parameters corresponds to an expected value for real warehouse systems.

Next figure shows with more detail two racking paths and the correspondent racking vehicles during the simulation run.



Each racking vehicle was configured to have an average speed of 3 m/s and load and unload times of 12 seconds. In this figure are also visible a part of the conveyor network, some transfer tables and 4 picking points where human operators handle the material. Each of these picking points was configured with a buffer of 4 palletes and an average working time of 4.5 minutes. The action time of the transfer tables was established to be 3 seconds and the mean speed of each conveyor 0.2 m/s. The coloured rectangle elements represent the palletes on the layout.

Figure 11 represents a zoom view of the input and output zones connected to the production zone. Here one can see with more detail an inspection table and a rejection machine modelled as a rejection point.



In accordance with the client demands the warehouse overall input rate was established to be 35.6 pal/hour coming from the production zone and a direct output demand of 0.8 pal/hour in the output point. The overall picking points were defined to ensure a 60 pal/hour processing rate, half of which was directed to the output point after passing by the inspection table, and the remaining redirected to the warehouse storage.

Although the time spent on modelling this layout has represented two days of work, the main difficulties were related with the element representation on the simulator's view and the configuration of those elements, therefore depending the speed of modelling on the layout's complexity. Nevertheless, after finishing that task it is easy and fast to simulate the warehouse behaviour, and in this case it could be observed that simulation time was 30 times faster than the real time. This performance was achieved even with the simulator running in active representation mode (letting the user observe the movements of the objects on the computer screen), what means that higher speeds could be reached if only using background simulation.

Some of the results obtained with this simulation are presented in the figure 12, where it is plotted the amount of palettes *inputted* to the warehouse storage zone and the overall *output* back to the production zone along the 25 hours of simulation time versus the observed processing time.

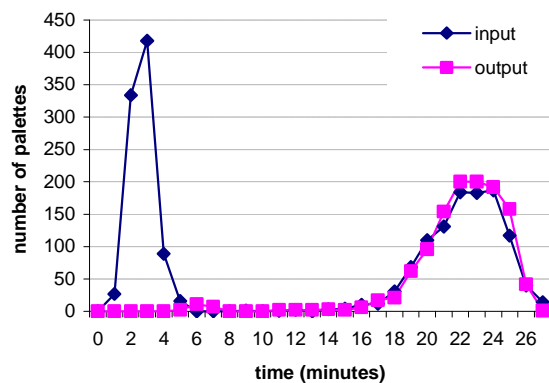


Fig.12

This graph shows a first spike for a large number of palettes with a processing time around 3 seconds, related with the palettes that directly enter in the system through the *input point* coming from the production zone. The second burst with a mean time processing near 23 minutes is due to the pallets redirected to the storage by the group of the six *picking points*. The direct *input rate* observed was 35.6 pal/hour.

What concerns the *output* process, the main contribution is due to those palettes coming from the group of the *picking points* in direction to the production zone (*output point*). The direct output demands were just 0.8 pal/hour, while the overall *output rate* observed was 47.4 pal/hour.

These results were obtained after 24.9 hours of simulation time in which 886 palettes have entered directly to the warehouse storage and 1180 palettes were moving out from the warehouse to the production area. The remaining palettes stayed circulating between the group of *picking points* and the warehouse storage zone. In these processes it have been observed a time occupation of each *racking vehicle* of about 60%.

6. Conclusions

After using this simulator for modelling some practical layouts, each of them proposed by the company EFACEC with industrial intents, one concludes that in fact this simulator allows a fast development of the model comparing with other multipurpose simulation software packages. This fact is mainly due to the inclusion of basic warehouse elements on this simulator which the user can access and configure in a very efficient way. Those elements, like *conveyors*, *racking vehicles*, *tables*, etc., are independently modelled and accessible to the user with a simple mouse CLICK. Thus, the representation of the layout is reduced to the creation of such elements followed by its positioning on the simulator's view. This method of building the model can therefore be compared with a game where one has to represent the warehouse as closer as possible to the real warehouse. Of course the time consumed on this process depends on the layout's complexity as well as on the precision with which the user wants to build the model. Nevertheless, the mean time expected to build such a model using old simulation resources was much longer than that achieved with this simulator, at least for the complexity of the layouts modelled to the company EFACEC.

Another advantage of this simulator is it's the simple way how the warehouse elements are configured, as each of them has associated a *property box* where parameters can easily be modified.

What concerns the kind of results achieved with this simulator, one can say that they allow a good characterisation of the layout performances, mainly what is related with input and output processing rates, time processing distribution for palettes demands, vehicle occupation time, and some other statistical results. These results, together with the possibility of observing the behaviour of certain elements directly on the screen while the simulation is running, what many times is difficult to predict in practice, showed that this simulator is a very useful tool directed to warehouse designing and analysis. This explains the good results obtained with this simulator by EFACEC on several practical case studies.

References:

- i J. M. Feliz Teixeira, António E. S. Carvalho Brito, "Introduction to a warehouse visual simulator", LASSIP, Faculty of Engineering of University of Porto, PORTUGAL, November 1997.
- ii António E.S. Carvalho Brito, "The Use of CAD Techniques in Configuring Visual Interactive Simulation Models: A New Approach for Warehouse Design", Ph.D. Thesis, Cranfield Institute of Technology, U.K., 1992.
- iii António E.S. Carvalho Brito, "The Use of Computer Aided Design Techniques in Configuring Visual Interactive Simulation Models for Warehouse Design", Journal of Decision Systems, Volume 1 - n° 2-3, Hermès, Paris, 1992
- iv António E. S. Carvalho Brito, J. M. Feliz Teixeira, "Warehouse visual simulation applied to a practical layout (Brasil)", LASSIP, Faculty of Engineering of University of Porto, PORTUGAL, July 1998.
- v António E. S. Carvalho Brito, J. M. Feliz Teixeira, "Warehouse visual simulation applied to a practical layout (Iran)", LASSIP, Faculty of Engineering of University of Porto, PORTUGAL, September 1998.