Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

A SIMFACTORY TUTORIAL

Bruce Kleine
CACI, Inc.- Federal
12011 San Vicente Blvd.
Los Angeles, CA. 90049

1. INTRODUCTION

You have probably already heard at least a dozen speakers say that the use of simulation in manufacturing is increasing rapidly. Of course, they are all correct: witness the explosion in the manufacturing simulation software business. Less than ten years ago, there were probably no simulation tools designed specifically for manufacturing. Today, there are at least eight such tools - and they are all good. In fact, they are all good enough that a company would do better to choose one of these tools at random than to forsake simulation altogether.

However, these simulation tools are all very different. Each one has its own strengths and weaknesses. Each one is best suited to some subset of manufacturing simulation. Some tools, for example, might be best suited to simple problems requiring only rough analyses, while other tools might be better suited to complex problems requiring detailed analyses and a large programming staff.

Although any one of these tools is far superior to none at all, it is wise to select a simulation tool carefully. The purpose of my presentation is to introduce you to one of these simulation tools: SIMFACTORY. Perhaps I am not doing you a favor: like shopping for stereos, the more brands you see, with all their buttons and gadgets, the more confused you may get. But then, if you don't see all the brands, you might not get the right one for you.

By way of continuing my presentation, I will briefly outline what I intend to cover. I have divided the material into seven segments, most of them brief: (1) a synopsis of what SIMFACTORY is, (2) a history of SIMFACTORY, (3) the philosophy of SIMFACTORY, (4) an explanation of how SIMFACTORY works, (5) a discussion of the SIMFACTORY modelling primitives, (6) a summary of planned enhancements, and (7) a concluding note.

2. WHAT IS SIMFACTORY?

Like the other tools to which I have alluded, SIMFACTORY is a tool designed especially for simulating manufacturing operations. But that is about where the similarities end. However, my objective here is not to compare SIMFACTORY to other tools but rather to explain what SIMFACTORY is so that you can make the comparison. To do this, I will define some fundamental properties of manufacturing simulation tools and characterize SIMFACTORY in terms of these properties.

2.1. Host Language

The host language is the programming language in which the simulation tool was developed. Like everything else, different programming languages have different strengths and weaknesses and are designed for different applications. Hence, the choice of host language is an important one. The host language chosen for SIMFACTORY is SIMSCRIPT II.5. The history of SIMFACTORY (section 3) contains a brief discussion of SIMSCRIPT.

2.2. Approach

Factory simulation tools share the common goal of making simulation easier and thus more accessible to non-programmers. All of the tools now available accomplish this in one of two ways. One approach is to pre-process high level code into a lower level language. The other approach is to provide a general model that is data driven. SIMFACTORY takes the latter approach. I will discuss the reasons for that decision in a subsequent section.

2.3. Computer

SIMFACTORY runs on the IBM PC AT or IBM PC XT. There are plans to develop versions of SIMFACTORY for the VAX and IBM mainframes.

2.4. Graphics

The use of graphics in simulation has risen dramatically in the past few years. Graphics can play an important role in simulation. Many of the simulation tools provide some form of graphics. However, the spectrum is broad, covering everything from character graphics to detailed pictures of human figures. SIMFACTORY lies somewhere in between, providing a simple but meaningful animated display.

3. THE HISTORY OF SIMFACTORY

About two years ago, the Ralph M. Parsons engineering company, headquartered in Pasadena, California, called upon CACI to develop what later became the precursor to SIMFACTORY. At the time, Parsons was designing a metal fabrication plant. To help insure that the plant, as designed, would meet certain throughput requirements, Parsons decided to conduct a simulation of the plant.

Since the Ralph Parsons Company had other, similar work, CACI proposed to develop a general simulation tool that would not only solve the immediate problem, but would apply to other projects as well. Reinforced by the success of a prior teamup of the two companies and by the knowledge that CACI has been in the simulation business for over twenty years, the Ralph M. Parsons company accepted CACI'S proposal. The design of the tool was begun as a joint effort, combining industrial engineering expertise from Ralph M. Parsons with simulation expertise from CACI. The resulting tool was designed especially for evaluating production capacity and testing design alternatives.

The SIMSCRIPT II.5 programming language was chosen for implementing the model. This most powerful simulation language provides a "world view" that promotes natural, realistic modeling. The SIMSCRIPT primitives of entities, attributes, and sets form the flexible basis for modeling any physical system. The SIMSCRIPT primitives of events and processes round out the "world view" with a powerful capability for modeling time dependencies.

Using the model developed for Parsons as a foundation, CACI went on to develop its own manufacturing simulation tool. First, the basic model was enhanced and streamlined. Next, animation was added. Finally, a user friendly interface was developed to assist users in developing their models.

SIMFACTORY is now complete only in the sense that it is available and can be used to model manufacturing operations. But enhancements to SIMFACTORY have been and indeed are now underway. In fact, CACI has an aggressive enhancement plan for SIMFACTORY which I will discuss later. In this sense, SIMFACTORY will never really be finished.

4. THE PHILOSOPHY OF SIMFACTORY

The philosophy of SIMFACTORY, if a software product can be said to have a philosophy, is to provide flexibility and power and to be easy to use. Later, I will explain how this philosophy affected our decision to adopt the approach of the generalized simulator in preference to the approach of the pre-processor. But first, I would like to elaborate briefly on the concepts of flexibility, power, and ease of use as those concepts apply to SIMFACTORY.

4.1. Flexibility

Flexibility is achieved in SIMFACTORY by providing a set of primitives which can be combined to form models of factories. These "building blocks", or "modelling primitives", will be discussed in more detail later. The concept of modelling primitives is mentioned here because it is so fundamental to SIMFACTORY and because it is relevant to the discussion of philosophy. The important point here is that these primitives were tailored for manufacturing simulation and designed to maximize flexibility. It is hoped that modellers will use these primitives and combine them in imaginative ways - perhaps in ways not conceived of by the developers of SIMFACTORY.

4.2. Power

In the context of modelling, "power" may have different meanings. The kind of power that SIMFACTORY provides is leverage. By leverage, I mean the ability to quickly set up a model and to quickly test different hypotheses. In SIMFACTORY, this leverage is derived from its programming free environment.

4.3. Ease of Use

No matter how useful a product is, if it is not easy to use, it probably will not be used very much. This fact was responsible, more than any other single factor, for the decision to make SIMFACTORY a non-programming tool. Just the fact that SIMFACTORY requires no programming makes it fairly easy to use. The modeler needs only to enter data to describe his factory. However, to make SIMFACTORY even easier, a user interface was added. The user interface provides a friendly, forgiving environment which simplifies the task of data entry.

4.4. Modelling Approach

As stated earlier, the philosophy behind SIMFACTORY affected the decision to adopt the simulator modelling approach instead of the pre-processor approach. To understand this decision, it helps to understand the differences between the two approaches.

Language pre-processing systems are essentially very high level programming languages that provide the user with a great deal of leverage over lower level languages. That is, the user can develop a program by manipulating some very high level structures, usually via a graphical interface. The pre-processor then translates these high level structures into a lower level programming language. Such systems provide a good deal of flexibility but have several drawbacks. First, because they are programming languages, models must still be compiled, linked, and debugged - after the pre-processing step. Some of these steps are often "hidden" from the user. However, in more complicated models, these steps can become obtrusive.

Another drawback of language pre-processors is that modifications to the code are often required. Parameter studies can be made relatively easily, but experiments with factory layouts usually require some reprogramming.

A third important disadvantage of language pre-processors becomes evident when developing complicated models. The structures provided by pre-processor systems are adequate for modeling simple problems but cannot handle more complex ones. In these circumstances, the user must resort to coding special routines in the host language of the pre-processor, which in many cases, is FORTRAN. Indeed, most real-world problems are complicated enough to require programming in the low level host language. This is a serious drawback for end users who prefer not to do any programming.

The other approach to application-oriented modelling tools is the data-driven simulator. Properly designed, a simulator can provide both flexibility and ease of use. A good design will enable a simulator to experiment not only with parameters but with operational policies and factory layouts as well.

The one disadvantage of the simulator approach is that if a particular modeling capability is missing, users cannot generate their own by writing code. However, this objection can be overcome by providing new capabilities as they are needed.

The features most important to SIMFACTORY are flexibility, power, and ease of use. Based on this principle, the comparison of modelling approaches made the decision to develop a simulator fairly straightforward.

5. HOW SIMFACTORY WORKS

SIMFACTORY consists of three major components: the user interface, the consistency checking initialization, and the model. In addition, three kinds of output are generated: data echos, snapshots, and summary reports.

The SIMFACTORY User Interface is a menu driven input data manager with a graphical technique for entering factory layout information. The user interface has three important features: it simplifies data entry, it reduces data errors, and it helps the modeler to manage and organize his data.

After the modeler has described his factory with the user interface, he can initiate a simulation of that factory. Before SIMFACTORY will begin a simulation, however, it will first verify the consistency of the factory description provided. This verification process is performed by the initialization component. If any inconsistencies are detected during initialization, they are reported. The modeler can then correct the errors and try again.

Upon successful verification of a factory description, SIMFACTORY echos the input data and proceeds to the simulation. The data echos present data entered explicitly by the user along with some additional, implied information. This is useful both for verifying input data and for correlating summary reports with their corresponding inputs.

Concurrent with the simulation, SIMFACTORY provides an animated display of the factory at work. Using the snapshots, the simulation can be interrupted at any point to observe the status of the factory. The snapshots provide detailed information on the current status of processing stations, resources, queues, transporters, pending work orders, and future events. The snapshots also allow the user to control the animation and to obtain logic traces.

Summary reports are generated periodically during the simulation. These reports provide both periodic and final statistical information about system performance measures. Reports are available on processing station utilization, queue levels, transporter utilization, raw material consumption, throughput, and resource utilization. Any of these reports can be requested at any desired reporting frequency.

6. SIMFACTORY MODELLING PRIMITIVES

SIMFACTORY derives much of its flexibility from the premise that a factory can be reduced to three kinds of operations performed on one kind of object. The operations are movement, processing, and storage. The object is a part, or workpiece. SIMFACTORY provides nine primitives related to these concepts: processing stations, queues, transporters, process plans, maintenance actions, resources, the factory layout, the product description, and the production schedule. Each of these primitives is explained below.

6.1. Processing Stations

A processing station is where parts are processed. Processing can represent inspection operations as well as operations that physically modify parts. Both manual and machining operations can be modeled as processing stations.

In SIMFACTORY, a processing station is defined in terms of the set of operations that it can perform. Any number of operations can be defined for a processing station. Also, common operations may be performed by different processing stations. Operations are named by the modeler and are referenced by process plans (explained below) to indicate what steps are required to fabricate parts.

Each operation has a setup time, an efficiency, and may involve the use of resources. Setup time is the time required to set up the processing station for the operation. Efficiency is a measure of how well the processing station can perform the operation compared to other processing stations capable of performing the same operation. Resources can be defined for use in an operation. In SIMFACTORY, resources are thought of as catalysts to performing an operation and can represent things such as jigs and fixtures.

6.2. Queues

A queue is used for storing parts or resources. Queues also serve as the entry points for raw materials. Queues can be defined to have limited or infinite capacity.

6.3. Transporters

Transporters are used to move parts and resources around within the factory. Each transporter has a speed and a domain. The domain is the list of pickup and dropoff points served by the transporter. This is specified in the factory layout, described below.

6.4. Process Plans

A process plan defines how a subassembly is manufactured. A process plan consists of a set of one or more inputs, a sequence of operations, and one or more outputs. The operation sequence references the names of various operations performed by processing stations. A processing time is specified for each operation. If more than one input is specified, the first step in the operation sequence is assumed to be an assembly. If more than one output is specified, the last step in the operation sequence is assumed to be a disassembly.

Each output of a process plan can have its own reject probability. Three options exist for handling rejects: they can be ignored, scrapped or reworked. The default is to ignore rejects. In this case, the rejects are counted in the statistics but the actual rejected parts are never processed again. To scrap rejects means to model the disposal of the rejects. To rework means to recycle rejected parts for later use.

6.5. Interruptions

Interruptions on processing stations and transporters can be modeled in SIMFACTORY. Interruptions are defined in terms of interruption type, clock type, mean time between interruptions, and mean time to resume. Interruption type indicates whether the interruption is a priority interrupt or a passive interrupt. A priority interrupt can interrupt an operation in progress but a passive interrupt cannot. Clock type indicates whether the mean time between interruption is based on calendar time, i.e., total elapsed time, or operating time, i.e., actual productive time. Interruptions can be used to model maintenance, shifts, breaks, etc.

6.6. Resources

Resources are used by processing stations to perform operations. Resources can be used to model jigs, fixtures, containers, and even operators.

6.7. The Factory Layout

The factory layout serves two purposes: it identifies which of the pre-defined processing stations, queues, and transporters are to be included in the factory, and it defines how parts may flow through the factory. The modeler constructs a factory layout by placing the desired processing stations and queues on the factory layout screen and connecting them via transporters. Direct connections between stations can also be defined. A direct connection between stations permits parts to flow between the stations without the aid of a transporter.

In addition to defining these "physical connections" between stations, "logical connections" can also be defined. Logical connections further restrict the flow of parts within the factory to logically permitted paths.

6.8. The Product Description

The product description identifies which process plans are to be included to represent the bill of materials. Recall that a process plan defines how a subassembly is made. Many process plans may exist to represent different ways of manufacturing the same subassembly. The product description can thus reflect different manufacturing alternatives for a given product by selecting different process plans.

6.9. The Production Schedule

The product description, together with the process plans it references, define a bill of materials. This bill of materials contains raw materials, intermediate parts, and final products. The production schedule consists of a sequence of work orders for final products. Each work order identifies a final product and specifies a quantity and a start time. SIMFACTORY processes a work order by releasing the proper amount of each raw material required to make the specified quantity of final product at the specified start time.

7. PLANNED ENHANCEMENTS

Several enhancements are planned for SIMFACTORY. These enhancements fall into two categories: those that improve the interface and the aesthetics and those that provide new modelling capabilities. Some of the planned capability enhancements are summarized below.

Scheduling. Improved scheduling algorithms will permit modelling of MRP and JIT.

Conveyors. The transportation model will be augmented with conveyors for modelling continuous movement of parts.

Personnel. A capability for representing personnel will be added for more accurate modeling of operators.

Additional enhancements are planned for providing the user with more dynamic control over the simulation.

8. CONCLUSION

The field of manufacturing simulation has grown dramatically in recent years. There are now at least eight manufacturing oriented simulation tools that are fairly well known and fairly widely used. Each of these tools is good enough that using any one of them would be preferable to not doing any simulation at all. However, different tools are best suited to different applications. It is wise to carefully choose the right tool for the job.

In some ways, I have made your decision more difficult by introducing another tool for you to consider. Perhaps, a better understanding of what is available, however, will, in the long run, help to make your decision a better one. I hope that this presentation in some way contributes to making your choice of simulation tools the right choice for you.

AUTHOR'S BIOGRAPHY

Bruce Kleine is a simulation analyst at CACI. He received a B.A. in mathematics from UCLA in 1980. He has been involved in simulation for about ten years and is now the product manager for **SIMFACTORY.**

Bruce L. Kleine
CACI, INC.-Federal
Los Angeles, CA. 90049
(213)476-6511