Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

TEMPLATE BASED SIMULATORS: AN EXAMPLE FROM MANUFACTURING

Arne Thesen

Department of Industrial Engineering
University of Wisconsin-Madison
Madison, Wisconsin 53706

ABSTRACT

A template based simulator implements a generic model of a broad class of situations. To simulate a specific situation, the user specifies parameter values that describe the situation of interest. The utility of such a simulator depends on the usefulness of the generic model, the ease of use of the interface and the speed of the resulting simulation run. We show that a template based simulator can be used to analyze even fairly complex manufacturing situations involving multiple, unreliable work centers, multiple part routings with branching and multiple material handling systems.

1. INTRODUCTION

A number of simulation systems with friendly user interfaces are now available. Some of these are extensions of conventional simulation languages. Others are "simulators" that implement a model of a specific class of systems. Law and Haider (1989) identifies eight simulation languages and eleven manufacturing simulators with mouse driven graphical user interfaces. Simulators come in many forms. Some give the user access to an embedded simulation language (e.g. Witness, (Gilman and Billingham, 1989)), while others restrict user options to the entry of values for predefined parameters (e.g. Starcell, (Steudel and Park, 1987)). We refer to the first type as *language* based simulators and the second as *template* based simulators. Needless to say, a language based simulator has the potential of being able to model a wider range of situations than a template based one. On the other hand, a template based simulator has the potential of being easier to use than a language based simulator.

Template simulators are particularly useful when model features are easy to state but difficult to implement. For example, while it is as easy to specify an input buffer of limited capacity as it is to specify one with unlimited capacity, the "limited capacity" program is much more complex than the "unlimited capacity" program, and hence much more difficult to write. Among features suitable for template implementation are:

- Work centers with finite buffer sizes
- Multiple part routings
- State dependent routings
- Multiple part transfer systems
- Arbitrary distributions
- Unreliable machines
- Transporters
- Cranes
- Complex scheduling rules
- AGVS w/track contention
- Limited battery life
- Shared manpower

Most of these features can be modeled in currently available manufacturing simulators (most notably PROMOD), and many of these features can also be modeled in some of the existing template based simulators (most notably XCELL). To our knowledge, no template simulator is at this time capable of simulating all of the features listed above.

The usefulness of a template simulator depends on the appropriateness of the underlying model, the friendliness of the user interface, and the length of the resulting simulations. In this paper we show a template based simulator that can be used to analyze even fairly complex manufacturing situations involving multiple, unreliable work centers, multiple part routings with branching, multiple material handling systems and complex scheduling rules.

2. A GENERIC MODEL

A generic model that fits many manufacturing situations is given in Figure 1. We see that this model uses part routings to describe how parts from a number of different part families flow through a systems of unreliable work centers with limited input and output buffer capacity. Parts usually flow between work stations in the order that they are listed in the routing. However, an alternate path may occasionally be called for (for example when the part is bad). This is done by specifying the index of the alternate next step and the probability that this step is to be used. Parts may move between work centers by themselves (i.e. "free flow") or they may be moved by a truck belonging to one of several material handling systems.

Note that the model allows separate and independent templates for different model elements such as parts, routings, work centers, material handling systems, scheduling rules etc. Since each of these elements are well defined, and since the user is not asked to provide any further description of how these elements interact, model entry is now simply a task of entering the appropriate coefficient values for each model element. The systems data structure and simulation engine then integrates this information into a running simulation. Examples of situations that can be modeled using this model are given below.

2.1 Queuing Systems

Using the generic model given in Figure 1, the conventional barber shop model is described by specifying:

- one work center with one reliable machine,
- infinite capacity of the input buffer,
- one part family with an exponential arrival process,
- a one step routing specifying an exponential service time,
- no material handling,
- FCFS scheduling.

Extensions to this model such as:

- balking,
- other queuing disciplines,
- multiple and/or unreliable servers, and,
- customers with different priorities,

are easily described by specifying a finite input buffer, a different scheduling rule, part types with different priorities, additional servers and finite means for the breakdown and repair processes.

Further extensions such as those modeled by open queuing networks are described by representing each node in the network as a work center and by specifying one work center per node, and by using a part routing to describe the topology of the network. Closed queuing systems are similarly modeled. In this case the part arrival process is turned off after the desired number of parts have entered the system.

Work Centers
There are **W** work centers. Each work center:
- has an input buffer with fixed capacity,
- has one or more identical machines,
- has an output buffer with fixed capacity.

Each machine:
- fails (while busy) at random intervals with known distribution,
- has a random repair time with known distribution.

Routings
There are **R** part routings. Each routing specify a partially ordered sequence of steps. Each step defines the following items:
- the index of the work center performing the operation, (work center zero terminates good parts, work center 99 terminates bad parts),
- the probability distribution describing the duration of the operation,
- the index of an alternate next step,
- the probability that the alternate step is next.

Parts Families
There are **P** part families. Parts from a given family family:
- are processed according to a specified routing,
- have a specified priority *P*,
- are generated according to one of the following patterns:
  1. random intervals from a specified distribution,
  2. immediate replacement of parts in first input buffer,
- arrive until a given count of parts have entered the system.

Flow of parts between work centers.
Parts normally move without delay between work centers as specified in their routings. A material handling system (see below) is used if the appropriate buffers are assigned to this system.

Material Handling Systems (MHS)
A material handling system is represented as directed graph and a collection of carts. Each cart:
- Has room for one part
- Has a fixed speed.

Each network node:
- may serve as a transfer point between workstation buffers and the MHS system,
- can hold a limited number of carts.

Each network link:
- represent the travel path between two nodes,
- has a given distance,
- allows flow in one direction only,
- has room for a limited number of carts.

Scheduling
Parts are processed in priority order. Parts with equal priorities are processed according to sequencing rules such as the following:
1. First come first served
2. Longest Processing time
3. Shortest processing time

A decision to move a cart may be made under four conditions:
1. One part to be moved, one unit available
2. One part to be moved, many units available
3. Many parts to be moved, one unit available
4. Many parts to be moved, many units available

Different rules may be used to identify the part/unit combination to be moved for each of these conditions.

Figure 1. A Generic Template Model

2.2 Material Handling

Unless otherwise specified, parts move without delay between workstations as specified in their routings. Delays due to material handling are specified simply by assigning the relevant input and output buffers to a material handling system. A directed graph is then used to model this material handling systems. The nodes in the graph serve as transit points and as loading and unloading stations. The links represents the travel paths of the system's carts. Both nodes and links may have limited capacities. Hence physical restrictions such as track contention (one cart per link) and limited loading and unloading space are easily represented.

2.3 Implementation

The generic model proposed in Figure 1 was used as a basis for a mouse and menu-bar driven TBS running on an IBM/PC or equivalent. Called **TBS**, the system was first developed to support research activities in the area of scheduling of material handling equipment. As our experience with, and confidence in, the system has grown, the use of **TBS** has expanded, and **TBS** is now also used in teaching and in a limited number of industrial applications. Further information about **TBS** is given in Thesen (1990), and Thesen and Travis (1991).

3. EXAMPLE: AN INSPECT/REPAIR STATION WITH MATERIAL HANDLING

Consider the eight station Inspect/Repair system shown in Figure 2. Centers 1 and 8 are automated loading and unloading centers. Centers 2 and 6 are automated inspection stations. The actual repair is performed by two expert repairmen at center 4. Centers 3, 5 and 7 perform automated disassembly, assembly and packaging tasks. Most centers are quite reliable, however the disassembly center occasionally goes down.

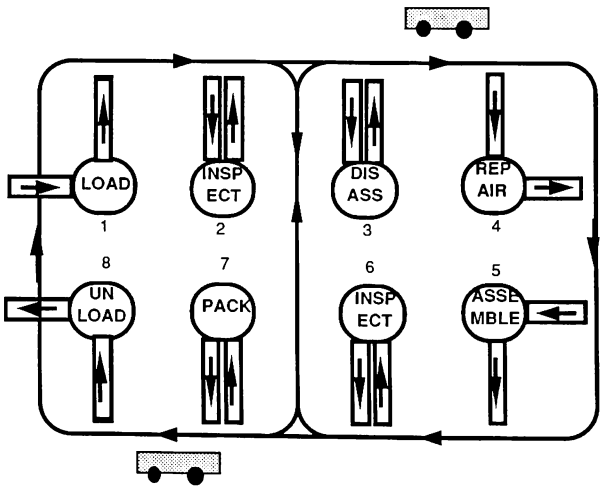

Figure 2. Inspect/Repair with Eight Work Centers and Two Trucks

During the initial inspection it is determined that twenty percent of all parts need to be repaired. Ninety percent of these are repaired on their first pass through the repair loop. Ten percent of the parts making a second pass through the repair loop are found to still be bad. These parts are discarded. A graph showing this pattern of flow of parts through the system is shown in Figure 3.

All transfer of parts between centers is performed by two automated guided vehicles or AGVs. The AGVs are not allowed to pass each other, and only one AGV is allowed on any track segment

and loading/unloading station. A directed graph showing the structure of the AGV network is shown in Figure 4.

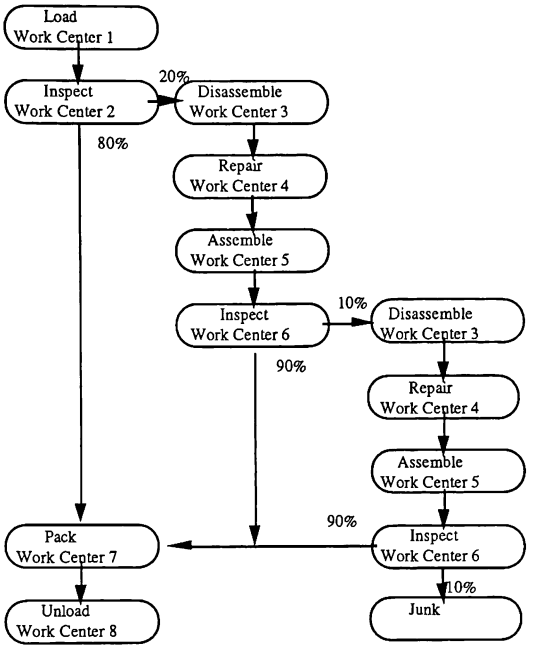

Figure 3. The Pattern of the Flow of Parts Through the Inspect/Repair System

All transfer of parts between centers is performed by two automated guided vehicles or AGVs. The AGVs are not allowed to pass each other, and only one AGV is allowed on any track segment and loading/unloading station. A directed graph showing the structure of the AGV network is shown in Figure 4.

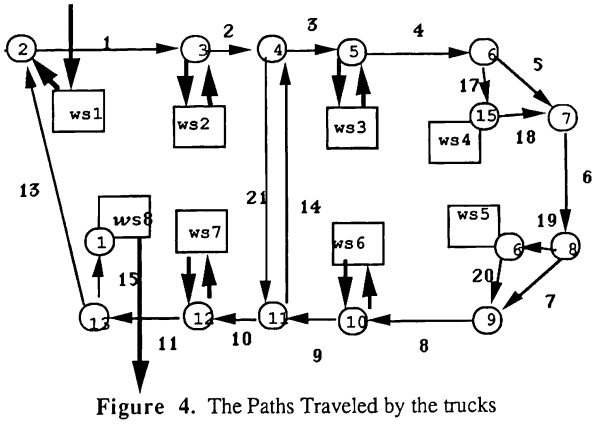

Figure 4. The Paths Traveled by the trucks

To build the simulation model, we enter the following information into TBS by filling in the appropriate cells in templates describing the different model elements.

Work centers: Number of machines at work center, capacities of input and output buffers, failure and repair rates for machines, priority rules for processing parts at work center (Figure 5).

Parts: Interarrival time distribution for each type of part, index of the routing, number of arrivals to be simulated.

Part routings: Processing times required by each type of part at each center, sequence of centers that each type of part must visit (Figure 6, note that bad parts are sent to work center 99 and that good parts are absorbed by work center 0).

Index	Work Center Name	Capacities			Reliability	
		Input Buffer	Machns	Output Buffer	MTTF (Hrs)	MRT (Hrs)
1	Load	4	1	4	999,999	0
2	Inspect	4	1	4	999,999	0
3	Disassemble	4	1	4	2.5	0.25
4	Repair	4	2	4	999,999	0
5	Assemble	4	1	4	999,999	0
6	Inspect	4	1	4	999,999	0
7	Pack	4	1	4	999,999	0
8	Unload	4	1	4	999,999	0

Figure 5. The Work Centers Template

Step No	Work Center Index	Distribution	Mean Time	Second Parameter	Prob	Alternate Step
1	1	Constant	1	0	0	0
2	2	Constant	1	0	0.8	11
3	3	Erlang	5	3	0	0
4	4	Exponent	18	0	0	0
5	5	Erlang	4	3	0	0
6	6	Erlang	1	6	0.9	11
7	3	Erlang	5	3	0	0
8	4	Erlang	18	0	0	0
9	5	Erlang	4	3	0	0
10	6	Erlang	1	6	0.1	99
11	7	Constant	1	0	0	0

Figure 6. The Routing Template

Material Handling System: Speed and initial position of trucks, location of each portion of the path or track, length and capacity of each portion of track (Figure 7), scheduling rules to determine which truck performs which tasks in what order.

Link	From Node	To Node	Distance (ft)	Capacity (carts)	Transport System
1	2	3	10	1	1
2	3	4	5	1	1
3	4	5	5	1	1
4	5	6	10	1	1
5	6	7	10	1	1

Figure 7. The Network Links Template the Mouse is Used to Expand the Window to Reveal the Rest of The Template

After the data entry is completed, the simulation can be run with or without automatic animation. During the run, a periodically updated status report report such as the one shown in Figure 8 is produced. Numerous other reports are also available. These show the utilization of work centers and vehicles as well as the production rate for parts and the utilization of the transportation network. The part flow report is shown in Figure 9.

Message.= Run Terminated by clock		Time (min)=	4800
File....=agvs.tbs		Day.....=	10
Wrk Ctrs= 8 Parts Arrived=	3098	Stop time.=	4800
Machines= 9 Parts Refused=	54	Clear time=	0
Prt Tpes= 1 Good Parts Md=	2995	Tick intrv=	100
Routings= 1 Trashed Parts=	8	Hours/Day.=	8

Figure 8. Abbreviated Status Report

Part Flow at Time = 4800.00

From	To Outside		To Work Center				
	Good	Bad	1	2	3	4	5
Initial	-	-	0	0	0	0	0
Outside	54		3044	0	0	0	0
ws1	0	0	0	3040	0	0	0
ws2	0	0	0	0	610	0	0
ws3	0	0	0	0	0	663	0
ws4	0	0	0	0	0	0	654
ws5	0	0	0	0	0	0	0
ws6	0	8	0	0	0	0	0
ws7	0	0	0	0	0	0	0
ws8	2995	0	0	0	0	0	0
now			4	7	7	9	8

Figure 9. A Section of the Part Flow Report

Notice that 2995 good parts were produced in a total of 4800 simulated minutes. This information helps the user assess whether the proposed system is capable of handling the desired throughput.

4. DISCUSSION

Our first challenge: to design a generic model sufficiently flexible to be useful and yet sufficiently well defined to serve as a basis for a template based simulator. Certainly are a large number of potential features have been omitted from the generic model (storages for example), yet our experience to date indicates that the general structure of the model indeed is quite flexible and that many features currently omitted from the model can be added without a significant increase in apparent model and interface complexity. One such feature is the use of machines that can process more than one part at a time.

Our second challenge: to develop a data structure that allows independent specifications of the different modeling elements while still facilitating the immediate execution of the resulting simulation model. Our success in reaching this goal is mainly due to the use of a combined next-event and activity scan world view and the use of part routings to control the movement of parts. Each part knows where it is and where it is going. Decisions about how to get there are therefore easily made at the time of the move.

Our third challenge: to find a cost effective way to implement a mouse driven user interface using pull-down menus and spread-sheet-like data entry screens. The time and budget constraints on this project are such that a commercial user interface was indicated. A number of such packages are available. Based on a comparison of features, royalty and distribution restrictions as well as portability between operating systems, the C-Scape system was selected (Cooke et. al., 1989). The use of C-Scape substantially reduces program development time while at the same time allowing for a great deal of flexibility in experimenting with different menu designs and report formats. Perhaps the only disappointment in using C-Scape is the rather long time (a few seconds) required to load data from the simulator's data structure into the editors' data structure. However this is a minor nuisance and not a serious problem.

TBS is capable of simulating a wide range of material handling systems. However, simulations of models using AGV systems with restricted access to tracks occasionally terminate early as deadlock situations occurs. The present version of TBS has no automatic way of resolving such deadlocks. The problem is often overcome by specifying better scheduling rules and/or by relaxing capacity restrictions on key network nodes.

Finally, the development of a 25,000 line program such as **TBS** is a fairly difficult and error prone task. To ensure that the present version is reasonably bug free, it has been used to replicate several dozen published simulations. A similar number of simulations for which analytic results were known have also been simulated. **TBS** always gave the expected result. We found no bugs in C-Scape, but we found several serious bugs in our QuickC 2.0 compiler. These appears to be eliminated in the new 6.0 implementation.

5. CONCLUSION

Our goal was to show that a template based simulator based on a well designed generic model can be used to analyze the performance of a broad class of systems. We feel that this goal has been met. Three factors contributed to this success. First, we used a generic model to describe moving resources, stationary resources and parts in independent templates. Second, we used a commercial interface program to implement pull down mouse and screen editors. Finally, our simulation "engine" is designed to incorporate arbitrary material handling systems and part routings.

TBS is currently being used in our teaching and research as well as in a limited number of industrial applications. Design changes and refinements are continually made as different needs are encountered. While **TBS** is not in the public domain, we are more than happy to make **TBS** available for use by fellow simulators.

BIBLIOGRAPHY

Conway, R. and W. Maxwell (1987), "Modeling Ascyncronous Material Handling in XCELL+," *Proceedings of the 1987 Winter Simulation Conference,* A. Thesen, H. Grant and W.David Kelton , Eds. IEEE, Piscataway, NJ, 202-206.

Cooke, DeSantis Peck and Whitman (1989), *C-Scape Interface Management System, Function Reference,* Oakland Group Inc, Cambridge, MA 12989.

Gilman, Andrew R. and Claire Billingham (1989), "A Tutorial on SEE WHY and WITNESS," *Proceedings of the 1989 Winter Simulation Conference*, P. Heidelberger et.al., Eds. IEEE, Piscataway, NJ,

Law, Averil M. and S. Wali Haider (1989), "Selecting Simulation Software for Manufacturing Applications: Practical Guidelines & Software Survey," *Industrial Engineering,* May 1989, 33-46.

Steudel, Harold J. and Tahoe Park (1987), "STAR*CELL: A Flexible Manufacturing Cell Simulator," *Proceedings of the 1987 Winter Simulation Conference,* A. Thesen, H. Grant and W. David Kelton , Eds. IEEE, Piscataway, NJ.

Thesen, Arne and Laurel E. Travis (1991), *Simulation For Decision Making,* West Publishing, St.Paul, MN.

Thesen, Arne (1990), "TBS: A Template Based Simulator," *Department of Industrial Engineering Technical Report 90-2.*