Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

SIMULATION MODELS OF MATERIAL DELIVERY SYSTEMS

Vadivelu Jeyabalan
Norman C. Otto

Manufacturing Systems Department
Research Staff, Ford Motor Company
Suite 1100, 23400 Michigan Avenue
Dearborn, MI. 48124

ABSTRACT

One important aspect of JIT manufacturing is the timely delivery of material from its arrival point at the facility to the point of processing. In this paper, a simulation modeling method which captures the essentials of this process is described. This method was developed through several applications to actual systems. These applications cannot be modeled by simple simulation modeling constructs. Specialized modeling logic and constructs for truck arrival and unloading, moving material to an intermediate storage area, reloading the truck with empty containers, and feeding the production lines are presented. Several applications are discussed and used to demonstrate the types of issues and problems these models are capable of addressing.

KEY WORDS: JIT, Material Handling, Material Delivery Systems

1 INTRODUCTION

Just In Time (JIT) manufacturing is an area which has received a great deal of attention in recent years. While JIT is a concept that has many facets, the efficient and timely movement of material is certainly one key element. From a long range perspective, this material movement has four main parts; (1) the transportation of raw materials and purchased parts from the supplier to the manufacturing facility, (2) delivery of this material from the point where it enters the plant to the point of processing, (3) the movement through all the production steps required to produce finished goods, and (4) the timely distribution of the final product to the customer. While items (1), (3), and (4) have been the focus of many studies, the demands of JIT on the internal material delivery system have not received as much attention. In this paper, we will present a simulation modeling approach that can be used to design and evaluate the performance of these material delivery systems. This approach was developed through modeling a number of different real life systems and, thus, it accounts for most of the major effects found in these systems.

2 SYSTEM OVERVIEW

Figure 1 shows the basic system addressed by these models. Incoming material arrives by truck according to some specified schedule (which may be varied). These trucks may be dedicated to a single type of part or material or they may contain a number of different part types picked up from various suppliers (on a so called "milk run"). Although this discussion assumes that the incoming material arrives by truck, rail transport could be treated in an analogous fashion.

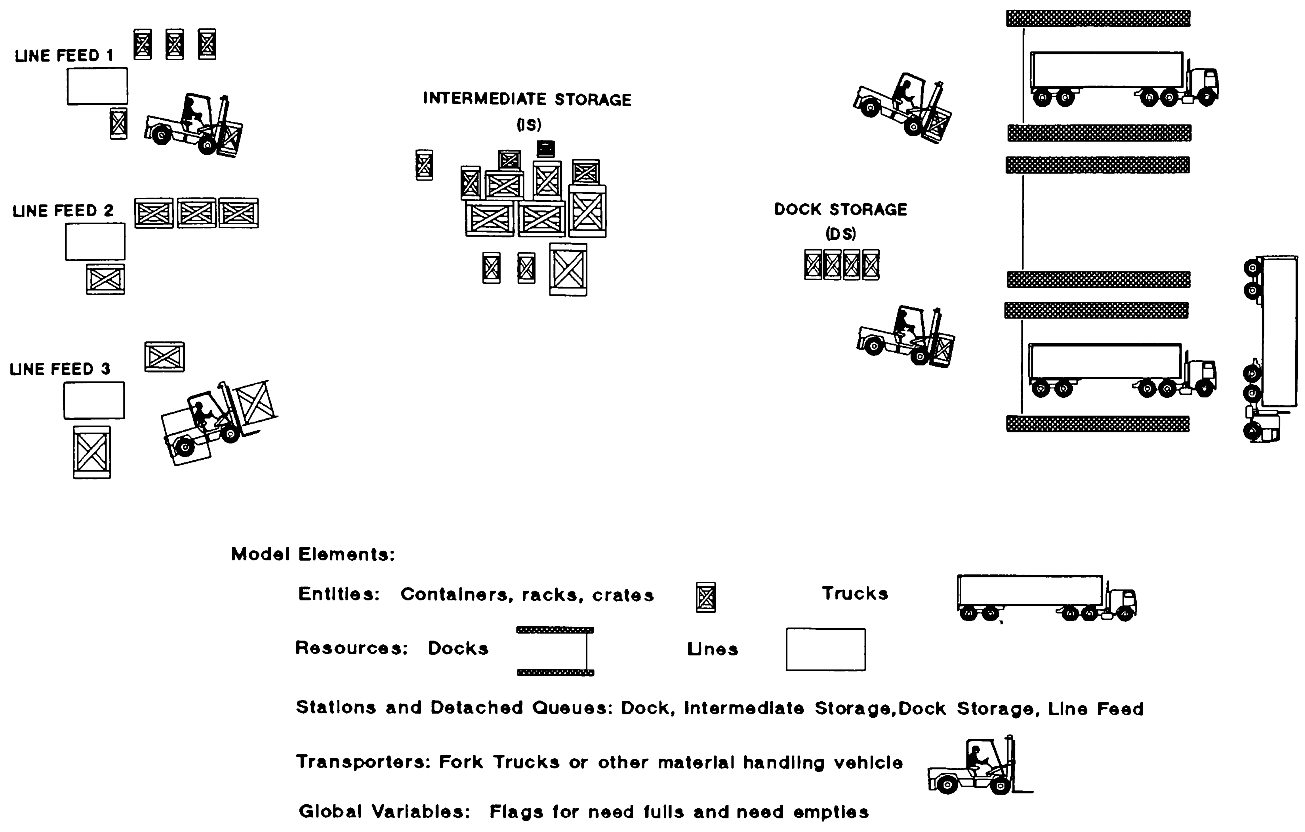
LINE FEED 1
LINE FEED 2
LINE FEED 3
INTERMEDIATE STORAGE
(IS)
DOCK STORAGE
(DS)
Model Elements:
Entities: Containers, racks, crates
Trucks
Resources: Docks
Lines
Stations and Detached Queues: Dock, Intermediate Storage,Dock Storage, Line Feed
Transporters: Fork Trucks or other material handling vehicle
Global Variables: Flags for need fulls and need empties

Figure 1. System Overview and Basic Model Elements

The trucks are unloaded at docks using fork lifts and the containers holding the parts ("fulls)" transported to an intermediate storage area. From here, each type of part is taken, as needed, to a specific line feed location where they enter the production process. Because most JIT systems employ reusable containers, the delivery system must also handle "empties". When all the parts in a container have been fed to the line, the empties are moved back to intermediate storage. From here, they are transported to a dock storage area and, at the appropriate time, loaded back on the trailer truck for return to the suppliers. Very often, to avoid "deadheading", the delivery of full containers is linked to a pickup of an empty, both between the dock and intermediate storage and between intermediate storage and the line feed locations.

2.1 Model Elements

As described below, the models used to capture the behavior of these systems were a combination of SIMAN (V3.5) and customized FORTRAN code. At the bottom of Figure 1, the basic elements of these models are shown. These will next be discussed individually.

2.1.1 Entities

The transportable unit loads in which material is moved and stored are modeled as entities. These are referred to, synonymously, as containers, racks, or crates. They are generated upon truck arrival and carry as attributes information about the type of part, the number of parts, and whether the crate is full or empty. Part type identity is retained for as long as these entities are in the system.

In cases where more than one container is transported at the same time, JOIN blocks may have to be used. Trucks are also modeled as entities with an attribute to identify truck "type". Truck type defines the number of loads and type of material contained in each truck. Both rear and side unloading trucks may be accommodated.

2.1.2 Resources

The docks are indexed resources that must be seized by a truck before it can be unloaded and are not released until the truck is reloaded with empties. If there are physically separate or dedicated dock areas, these may be modeled as specific resources (DOCK1, DOCK2, etc.). At each line feed location, we also use an indexed resource (LINE) to represent the part processing system. A container entity seizes the appropriate LINE entity when the material in that container is being fed to the line. The LINE resource is held for the time necessary to unload the container, after which it is released and an empty crate created.

2.1.3 Stations

Stations are used to represent the locations to which material is transported. Thus, we have station blocks at the dock, at both the intermediate and dock storage areas, and at each of the line feed locations.

2.1.4 Detached Queues

Detached queues are used to model the storage locations where container entities await movement. These material delivery systems are not continuous flow systems and, thus, detached queues provide holding areas at each station location. The model logic controls when entities are inserted and removed from these queues.

2.1.5 Transporters

The models presented here can be used for any type of manually operated material handling equipment and fixed material handling automation (like conveyors). AGVS and ASRS are not covered. In this paper, we will focus on fork trucks as the material handling method. However, other, more complicated systems, like tow trains with dollies have also been addressed. These will be discussed at another time.

2.1.6 Global Variables

While global variables are used for many purposes, one use deserves special mention. This is the use of these variables as flags which control where material will be delivered. For each part type, a pair of global variables is employed. The first indicates the number of unfillable requests for full containers that exists. A request for a full container at the line is unfillable when no containers of that type are present in intermediate storage. When this condition exists, any incoming trucks will have their containers taken directly to the line rather than intermediate storage. Similarly, the second flag gives the number of unfillable requests that exist for empties at the dock.

Truck Arrival Pattern
Truck Composition (# loads, part types)
Container Size and Parts/Container
Full/Empty Return Ratio
# of Docks
Dock Location
Unloading Method (End or Side)
Dock Selection and Constraints
Storage Capacities, intermediate and Dock
Storage Area Locations
# Forks
Fork Truck Speed, Breakdowns, Maintenance
Returnable or Disposable Containers
Work Rules
Line Feed Locations
Line Feed Buffer Capacities
Processing rates
Process Downtime (Scheduled & Unscheduled)

Figure 2. Model Input Data

2.1.7 Input Data

Figure 2 shows the various types of data required to run the models. These should require no further explanation other than to note that "What If?" studies can be run with these parameters to determine their effect on system performance.

3 MODELING

In this section, we will present the modeling approach used to combine the basic model elements discussed above into a simulation of the system behavior. While some small modifications have usually been required to fit individual applications, the general approach has proven to be quite generic. This section has been subdivided into four parts (truck arrival, truck unloading, truck loading, and line feeding), each of which will be discussed individually. Rather than show actual code, logical flow diagrams have been constructed to aid in this discussion.

3.1 Truck Arrival

Truck arrival encompasses the creation (at the appropriate times) of the incoming truck entities, the selection and acquisition of a dock, and the creation of the individual container entities contained in the truck. This process is modeled as shown in the top portion of Figure 3. We note that several Fortran event subroutines (as indicated by the {}) are used. This is largely because in most actual manufacturing facilities, the incoming material flow is extremely complex and requires a great deal of data to describe. Much of this description is more easily handled with Fortran than with SIMAN. For truck creation, we use a truck generation entity which, at the time of a truck arrival clones a truck entity and, in the NXTTRK event, reads the type and arrival time of the next truck from a file. This generation entity then delays for the time to the next arrival and then repeats the process. With this procedure, very complex arrival patterns can be generated "off line" (outside of the simulation model) with a simple preprocessing program. The truck entity then calls GETDOC to select which dock this truck will use and to assign a dock unloading priority.

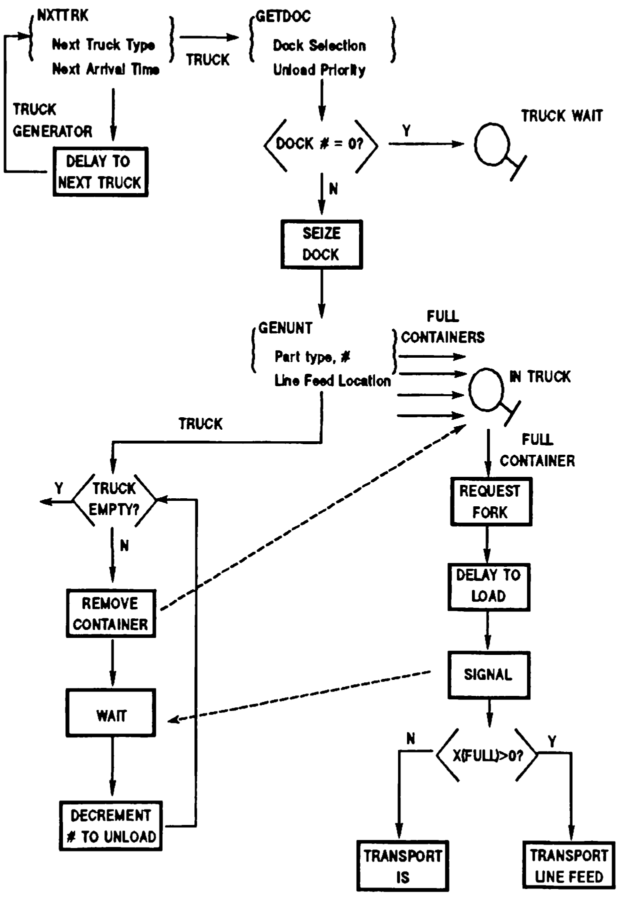

Figure 3.Truck Arrival and Truck Unloading

Customized rules other than the usual SIMAN resource allocation rules are often needed for dock selection. If no docks are available, the returning truck entity is sent to a detached queue (TRUCK WAIT) to await a dock to be freed. Otherwise the selected dock is seized. At this point, the truck composition data (# and types of loads) is read from a file and the individual container entities in the truck are created. Very often the truck composition data base may be extremely large and provisions for reading the relevant data from, for example, an ASCII dump of a spreadsheet are required. This is done in the Fortran routine GENUNT. For each container entity created, information as to the type of part, number of parts in a container, and line feed location may be stored in attributes. In addition, if containers are transported in multiple units, the individual container entities may be combined at this point. After creation, the containers are placed in a detached queue associated with the dock.

3.2 Truck Unloading

The process of truck unloading is illustrated in the lower half of Figure 3. In this section of the model, the truck entity acts to control its own unloading. An attribute is used to track the number of loads remaining in the truck. If loads remain, this entity directs the removal of a container entity from the IN TRUCK detached queue and sends this entity to a request block where a request for a FORK is made. The dock unload priority is passed on to the container entity. The truck entity waits for that fork to arrive, to load and to leave the truck. The truck is signaled out of the wait state by the container entity and the number of loads in the truck decremented. Such a control scheme is necessary to insure that only one fork truck is allowed in the truck at any one time. If the container entities were sent simultaneously to request blocks, then they would, if multiple fork resources were available, be unloaded simultaneously. For a truck that is endloaded, all the full containers must be removed before any empties are reloaded. Thus, we repeat this process until the truck is empty. The truck entity then attempts to reload the truck with empties (see below). The full containers, on the other hand, are transported by the fork to either intermediate storage or, if the full needed flag for that part is up, to the line feed location.

As mentioned earlier, we would like, for reasons of efficiency, to couple, or link, the delivery of a full container to intermediate storage with the return of an empty container to dock storage. The means by which this linking is accomplished is illustrated in Figure 4. Upon arrival of the fork truck at the intermediate storage station, a delay for drop

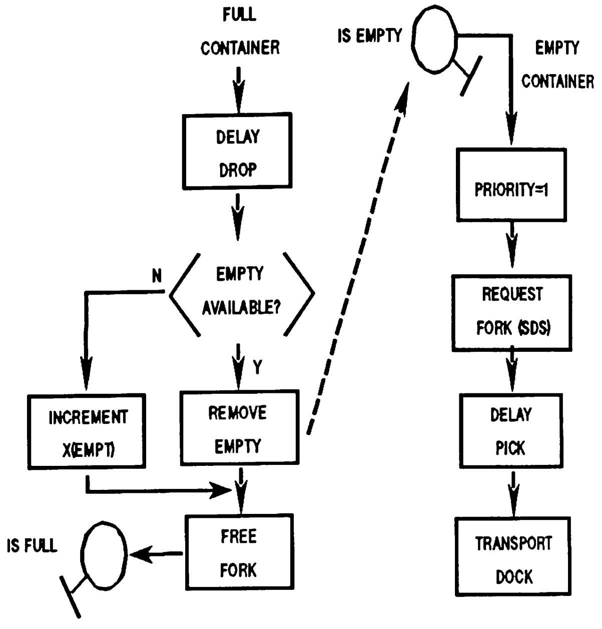

Figure 4. Intermediate Storage

is made followed by a check to see if any empty containers of the same type are available. This is done by searching the IS EMPTY detached queue for a match to the incoming part type. If one is found, the empty container entity is removed and sent to a fork request block. The fork that delivered the full is then freed and the full container placed in the IS FULL detached queue. This fork would then be free to pick up the empty and return it to the dock. Note, however, there is no guaranty that this will happen, the fork could leave to fulfill another request. To minimize this possibility, we assign the highest priority to the empty return request and use an SDS (smallest distance to station) selection rule. If no empty is found, the need empty flag for this part is incremented.

3.3 Truck Loading

Loading an empty truck works, in many ways, like truck unloading. Again, the truck entity acts to control the process initiating the transport of empties to the truck. However, as shown in Figure 5, there are some differences between loading and unloading.

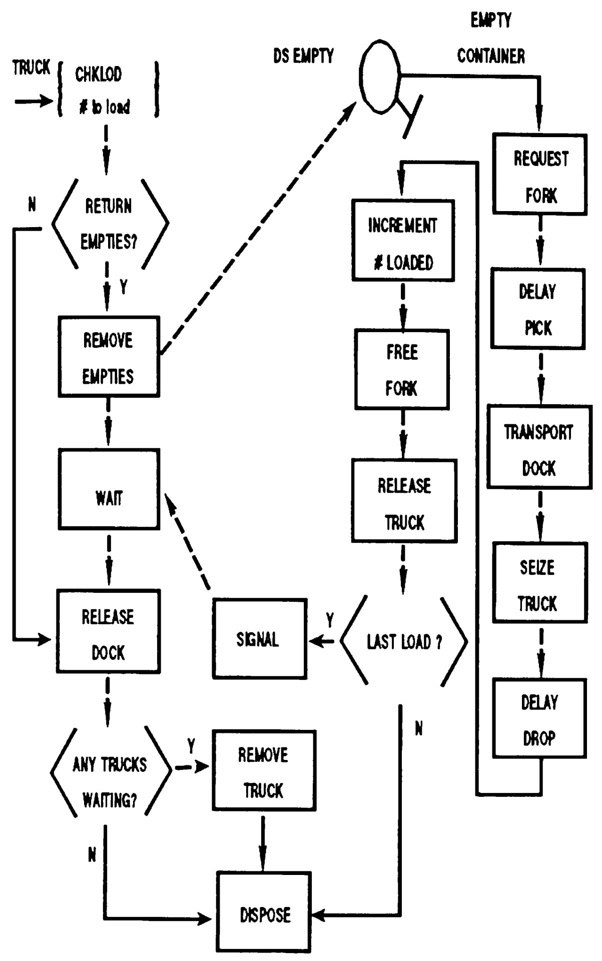

Figure 5. Truck Loading with Empties

First of all, not every truck is used to return empties. Trucks generally have a greater capacity for empties (which, of course, weigh less) than for fulls. Efficiency considerations dictate that this capacity be used as much as possible and, thus, empties are returned only periodically. The empty capacity relative to the capacity for full containers is called the return ratio. The determination of whether reload a truck with empties is made through a call, by the truck entity, to the CHKLOD routine upon the completion of unloading. For each truck type, a running count is maintained (with global variables) of the difference between the number of fulls delivered and the number of empties returned. When this difference equals or exceeds the truck capacity for empties, a return is made. Empty container entities are removed from the DS EMPTY detached queue and sent to a request queue for transportation to the dock. Dock number and priority are transferred from the truck entity to the empty container entities. Unlike unloading, all the entities are removed from the queue at once rather than one at a time. This is because the dock storage area may be relatively far from the actual dock and waiting for one empty to be loaded before requesting transportation of the next would be inefficient. Although not shown in Figure 5, the IS EMPTY queue is also searched if there are not enough empties in the dock storage area. If this fails to produce the desired number of empties, the trucks will leave with however many are available. Trucks will not wait for additional empties to be generated. After removing the empty container entity, trucks must wait for the last load to be put into the truck. In the meantime empties are transported to the dock where they must seize a truck resource before being loaded. This is necessary so that only one fork occupies the truck at a time. With each load, a check for last load is made. When found, the truck is signaled out of its wait state and the dock released to other trucks that might be waiting. Both the truck and container entities are disposed when truck loading is complete.

3.4 Line Feed

Line feeding is modeled by the cyclic process depicted in Figure 6. The cycle is triggered when a container entity has been emptied and the line resource released. When this occurs, an attempt is made to replenish the material just consumed at the line feed location. The empty container first checks to see if any fulls are available by searching the IS FULL detached queue. If so, the container entity is removed, delivery priority set, and a request made for a fork to transport this material to the line. If there are no fulls available, the need full flag is incremented. In either case, the empty is sent to the LINE EMPTY detached queue.

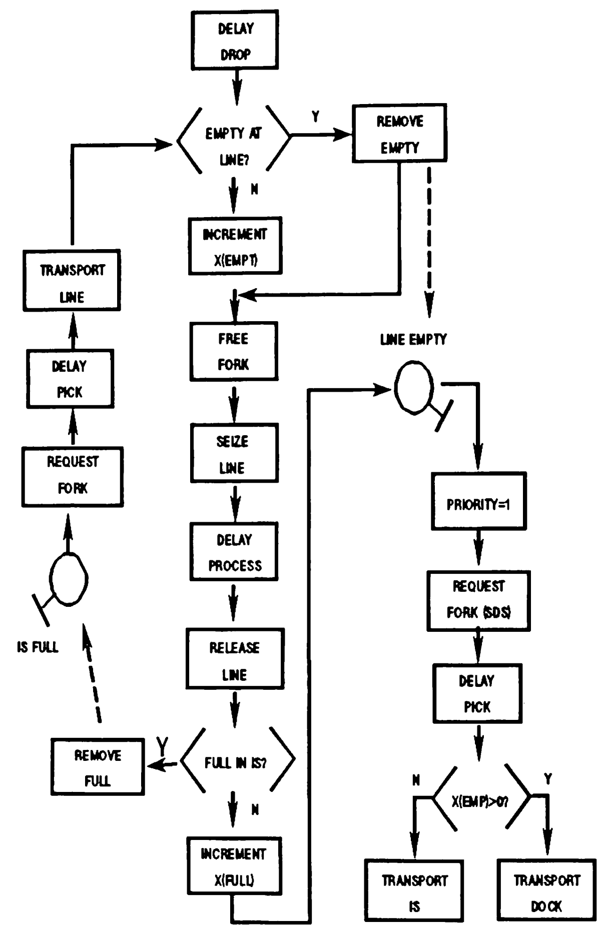

Figure 6. Line Feeding

Full containers arriving at the line delay for unload and then attempt to initiate a return of an empty to intermediate storage. Unless the full container is a direct delivery from the dock, an empty should be available at the line whenever a full is delivered. This empty, is removed from its detached queue, set to the highest priority (so its return will be by the same fork that delivered the full), and then put in a fork request block for transport back to either intermediate or dock storage. In the mean time, the full container releases its fork and then waits to seize the LINE resource for processing. This resource is held for the time necessary to process one container's worth of parts. In order to capture the peak material handling demand periods, it is important that actual processing rates be included in the model rather using a rate that is reduced to account for process interruptions. To do this, we explicitly model interruption through the use of a control entity which preempts the LINE resource when processing is stopped.

In summary then, it should be clear that actual material delivery systems cannot be modeled by simple simulation modeling constructs. Issues such as timing and control of the material movement and dynamic redirection require the more complex models we have presented here.

4 APPLICATIONS AND RESULTS

The primary performance measure of a material delivery and handling system is whether that system is able to support the demands of the production systems that it serves. The production line should <u>never</u> be starved for parts. By monitoring the LINE resources in the model, the simulation is able to track this performance. In addition, because material handling is not a value added function, we would like to minimize the cost of supporting production. Most of this cost is directly linked to the amount of material handling resources required. In these models, the resources are forklifts and docks. Thus, the utilization of these is another important piece of information that come out of the simulation. Material handling engineers generally want to keep the utilization levels of these resources sufficiently high (normally 80% for forklifts and 70% for dock).

The impact on system performance of any of the input data parameters given in Figure 2 can be studied with these models. Among the parameters that have been studied are buffer capacities, storage and line feed locations, speeds of line and forklifts, unloading/loading methods, work patterns, truck composition and their arrival pattern. In addition to the standard SIMAN output, customized reports, consolidated graphs and summaries are also produced by the simulation. The model structure presented here has also lent itself very nicely to animation using CINEMA. These animations have helped material handling engineers to visually study the effect of route and path layout patterns, forklift traffic congestion effects etc. Some real applications to which these models were used are described in the remainder of this section.

One of our first applications was an automotive assembly plant that was serving as a pilot implementation of JIT. This plant used hundreds of parts of varying types and sizes obtained from suppliers located throughout the country. A very complex delivery plan had been developed and captured in a large spreadsheet data base. The goal of our study was to use the simulation model to determine how that plan effected the internal material delivery system at the plant and whether additional resources were needed. Prior to this study, no estimate of the <u>in plant</u> implications of JIT had been made. The model helped us to study the effects of truck arrival pattern, returnable versus disposable containers, number of docks and number of forklifts on resource utilization.

The second application (at a transmission plant) was somewhat unique. Here the models were not used as planning or design tools but rather as operational aids. At the end of each week, the material handling department would run the models to determine how many fork lift drivers would be needed in each area based on the production schedule and anticipated deliveries for the next week. Thus, by simply changing the appropriate input data, the models could be used to help to allocate material handling labor to the appropriate areas.

The majority of our applications, however, have been at engine plants. In fact, the use of our models has been made mandatory whenever a new engine facility is being designed. These facilities consist of two major sections, a machining area where rough stock (unfinished blocks, cylinder heads, connecting rods etc.) are machined and an assembly area where these finish machined parts, along with others purchased from suppliers, are assembled into a complete engine. Because of the large size and value of the parts, material delivery is an especially important function in an engine plant. Several specific applications are listed below.

- o Several alternate plant layout designs are usually examined in each project, especially when a new facility is being added or replacing an existing one.

- o The tradeoff of having all docks located at one end of the plant versus point of use docks, where docks are distributed around the plant closer to line feed locations.

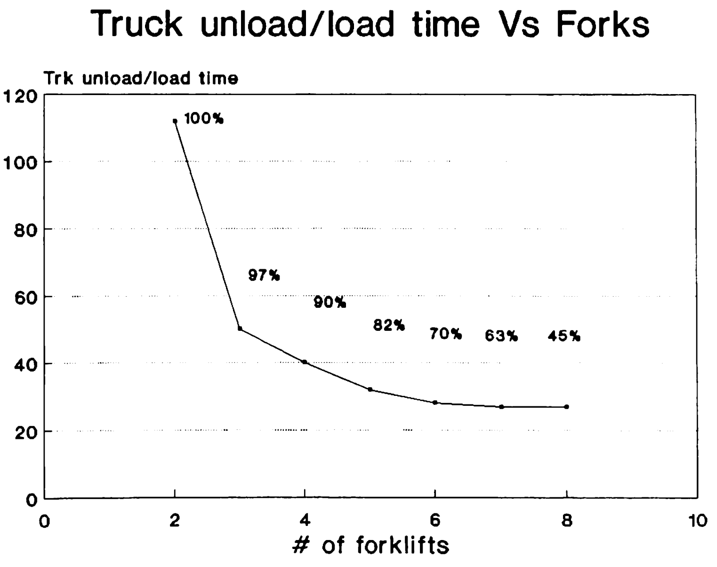

Figure 7. Cost/Performance Tradeoff

- o Effect of truck unload and load times based on the number of forks available at dock. Figure 7 shows a typical result of this sort of application. The model results can be used to construct a curve which clearly shows the cost/performance tradeoff. With more than 5 forklifts, there is little improvement in unload performance while the resources become increasingly underutilized.

o Dedicated versus shared forks for dock and line feed functions. This is an application where the model, by dynamically capturing the peak demand on the system, provides much more valid information than just using average demand numbers.

o Buffer sizes needed at dock, intermediate warehouse and line feed areas. The first two can be estimated by monitoring the average and maximum number of containers found in these areas during the simulation. The line buffers may be sized by repetitive runs with different values.

o Determining the truck arrival windows from supplier plants. The window is maximum time available after arrival for processing and releasing a truck. This quantity is crucial in establishing the viability of the incoming material transportation plan.

o The models were also applied to the engine shipping area where the roles of full and empty racks are reversed. Here, empties are received and racks filled with engines are shipped out. Modeling this process was accomplished with the same basic structure, modifying and adding some new fortran event routines where needed.

5 SUMMARY

The material delivery models described here have been used in a number of actual plant applications. In fact, much of the model's content and structure was developed in response to needs identified in these applications. By consolidating this work into a modular structure, we have now have a tool which is relatively generic and robust and capable of application to a number of different purposes and situations.

AUTHOR BIOGRAPHIES

VADIVELU JEYABALAN is a principal computer applications engineer at Ford. His research interests are computer networking, manufacturing systems modeling and simulation.

NORMAN C. OTTO is a Senior Research Engineer. His research interests include manufacturing systems analysis and computer aided engineering.