Natural Computer Components Oriented Simulation

Abdelhak Boubetra

University of Bordj Bou Arreridj, Computer Science Department, 34000, Algeria

Summary
Computer simulation utilizes computational models that allow the user to study real world situations without the requirement to invest in a real implementation. Without any doubt, the use of natural computer components to build a simulation model is of a great help to the simulation modeler since they offer a simplified and a structured approach to simulation modeling of various systems. This investigation is about such an issue and is supported with a case study using the ShowFlow simulation software to simulate a factory system.

Key words:
Simulation , modeling, natural computer component.

1. Introduction

Discrete event simulation is the act of experimenting with a model representing some real or virtual reality. This last reality can be a manufacturing system that can be defined as an automated of multi-functional machines that are interconnected by a material handling system. These systems are designed to combine the efficiency of mass-production line with the flexibility of a job shop to best suit the batch production. Simulation technology holds tremendous promise for reducing costs, improving quality, and shortening the time-to-market for manufactured goods. Unfortunately, this technology still remains largely underutilized by industry today.

This research suggests benefits to industry resulting from the widespread, pervasive implementation of manufacturing simulation technology. This suggestion is based on the use of natural components designed to imitate real components of the system under study. This imitation should concern descriptive knowledge of the manufacturing system in order to create high fidelity models of the manufacturing systems under study. How does one go about creating these types of models?

In this paper, we propose an approach to deal with that. The remainder of this paper is organized as follows. Section 2 reviews the Discrete Event Simulation (DES) process. In Section 3, we investigate the knowledge involved in DES. Section 4 discusses the simulation modelling of manufacturing systems. In Section 5, we present the simulation software on which our simulation approach is based. Section 6 illustrates the use of the approach in a manufacturing application. Finally, Section 7 concludes with future research directions

2. Overview on Discrete Event Simulation Process

Simulation, in the sense we use the word here, is a modeling process where a dynamic reality, either actual or projected, is imitated in terms of computer actions. It is accepted in the simulation literature that we are dealing with Discrete Event Simulation (DES) when the state changes in the system are represented by a series of discrete changes or events at specific instants. Shannon [1] characterises DES as entities having attributes and interact with activities under certain conditions creating events that change the state. From that, we understand that the knowledge we seek to learn, in DES, is belonging to a parallel world -many things happen at the same time.

The DES process, as shown in figure 1, starts by observing the behaviour of the relevant elements of the real world (a system) and creating representations of these real objects and their important characteristics. This step is referred to as the modeling phase. As a result of this phase a symbolic model is generated either by using some natural language or specific tools such as Petri nets [2] or activity cycles diagrams [3]. In the second phase, a general purpose programming languages or a specialised simulation language is used to transform the symbolic model to a suitable computer model which is called a simulation program. This last is designed to imitate the behaviour of the real world system in such a manner that, when experimenting with it in the third phase, its outputs could convince a knowledgeable person that they emanated from the world itself. Then, an analysis is done on the experimenting results in order to take adequate decisions dealing with best performances of the studied system.

Clearly this is not a trivial task and simulation projects require both skilled analysts and programmers with good access to a high quality computing service. In addition, every simulation should be viewed as a series of variables, representing the image of the system, and a set of rules

Manuscript received April 5, 2008
Manuscript revised April 20, 2008

characterizing the inter-related evolution of these variables in time.

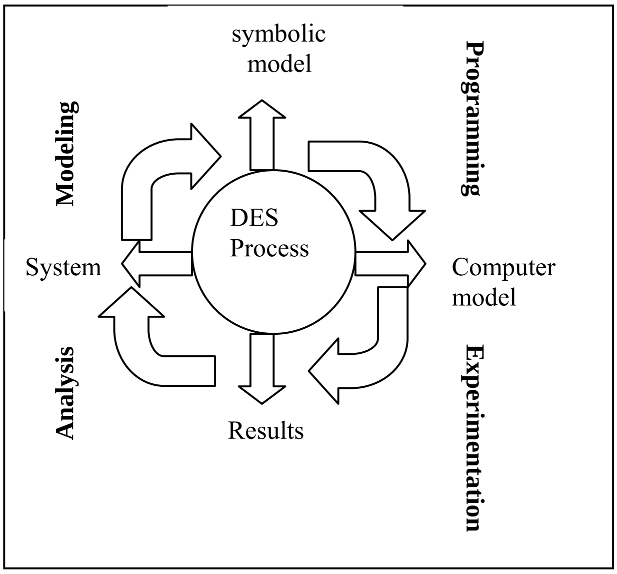

Fig. 1 The DES process.

Hence, some specific knowledge in DES is needed when creating the symbolic model and the programming model. On the other hand, the objective of experimenting with the programming model is to learn specific knowledge from DES results.

3. What is knowledge in DES

A simulation program contains knowledge about the system it simulates. Traditionally, this knowledge attempts to map the real world system in sufficient accuracy. This is why this knowledge is called descriptive. There are cases, however, where only a sample of the real world is available and therefore descriptive knowledge is not adequate. This knowledge is referred to as probabilistic.

3.1. The Descriptive Knowledge in DES

It is possible to distinguish three sorts of descriptive knowledge:

3.1.1. Description of the State of the Simulated System

We could further identify the internal description and the external one. The internal is the description of the system used by the analyst to describe the model in an adequate programming language, while the external one is the image of the system provided on the screen. This description changes with time.

The internal description is expressed by statements within the available simulation language. The external description is a set of visual computer objects.

3.1.2. Description of changes in the States of the Simulated System

These are the changes that are scheduled to happen to the simulated system at predictable points in the time. According to the three phase approach [4] terminology these changes are called ‘B’ activities for bound activities, and event routines to the two phase and the process approaches [5].

3.1.3. Description of the rules that govern the Simulated System

These are actions that are carried out or not, depending on the system state and in the three phase method they are called ‘C’ activities for conditional activities and in the two phase and the process approaches they are actions taken in the events and the processes, respectively, after the verification of certain conditions.

3.2. The Probabilistic Knowledge

In DES, probabilistic knowledge is what we are seeking to learn since we know in advance the activities and the processes and their rules and what we don’t know is when these things will happen. To deal with such situations, DES systems use statistical tools such pseudo random generators. Probabilistic knowledge, finally, attempts to provide insight not only to the facts but also to the reasons that rise to the facts.
Both descriptive and probabilistic knowledge are widely used in DES. Thus the state of the system, the events and the processes may be classified as descriptive knowledge, while probabilistic distributions which are used to describe durations of activities and processes are probabilistic knowledge.

4. Simulation Modeling of Manufacturing Systems

A discrete event simulation model of a manufacturing system can be usually considered as a number of connected components (e.g. machines connected by part routes) that receive and consume something in the input to produce something in the output.
In manufacturing systems, changes of state occur at discrete times and these are called events. Such systems are usually studied at these times only. Hence the manufacturing system is analyzed and studied in terms of events and actions at these events. These last can be

classified either time-driven 'occur at some known time in the future' or conditional 'occur when some conditions are satisfied' immediately after a time- driven event.
A variety of modelling approaches such as mathematical approaches [6,7,8], Markov chain approach [9], and queuing networks [10] exist for design and operational analysis of manufacturing systems. While these models provided insight into system behaviour, they introduced many restrictive assumptions making it difficult to model and evaluate manufacturing systems in dynamic situation.
In terms of modelling the DES knowledge involved in manufacturing systems, there is a need for a tool that offers a natural representation of this kind of systems, in which material (processes/transactions) flows through a set of machines and other resources. To meet this requirement, the simulation software Showflow [11] is used and discussed in this paper to illustrate such an approach.

5. Inside the ShowFlow Simulation Software

ShowFlow offers three fundamental entities to build a model. They are elements, products and clusters. Figure 2 illustrates some basic modelling elements to describe the state of the system. These are computer components representing machines, storage places and vehicles, or the resources needed to operate the equipment such as special tools. Products are the items that pass through the system, moving from element to element. A cluster is a group of elements and/or sub clusters that belong. ShowFlow offers means for the description of the changes in the states of the system according to the routing in the model. The probabilistic knowledge on the system is expressed throw an interactive system based on dialogue boxes to introduce the different parameters of the manufacturing system components. Here, we specify most of the behaviour of the model like the element parameters, the job parameters and the stage parameters.
Once the model is defined, ShowFlow provides a mechanism for running the simulation either in a step by step mode or in a continuous mode. After the simulation run has completed, the Standard Report may be displayed to view output statistics for the model from the Results menu bar.

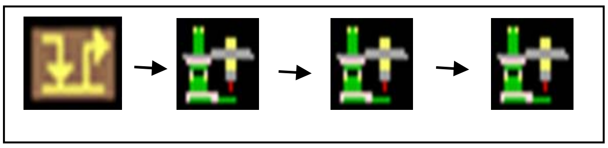

Fig. 2. Some Showflow elements.

6. A manufacturing system application

Let's consider a TV manufacturing system where a TV pass through two fundamentals operations the assembling then the testing as it is shown in figure 3. The manufacturing system line makes two different product types (TV A and TV B). There are three machines in the system, two assembling machines (Machine 1 and Machine 2) and a test station (Machine 3). Each TV type requires its own assembling machine, but the test station is used by both products. There are buffers between each station. In this study we want to get an idea on the capacity of the system, and also making possible improvements. The information rules that govern this system indicate that a batch of a TV A and a batch of a TV B enter the system every 1 minute. The batches vary between 1 and 10 (uniform). The time needed to assembly a TV on the automatic machines (Machine 1 and Machine 2) is 30 seconds per TV. Testing a TV takes 25 seconds on average, but as this is manual process the time varies greatly per individual product (exponential distribution). Machine 1 used for Product A runs without problems for an average of 15 minutes (exponential distribution). The time needed to repair this machine is exactly 1 minute (15 % down time). The buffers (buffer 2 and buffer 4) between machines have a capacity of 10 products. 5 % of the products is rejected and sent back to the assembling machines.
To model this problem with ShowFlow, we create a layout to this system by selecting the Model | Layout function from the main menu and position the elements in the following order:

[1: InOut, 2: Buffer, 3: Machine, 4: Buffer, 5: Machine, 6: Buffer, 7: InOut, 8: InOut, 9: Buffer, 10: Machine, 11: Buffer]

The layout of the model should look like figure 3.

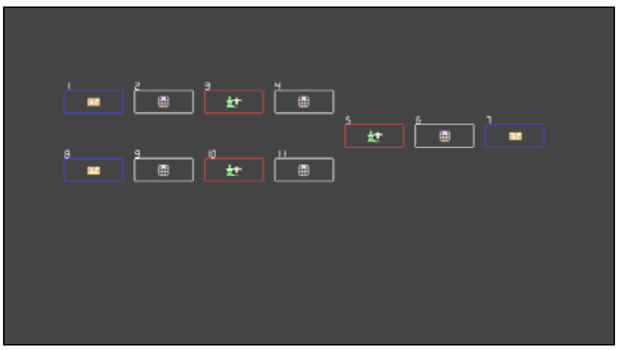

Fig. 3. A TV manufacturing system

To complete the rules that govern the system, we indicate the routing of the products through the elements. This is done, in ShowFlow, with the Model | Elements window and change the 'send to' fields description of the elements or by dragging and dropping lines between the elements in the case of using the Model | Layout | Edit | Mode | Routing function. For example, in the above proposed order : element 11 should send to element 5.
The next step is to enter some parameters in the model like the arrival of batches, the assembling time in the machines (Machine 1 and Machine 2) and the test time on the test station (Machine 3). Adding to that, we enter the down time for the first machine (Machine1) and we should mention that 5% of the products are sent back to the buffers just in front of the assembling machines.
Now the simulation model is ready to run through the command: Simulate |Single run | Start. During simulation we can change the visualisation mode to statistics by selecting it from the Animation menu in the simulation control window.
If the simulation has been interrupted, and we want to continue it later, we use the command: Simulate |Single Run | Continue . ShowFlow can write a history of events to disk, called the trace file. This can be viewed directly (Results|Trace), and is required for other reporting features. We can refer to this through the commands:

Trace report (Results | Trace)
Graphical analysis (Results | Element Graphs)
Status Diagram (Results | Status diagram)

In order to be able to make ideas on the simulation run, ShowFlow offers several ways to display different results. These last can be obtained by the Results | Element graphs command that give different graphs. For example the Queue graph of figure 4 shows queue length over time of element number 2, the Queue histogram of figure 5 shows how many time units the different queue lengths occurred in element number 4 and the Wait time histogram shows the distribution of waiting time per individual product.

7. Conclusion

In this paper, we have pointed out the involved knowledge in DES which is mainly of two sorts: descriptive and probabilistic. The first one is designated to build the simulation symbolic model and the simulation computer model. The second one is the knowledge that represents the purpose of any DES. These aspects were discussed through the ShowFlow simulation software objects.
This study is carried out to bridge the gap between virtual manufacturing technology and its counterpart in the real manufacturing world. This is achieved by the use of natural computer components to model the manufacturing systems components. We feel that such an approach provides a powerful modelling tool in manufacturing. Extensions can concentrate on the integration of tools to allow simulation data of manufacturing systems to be stored in spreadsheet or database form.

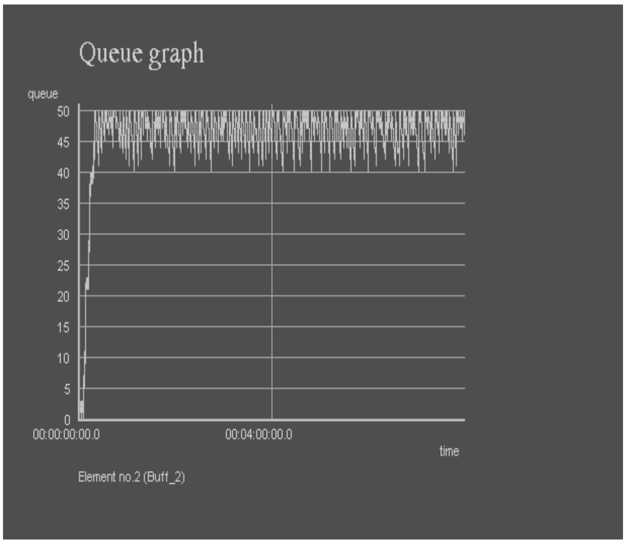

Fig. 4. Queue graph of element 2

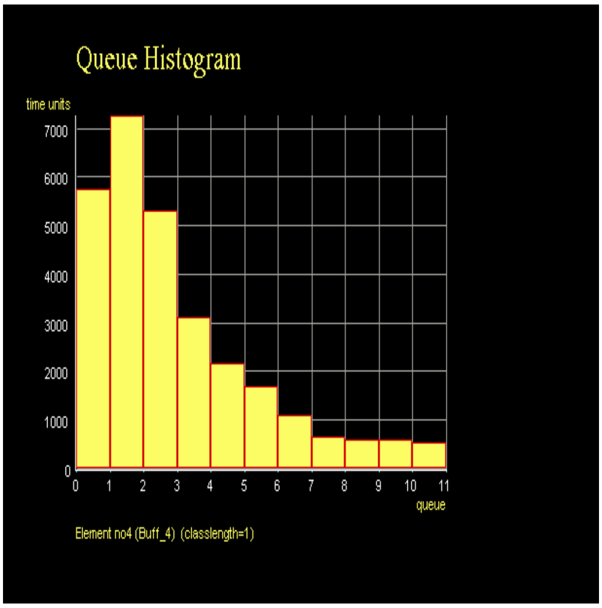

Fig. 5. Queue histogram of element 4

References

[1] H. Shannon, .Systems simulation. The Art and Science, ed. Prentice-Hall, 1975
[2] C. A. Petri, Kommunication mit automaten, Insitute fùr Instrumentelle MathematiK, Schriften des IMM, n°2, 1962.
[3] K. D. Tocher, The art of simulation, London, English Universities Press, 1962.
[4] M. Pidd, Computer simulation in Management Science, John Wilet & Sons Ltd., 1984.
[5] B. W.Hollocks, Forty years of discrete-event simulation- a personal reflection, J. of the Operational Research Society , 57, (2006), pp 1383– 1399.
[6] M. Berrada, K.E. Stecke, A branch and bound approach for machine load balancing in FMS, Management Science, vol. 32, (1986), pp. 1316-1335
[7] N. Kulatilaka, Valuing the flexibility of flexible manufacturing systems, IEEE Transactions in Engineering Management,, vol. 35, (1988), pp. 250-257
[8] K. Shanker, S. Rajamarthandan, Loading problems in FMS: part movement minimization, Proceeding of 3rd ORSA/TIMS conference on flexible manufacturing systems, Edited by K.E. Stecke and R. Suri,(1989), pp. 99-104
[9] K. Davis, W. Kennedy, Markovian modeling of manufacturing systems, International Journal of Production Research, vol. 25, No 3, (1987), pp. 337-351
[10] L. Zhuang, K.S. Hindi , Approximate mean value analyses for closed queuing network models for FMS with block-and-wait mechanism, Computers and industrial engineering, vol. 20, No 1, (1991), pp. 35-44
[11] www.showflow.co.com

Abdelhak Boubetra is a senior member at the Research Group of computer simulation in the computer Science department of the University of Bordj Bou Arreridj in Algeria (Algeria). He received a computer science engineer degree from the University of Constantine (Algeria), a Master Philosophy degree from the University of Lancaster (U.K) and a Doctorat degree from the University of Setif (Algeria) in computer science. His reasearch interests include data bases and software engineering.