A SIMULATION MODEL FOR THE ANALYSIS OF RAILWAY INTERMODAL TERMINAL OPERATIONS

by

Roy D. Hammesfahr

Dissertation submitted to the Graduate Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Business

APPROVED:

E. R. Clayton, Chairman

L. J. Moore

B. W. Taylor, III

P. Y. Huang

W. E. Leininger, Jr.

August, 1981

Blacksburg, Virginia

ACKNOWLEDGEMENTS

I am deeply indebted to many individuals who have contributed their time, expertise, and encouragement in order to make this dissertation possible. Dr. Edward R. Clayton, my chairman, and patient counselor throughout my program, is especially recognized. His advice, inspiration, criticism, and support provided immeasurable, and indispensable, guidance throughout this research endeavor. I also wish to extend my gratitude to the other members of my committee: Dr. Laurence J. Moore, Dr. Bernard W. Taylor, Dr. Philip Y. Huang, and Dr. Wayne E. Leininger, Jr. Their support, suggestions, and contributions have indeed broadened my perspective and provided a valued learning experience.

I particularly wish to thank , Director of Intermodal Sales and Service, and his assistant, , Manager, Intermodal Planning and Control of the Norfolk and Western Railway Company, Roanoke, Virginia. This research could not have been possible without their keen professional interest and enthusiastic support. has personally provided most of the data necessary for the development of the intermodal terminal model, answered innumerable questions, and devoted many hours to the success of the project.

deserves special recognition for her contribution to this effort. She has spent many hours editing a text

which, to her, must have been totally foreign in the technical areas. Yet, her suggestions have been invaluable, and the entire document is improved through her participation.

Finally, I wish to thank my wife, . Her encouragement, understanding and participation far exceed the normally expected call to duty. Not only has she typed the entire text through incalculable drafts and subsequent changes, but she has also endured my trauma with exceedingly good grace. I will always be greatful for her support and enthusiasm for my work.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

THE PROBLEM

Introduction

The movement of freight in trailers or containers is a convenient and economical method of shipping non-bulk commodities and merchandise for both the shipper and the mover. The shipper accrues the benefits of containerization by taking advantage of the transferability of a shipment between modes of transportation. The mover benefits through the use of standardized equipment that is designed specifically to reduce handling. A manufacturer can transship a product from a plant on the West Coast of the United States to a distribution warehouse on the East Coast utilizing both over-the-road and rail transportation. This enables the shipper to take advantage of the less expensive rail transportation over the long haul. However, the benefit could be offset by increased handling cost and freight charges if the shipment were non containerized. Containerization enables a more efficient transition from the road to the rail and back to the road again. Additionally, units within a container require handling only at the origin and final destination. The anticipated cost reductions and improved efficiency provide the basic motivation for freight containerization.

Containerized freight movement is achieving an increasing role in all common modes of transportation. These include tractor-trailer/container (truck) combinations that move over-the-road, flatcar trailer/container combinations that move over-the-rail, sea-borne movements where trailers/containers are carried by ships, air-borne shipments in highly specialized containers, and tug-barge combinations on rivers and inland waterways. The latter two modes are at opposite ends of the spectrum. Air freight movement usually involves high-value, high-priority items while river freight is most often low-priority non-containerized bulk material. Consequently, containerized interfaces between these two modes are extremely rare. Containerization interfaces between trucks, rail, and ships are quite common, however, and they occur in any combination. Trucks deliver and pick up containers at both rail and sea port terminals. Sea-borne containers originating at foreign ports are transferred to the rails at ports in the United States, transported to a distant rail terminal, then coupled to trucks for movement to a final destination. The capability to transfer containerized freight from one mode of movement to another has led to the development and expansion of this method of transporting freight. It has assumed an important role within the industry and is referred to as Intermodal Operations.

In the Railroad Industry, intermodal transportation is variously called piggyback, flatback, trailers-on-flatcars (TOFC), or containers-on-flatcars (COFC). Intermodal trailers are specially constructed to be more rugged than the familiar truck trailer and are equipped with a permanent undercarriage. They are usually associated with the interface between over-the-road and rail transportation. There is increasing interest in truck/rail/ship combinations with trailers, however. Containers are of two types; those that can be fitted with wheels for over-the-road transport, and those that cannot. Containers are most common in rail/ship combinations. Railroad flatcars are specially designed for intermodal use. In general, there are three types: those that can carry only trailers, those that can carry only containers, and those that can haul both trailers or containers. These terms, trailers and containers, are often used interchangeably. Container sizes are being standardized at eight feet high, eight feet wide, and in lengths of ten, twenty, thirty and forty feet.

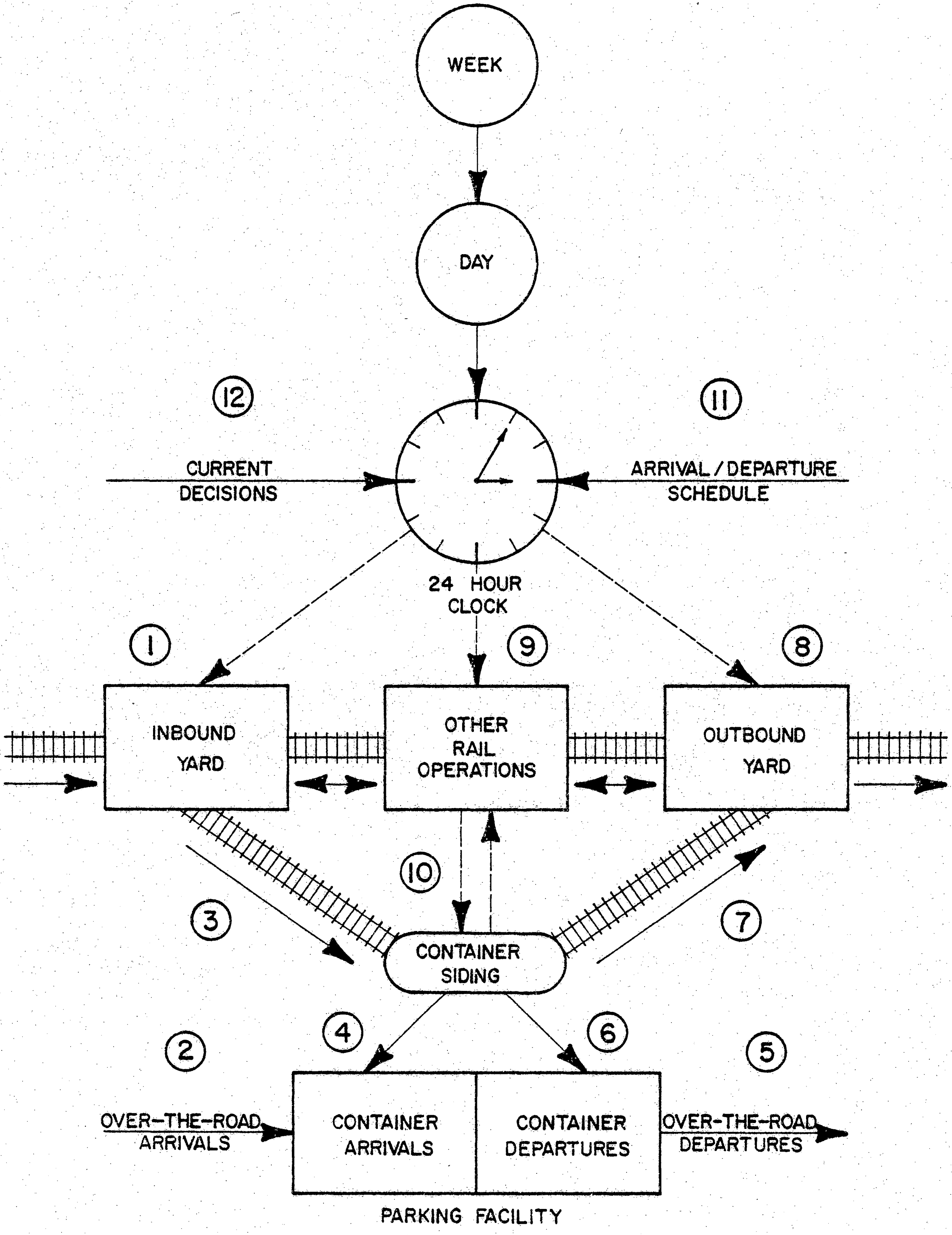
Intermodal rail transportation originated in the late 1800's when some farmers moved their produce wagons to city markets, and circuses shipped circus wagons between cities on railroad flatcars. However, it was not until the mid 1950's, when the Interstate Commerce Commission declared piggyback service legal for railroads, that this mode of transportation began to expand to its current magnitude [74]. For example, during the first nineteen weeks of 1980, there were approximately 623,000 piggyback flatcar loadings reported in the United States transporting about 1,150,000 trailers and containers. This volume is second only to the 2,204,000 coal cars for the railroad industry. In third place are the estimated 512,000 chemical car loadings during this same period [65].

The literature indicates that utilization of TOFC/COFC will continue to expand and that traffic increases will be stimulated by new technology in terminal and train operations. It has also been predicted that future growth for this mode of transportation will be at twice the rate in the next ten years as it has been over the past ten years [64]. Some justification for this optimism is founded in the advances that individual railroads, and the industry in general, have made in intermodal service [52, 55, 57, 63, 64, 70, 72, 87, and 88]. Morash, Hille, and Bruning [68] suggest that the railroads can increase intermodal traffic by concentrating on the shipment of high valued manufactured items and regaining shipments lost to the trucking industry. The only voice of pessimism encountered is by Beier and Frick [51]. They conclude that significant increases in traffic can only be attained if the cost of TOFC is reduced and services are improved. It is apparent that the development of innovative management analysis techniques will be an aid to the industry in seeking to attain the anticipated potential for growth in intermodal rail service. Intermodal terminal operations provide an area where new analysis techniques could prove to be very beneficial in planning for future growth and improving existing services.

Intermodal Terminal Operations

Intermodal railway operations are second only to coal in terms of the number of cars loaded by the nation's railroads [65]. These operations consist of receiving containerized freight at one terminal, loading the containers on flatcars, then delivering them by rail to another terminal where they are unloaded and transshipped to a final destination. The containers arrive and depart the rail terminals via rail, motor freight, or ship, and they must be compatible with the railroad equipment and the equipment of the interfacing mode of transshipment. The railway intermodal terminal area is a critical point for these operations.

Some insight into the complexity of intermodal operations in a rail terminal area is provided by Figure 1. This drawing illustrates the interface between motor freight and rail transportation. Containers arrive at the terminal from two sources; (1) the inbound yard by rail, and (2) the parking facility by truck. Containers arrive at the inbound yard on flatcars and fall into two classifications. They are at their final rail destination and are to be off-loaded from their flatcars, or they are to be transshipped, by rail to another terminal. If they are to be transshipped, the flatcars must be reclassified (9) according to their destination and coupled to a corresponding outbound train. If the containers are at their rail destination, the flatcars are "switched" from the inbound yard to the container siding (3) where they are off-loaded onto a ramp and transferred to a parking lot on the ground (4). These containers remain at the parking facility until they are picked up by a tractor (motor freight truck) for over-the-road transportation to their final destination (5). Containers arriving by truck (2) are decoupled from their tractor and are also placed into the parking facility. These remain in the lot until they can be scheduled for rail shipment to their destination. When these outbound containers are scheduled for departure, they are moved to the siding (7), loaded onto flatcars, and switched to an outbound yard (8). There,



INTERMODAL RAILROAD OPERATIONS AND RAILROAD YARD ACTIVITY

FIGURE 1

the flatcars are classified by container destination and coupled to outbound trains made available through other rail operations (9).

Figure 1 also illustrates the container siding and switching operations that must be coordinated with all other rail operations scheduled, or in progress, in the yard (10). These operations are conducted around the clock, every day of the year. Events, like the arrival and departure of trains at the yard, are scheduled (11). Current decisions (12) involving yard operations, such as classification and switching activities, must insure compatibility with the schedule of all yard related events. This includes the loading and off-loading of containers on flatcars.

Classification is one of the primary purposes of a typical rail yard. Trains arriving at the yard are broken down into individual cars and the yard supervisor decides where, among the several tracks available in the inbound yard (receiving area), each car is to be placed. A car that is at its destination is scheduled to be switched to a siding. If a car is to be forwarded to another destination, it is reclassified, consolidated with other cars for the same destination, and transferred to the outbound yard. Additionally, individual cars originating at the yard are classified in accordance with their destination before they are moved to the outbound yard. In the outbound yard, also consisting of several tracks, the train of cars is coupled to locomotives for scheduled departures.

The classification process is very complex. Information must be available on each car of a train prior to its scheduled arrival in the yard. Computers are employed by many railroads in order to provide this information. For example, approximately two hours before a train arrives at a yard, data is passed to a computer at the yard by a master computer which provides the make-up of the train. This information includes the weight, destination, and precise location of each car of the train. Current decisions can then be formulated, based on this information, to position the cars in the inbound yard and update the yard inventory (6). Switching to sidings and the yard classification activity can then be scheduled.

The Norfolk and Western yard, located in Roanoke, Virginia, is an example of how classification is accomplished through the use of a "hump". The hump is a small hill located at the end of the inbound yard. Cars are moved to the hump by a switching engine. There, they are sent over the hump individually, gravity providing the momentum, into the classification area. An operator in a control tower directs the cars onto one of the tracks of the classification area. Instrumentation, including wheel sensors, photo-electric sensors, weight-in-motion scales, wind gauges, and speed-reading-radar provide data on each car to a computer as it moves down the hill. Calculations are made to determine changes in rate of speed, weight, length, and the number of axles on the car. The speed of the car is then adjusted, by an elaborate breaking system built into the tracks, to provide just sufficient speed for the car to couple with the previously spotted car at its designated position on a classification track. The tower operator, in addition to selecting the receiving track, enters the number of cars to go on a track, and the first and last car numbers for the track. The operator also maintains the capability to override the automatic system. Once all of the cars for a designated train have been classified, they are moved to the outbound yard by a switching engine.

The intermodal terminal manager must function within the yard environment. Additionally, he must deal with the special circumstances that are peculiar to intermodal operations. For example, because there are three basic types of flatcars, some cars that are off-loaded at the terminal siding may have to be switched from the siding empty if matching containers are not available in the parking lot. Likewise, empty cars may be required at the siding to provide for specific containers that are to be shipped.

The flatcar capacity of the siding, in conjunction with physical characteristics of the loading/off-loading ramp and associated equipment, are physical constraints in the operation of the intermodal terminal. For each terminal, there is a finite number of cars that can be at the siding at any point in time. The time required to off-load and on-load these cars is dependent on the type of equipment employed, the layout of the ramp, and the proximity of the parking lot. The equipment varies from terminal to terminal; it ranges from the very expensive, highly specialized and fully automatic side-loaders, to a relatively inexpensive "circus" type of operation employing tractors. Containers are handled with side-loaders at terminals where the volume of traffic justifies their application. Circus handling, where the containers are driven on and off the flatcars, is employed at low traffic volume terminals.

The capacity of the parking lot and its location relative to the siding and adjacent ramp are also finite limiting factors in the terminal operation. Trailers are arriving and departing the lot daily. The maximum number permitted in the lot at any point in time is a function of the physical size of the containers, the space required for maneuvering them, and the total area available for the lot. A conflict in scheduling, or an increase in traffic, could conceivably drive the lot beyond saturation, interrupt operations, and create potentially costly delays.

Statement of the Problem

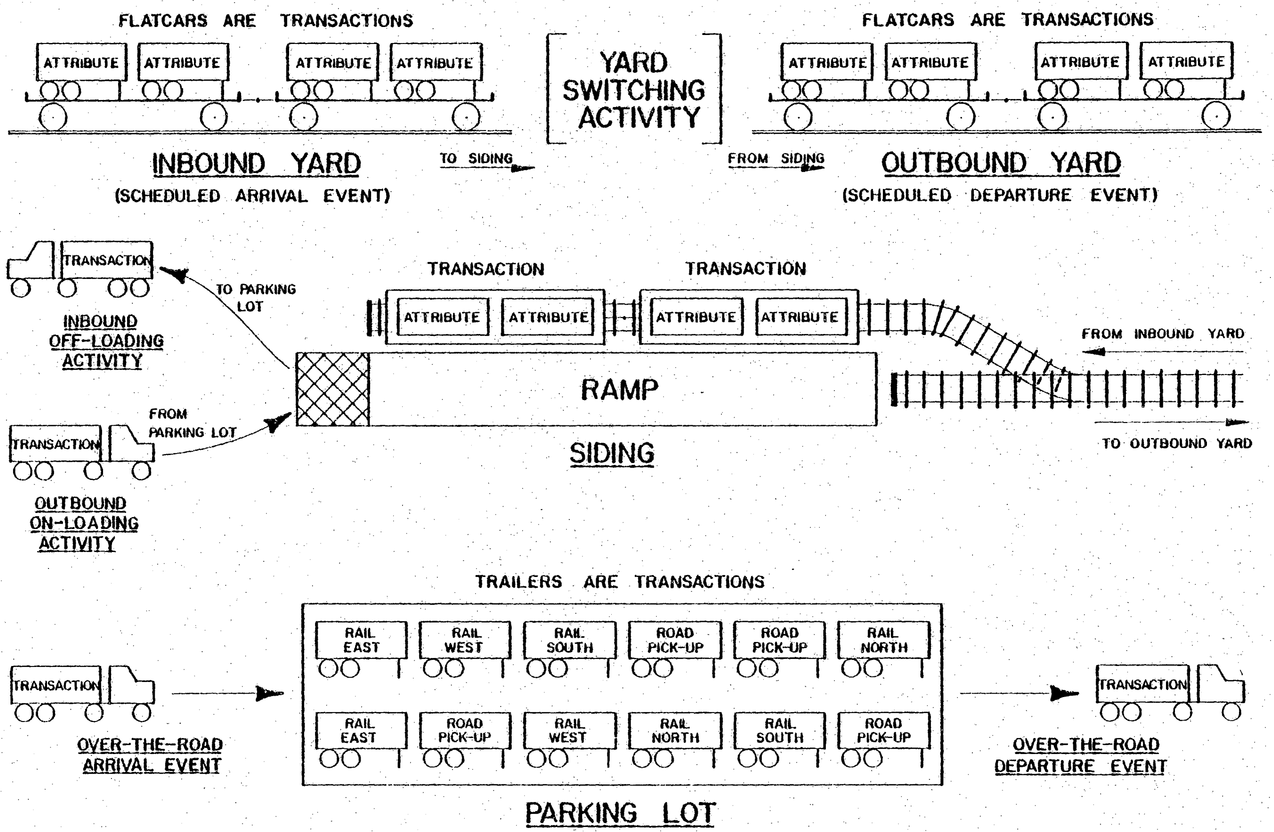
The complexity of terminal operations confronts the intermodal manager with some difficult problems, including the following. Given an increase in intermodal traffic, at what point could the manager expect the parking lot to exceed capacity? What changes would be desirable in the yard switching schedule to promote overall efficiency and parking lot utilization? When should the ramp be expanded, or new equipment, or personnel added? What are the expected costs of these actions?

The problem is that to experiment with the real system in order to assess the effects of the above described changes would be time consuming and costly, if not impossible. A computer simulation model adequately representing these facilities could provide a valuable management analysis tool to seek acceptable solutions to these problems at a reduced cost, in a shorter period of time, and without disrupting current operations.

Purpose

The purpose of this research is to develop a Queueing-Graphical Evaluation and Review Technique (Q-GERT) computer simulation model of a railway intermodal terminal system, incorporating the daily arrival/departure train schedule, the yard switching schedule for the terminal, and over-the-road traffic patterns to analyze the sensitivity of the system to changes in intermodal operations. For example, given a fixed parking lot capacity, a decision to unload containers at the siding at one point in time, because it would be compatible with the yard switching schedule, may result in an over-capacity situation in the parking lot. Yet, the same decision at a different point in time would be feasible. A computer simulation model of the terminal facility that would enable the intermodal manager to anticipate and evaluate schedules and contingencies prior to implementation would be very desirable.

Figure 2 illustrates the general operation of a typical railway/over-the-road intermodal terminal. Externally scheduled events are the arrival and departure of trains coupled with flatcars loaded with trailers. Each flatcar in the system (terminal) represents a transaction that must be processed in a specified manner. If it is arriving, the type of car, number and type of containers, and the cars' direction are recorded as attributes of the transaction. If it is to depart the system, the direction of departure (destination) and number



INTERMODAL RAIL/OVER-THE-ROAD TERMINAL OPERATIONS

FIGURE 2

of trailers associated with each car are attributes.

Internally scheduled events are: (1) the departure of a flatcar(s) from the inbound yard to the siding, (2) the arrival of a flatcar at the siding from the inbound yard, (3) the off-loading of the containers from the car, (4) reloading the flatcar with trailers, (5) the departure of a flatcar from the siding, and (6) the arrival of the car in the outbound yard. Other events that are dependent on these scheduled events are the arrival and departure of trailers in the parking lot via the siding. Each flatcar arriving at the siding generates transactions in the parking lot in accordance with the number of trailers that were associated with it when it arrived in the system. Likewise, each flatcar that is loaded at the siding reduces the number of transactions in the parking lot by the number of trailers that are loaded onto it. Each transaction in the parking lot has associated attributes designating type, mode of arrival (rail or over-the-road), and destination.

Stochastic activities and events are also an integral part of the system. However, they are dependent on the scheduled trains entering and leaving the yard. For example, over-the-road arrival and departure events at the parking lot occur randomly during the day. Yet, they are often generated by the arrival of a train in the yard, or the scheduled departure of a train from the yard. A trailer over-the-road departure from the parking lot is "activated" when the intermodal terminal supervisor is informed of the impending arrival of a train in the inbound yard. He is notified of the number of flatcars on the train that have trailers on them, the consignee of each trailer, and the car on which each trailer is located. Given this information, the supervisor then plans the future activity of the terminal, which may span several days. He will notify each trailer consignee when their trailer has arrived in the inbound yard, and the date and time he estimates the trailer to be on the ground in the parking lot ready for pick-up. The consignee will then make arrangements to pick up the trailer, and often, but not always, will deliver another trailer to be shipped in the process. This pick-up and delivery process for the parking lot, though related to the train schedule, follows a probabilistic distribution that is the result of independent decisions made by individual consignees.

Objective

The purpose of this research is to produce a model of intermodal railway terminal operations to obtain the following statistics:

1. The maximum number of containers in the parking lot.
2. Server utilization for the on/off loading activity at the siding.
3. Yard switching time.
4. Waiting time of containers at the parking lot to be shipped by rail.
5. Waiting time of containers at the parking lot to be picked up by consignees.
6. Waiting time of flatcars at the inbound yard.

7. Waiting time of flatcars at the outbound yard.
8. Siding/ramp utilization.

The primary objective of this research is to provide a management tool that will enable decision makers to view their intermodal terminal system as a whole, over time, to estimate their resource requirements and operational capabilities for given levels of activity. One should be able to vary the system parameters, either individually or simultaneously, and observe the expected changes in the system. For example, given an increase in traffic arrivals, what changes would be desirable in the yard switching operations, and what effects could be expected in the parking lot area? Should this situation be anticipated, managers could identify potential problem areas through the use of the simulation model and project planning and budget requests accordingly.

After the model has been developed, verification and validation described, then an example of model implementation is presented. This presentation includes the simulation of the current system's operation and indicates expected changes in the system when existing parameters are varied. This information is then used to illustrate how the simulation results could aid in future facility planning.

Significance

This research has been undertaken as a result of an expressed need for additional planning and evaluation techniques in the area of intermodal services by intermodal managers in the railway industry

[100 and 101]. The primary concerns of these experts are threefold: (1) Increased intermodal traffic is anticipated; at what point would existing terminal facilities reach saturation? (2) What scheduling and/or procedural changes in existing terminal or switching operations would result in greater efficiency? and (3) Can efficiency be improved and cost effectiveness maintained by relocating, modifying existing facilities, or by acquiring additional equipment?

Given the current magnitude and forecasted growth of intermodal containerized freight transportation by rail, and the complexity of the problem, some managers expect that a simulation model of existing systems could provide an excellent planning and analysis tool for intermodal managers throughout the industry. Furthermore, one expert [78, 79, and 80] has indicated that sound management techniques, economic costing, and management information systems must be developed and maintained if intermodal carriage is to be viable on the railroads, and if intermodal traffic is to reach its anticipated potential. The model developed through this research could assist in meeting these objectives.

The model described in this paper is adaptable enough, with some modifications, to be utilized by many railroads having intermodal facilities. The Federal Railroad Administration (FRA) has demonstrated more than a passing interest in intermodal services by awarding contracts totaling $500,000 in 1977 to two consulting firms for the development of long-term technological improvements and intermodal innovations [39 and 89]. Welty states that one of the consultants concludes:

> ....the most significant benefits to intermodal systems could result from improvements in terminal operations, with additional improvements to come via betterment in line-haul operations....

This consultant was looking at terminal design, among other alternatives, to identify potential improvements in layout, terminal access, and general increases in efficiency.

The availability of a general simulation model for intermodal terminals would provide a means of system analysis that has not been previously possible. This model satisfies a known demand for management information and could lead to the advancement of intermodal system technology.

Scope and Limitations

Scope

The research consists of the development of a computer simulation model that approximates the operation of an intermodal railroad terminal. The model is designed such that specific events and activities are provided for. These include:

1. Arrival and departure schedules of trains with containers/trailers.
2. Quantity and type of arriving containers/trailers.
3. Yard switching rules.
4. Flatcar onload/offload ramp procedure.
5. Outbound car handling procedure.

6. Parking lot arrival/departure distributions.
7. Siding/ramp capacities.
8. Parking lot capacities.

Having provided for these events and activities, containers and flatcars moving through the terminal facility can be simulated. The objective is to first simulate an existing facility by inputting historical train arrival/departure schedules, yard switching rules, siding/ramp capacities and parking lot capacities with all associated historical service times. Maximum queue lengths are recorded, average waiting times and server utilization are estimated for the simulation. Subsequent simulations are then conducted and the results compiled. An average maximum number in the queues, and an overall (grand) average for waiting times and server utilization could then be computed for the combined simulation. These overall averages provide the basis for the comparison of the results of subsequent experiments. Subsequent experiments are performed after changing one, or more of the parameters associated with schedules, capacities, or service time distributions. An analysis can then be undertaken on the desirability of implementing changes in the system. Specifically, the model would aid in the analysis of:

1. Parking lot requirements.
2. Alternate ramp/deramp procedures.
3. Siding/ramp capacities.

4. Hostler requirements.*
5. Proposed new facility evaluation.
6. The desirability and implications of seeking increases in traffic.

For example, assume a terminal facility is currently operating within its capacity, at the present level of intermodal activity arriving and departing in the yard. The Marketing Department estimates that traffic will increase twenty percent over the next two years. What changes could be expected to occur in terminal operations, and what alternatives exist to assist the manager in dealing with these changes? The current system is simulated, using the past year's arrival and departure train schedule, current service times and yard, parking lot and siding capacities. Maximum average queue lengths, server utilization and siding utilization are obtained. The arrival and departure schedule is then increased by twenty percent, the simulation repeated, and the results recorded. Analysis of the results of the two simulations may indicate that the inbound yard queue is steadily increasing, and would eventually become infinitely large. The corrective action indicated might be to increase the scheduled switchings from the inbound yard to the siding from once to twice a day. This change would be incorporated in the model, the modified system simulated and the

*The hostler activity includes removing containers from the ramp and placing them in the parking lot, or taking them from the parking lot to the ramp.

results analyzed. The procedure would be repeated, changing parameters individually or simultaneously, until an acceptable solution is obtained.

The analysis should, of course, include the estimated cost of the various alternatives as compared to the expected increase in revenues. The decision may be that an increase in traffic for a given facility is not desirable. This type of information should, if it exists, be passed to the appropriate levels of management for further consideration and overall planning.

Limitations

The focus of this research is on the development of a computer simulation model for a railway intermodal terminal system and to demonstrate how the model could be used as an aid to the management decision process. Time constraints do not permit the investigation of all possible applications of the model to this process, however, experience with the model should suggest additional areas for exploration and research.

The model is intended to duplicate, as nearly as possible, the flow of flatcars and containers/trailers through an existing intermodal terminal. It is not designed to represent a new, or innovative system that can be compared to the current system. The purpose is to gain information about the capabilities and characteristics of what is in being to promote efficiency and plan for future activity. This is not to say that the model could not be modified to compare one system to another, but first a working model of the original system must be developed, which is the primary thrust of this research.

No attempt is made during the course of this research to develop an algorithm that seeks optimization of the system. A primary limiting factor in an intermodal terminal operation is the maximum capacity of the parking lot. Therefore, an important question to be addressed through the simulation is at what level of activity will the parking lot reach saturation; not the optimum level of activity that can be supported by the parking lot. This is not to imply that an optimum seeking algorithm, or search technique, could not be developed, to optimize selected parameters. This could be very beneficial in evaluating proposed new facilities. However, optimization should be the topic of future research efforts once an operational model of intermodal operation has been developed.

Further limitations of this research are the same as those associated with all simulation models. Since the data used to develop the model is supplied by a single railroad, this may not be typical of railroads in general. Thus, the validity of the model is dependent on these data and the design of this railroad's terminals, which also provides the logic used to design the components of the model. The railroad does operate several intermodal terminals, however, and the number of flatcars and containers processed by the railroad are a matter of record. Therefore, the construction of the model is such that with slight modification and the appropriate changes in the input data, more than one facility can be simulated. The validation of a simulation for a facility can then be checked, and confidence in the simulation established, by comparing the output measures to historical output measures for the terminal. Methods used to test the output of the model are specified in the validation description.

Review of the Literature

A search of transportation research, including highway, maritime and rail transportation was conducted. The search was made through the services provided by the National Aeronautics and Space Administration/University of Kentucky Technology Applications Program (NASA/UK TAP) which provides computer access to over one hundred ninety (190) databases. The Transportation Research Information System was the data base accessed for this search, using various combinations of the following key words:

1. Railway, Railroads, or Rail
2. Intermodal, or Piggyback
3. Rail Intermodal Terminal, or Rail Terminal
4. Rail and Tractor-Trailer
5. Rail and Truck, or Motor Freight
6. Rail and Ship
7. Rail and Truck and Ship
8. Container on Flatcars, or Trailer on Flatcars
9. TOFC, or COFC, and TOFC/COFC
10. Simulation, or Computer Simulation, or Models, or Computers
11. Domestic or Foreign, or International, or Europe, or Asia

Three hundred forty (340) references relating to intermodal transportation were identified. Thirty-eight of these addressed areas of intermodal operations associated with maritime, rail, and over-the-road terminals, both foreign and domestic. The search produced no evidence of a computer simulation model of an intermodal terminal. A search of the periodical literature also discloses no previous publication in this subject area.

Related research in the intermodal terminal area is quite limited, but does provide technical support for this study. For example, the Federal Railroad Administration's Office of Research and Development (ORD) initiated a plan for a two phase study to investigate the viability and improvement of intermodal rail freight service in 1977 [40]. Phase I, completed in August, 1978, entails a comprehensive study of current intermodal equipment, operations, and technology [16, 25, 26, 34, and 39]. Alternative systems and equipment proposals were incorporated into the study and the most promising alternatives identified. Phase II of the project, scheduled for implementation in October, 1979, and currently pending, is to provide a more detailed examination of the most acceptable alternatives identified in Phase I. Two independent consulting firms were involved in the first phase of this study, each preparing their own reports, using different approaches [40].

One report on Phase I to the FRA, by the A. T. Kearney Company [34], contains a computer simulation of alternatives to the present intermodal system. The simulation is an approximate model of terminal to terminal intermodal rail capabilities with existing equipment and capacities. Proposed equipment design changes are incorporated into the model and the results compared with previous runs. The objective of the simulation is to identify which proposed new equipment designs would be most promising to pursue for future development. For example, flatcars with the capability of stacking containers such that four could be carried per car might be desirable. However, this new capability would require redesign and construction of many of the railroad tunnels on existing rail lines. The simulation, written in SIMSCRIPT, addressed over-the-rail terminal to terminal activity. A recommendation of the study is that a model be developed for intermodal terminals in order to study these operations [16 and 39].

One Japanese study [59] reported the use of a simulation model to evaluate an automatic container transfer device. The model was to aid in the design of the terminal facility and its layout in order to obtain practical utilization of the transfer equipment. The simulation estimates expected transfer times for the movement of containers to and from shipboard and storage areas for a maritime terminal. No rail transportation interface was provided, nor were individual containers accounted for.

The European literature primarily addresses difficulties associated with the different rail systems from country to country as it relates to intermodal transportation [20, 23, 37, 50, 61, 62, 81, and 83]. The primary intermodal emphasis in Europe is on sea-borne containers that are moved by rail, or containers that are moved by rail or truck to maritime terminals for shipment. The containers used for this traffic are more compact and durable than over-the-road trailers. One paper [56] describes the terminal at Hamburg, West Germany, where it is possible to stack as many as 4,000 of these containers five deep. Large, fully automated cranes provide the equipment handling for this system, and the operation is reported to be the most modern on the Continent.

Over-the-road/rail interfaces, in Europe, however, are more dependent on the physical construction of the trailers for road transport and the restrictions imposed by the existing rail systems. For example, width is limited by the clearances required for passing trains. Height is restricted by tunnel and bridge construction. Trailer length has similar constraints. Consequently, the standard flatcar, as used domestically, does not work well in Europe. Specially designed railcars, with sunken beds, or entirely enclosed, are often used [62 and 81]. These require special handling equipment and techniques, in addition to special cars, which increase cost. These higher costs and physical restrictions, coupled with different laws from country to country (trucks cannot use the highways on weekends in many areas of West Germany [23]) tend to retard the development of over-the-road/rail transport in Western Europe [20 and 61]. Therefore, this intermodal method of freight movement is not as highly utilized there as it is in the United States.

The literature addressing intermodal terminal operations in this country clearly indicates that new analysis techniques are required in order to seek improved efficiency [15, 17, 19, 31, 54, 82, and 90]. One of the major concerns is the location of over-the-road/rail interface terminals in metropolitan areas and the corresponding contribution to congestion and traffic delays. Mayer [32] recommends that more emphasis be placed on the development of models of freight movement, corresponding the same considerations given people movement, for urban and metropolitan planning. In 1976, Kuhns and Mulinazzi [60] developed a methodology that considers rail, highway, market, and land characteristics in the selection of terminal locations in metropolitan areas. However, their analysis does not include any considerations of the intermodal terminal operation, which may be significant in terminal location decisions.

The PRC Systems Sciences Company was contracted by the FRA to develop an Intermodal Management Information System (IMIS) in 1977 [35, 36, and 37]. The IMIS, completed in September, 1979, was developed through the cooperation of the Norfolk and Western Railway Company. This system provides for repetitive waybilling, profit analysis, and intermodal equipment control. The equipment control feature primarily provides information on the movement and location of flatcars and containers, in approximately a real-time sense, for the entire railroad. Intermodal experts [100 and 101] indicate that the development of a computer simulation model of the intermodal terminal facility operation would provide a valuable management analysis technique that could supplement the IMIS in the decision process.

Computer simulation studies of railroad operations are extremely rare, and such studies related to intermodal railroad operations are practically non-existent. The only simulation of an intermodal system that has been identified is the one reported by the A. T. Kearney and Company, Inc., that was discussed earlier in this section [34]. A simulation model was developed by Tripp [41] in 1972, to aid in the analysis of the economic impact of developing multi-modal transportation companies. Recently, Whitehurst [91] concluded that the time has come to give multi-modal companies an opportunity to organize. A computer model was used by Bushness, Low, and Pearsall [18] to analyze shifts in modes of freight transport due to changing cost factors.

Wong [94, 95, 96, and 97] is the most active in the railroad simulation field having developed railroad yard related models to study yard design, car utilization, track assignment, and scheduling procedures. The scheduling model is a study utilizing a simulation model which he calls a Dynamic Movement Predictor. A railroad network model has been developed by Minger and Cetinich [47] for the Association of American Railroads which utilizes SIMSCRIPT. However, it was first reported in 1969 as a "pilot model" and only one reference to the model has been identified since that time. In 1975, Shen, Kang, and Kang [49] developed a SIMSCRIPT based model of a small passenger service railroad in Taiwan. This paper made reference to the above mentioned model, but also acknowledged that their model was not fully developed. Assarabowske and Sussman [42] reported a simulation study of railcar utilization under assigned "fleet" operations in 1976. No mention of the language the model incorporated is included, however, and it is assumed to be written in FORTRAN.

Two studies were identified where simulation was applied to intermodal terminal operations. The first, by Holland and Clayton [46], developed a GERT simulation model to study the effect of changing locations of intermodal terminals and the expected change in equipment costs. It was observed that by relocating facilities in the Chicago area a thirty-nine percent reduction in costs could be realized. The second paper, by Rakes and Clayton [48] used the GASP IV Simulation Language to study the cost effectiveness of constructing a new intermodal ramp at an existing facility. The study addressed the problems and cost associated with flatcars waiting in the yard to be offloaded, waiting time at the ramp, and switching. No attempt was made to include parking facilities at the ramp, crew size, equipment changes, or daily scheduling. These factors, in addition to others, are provided for in the simulation model developed through this research project.

CHAPTER II

DESIGN CONSIDERATIONS OF THE SIMULATION MODEL

The prototype simulation model developed in this research was patterned from the Norfolk and Western Railway Intermodal Terminal located at Roanoke, Virginia. Experience gained by on-site observations of the operation of this facility and interviews with the intermodal managers of the railroad provided the technical background related to these operations that was required for the construction of the model. The nature and magnitude of the pertinent variables and parameters that affect the operation of the terminal were identified and are incorporated into the model. Scheduling input and operational data used to verify and validate the model are based on actual events that have occurred and operational criteria in effect at this railway yard. The Roanoke facility has also been used as the testing ground for the implementation phase of this research, which is presented in detail in Chapter IV.

Intermodal Terminal System Characteristics and Simulation Model Programming Considerations

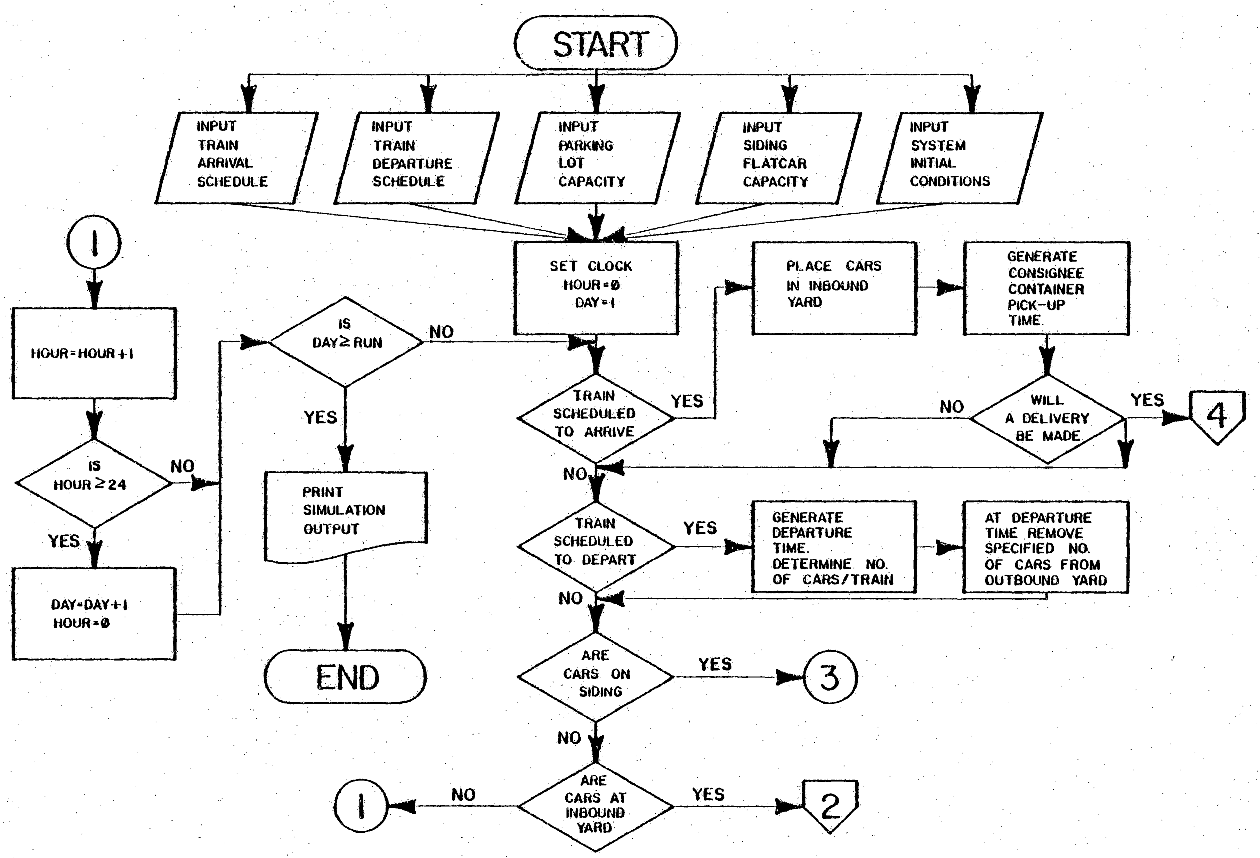
Preliminary investigation and the related review of the literature indicate that the intermodal terminal operation essentially consists of the following primary elements:

1. Trains enter the yard area with flatcars that are designed to carry containers. These flatcars are loaded to capacity, partially loaded, or they are empty.
2. The flatcars wait in the yard until they can be transferred (switched) to a siding that is equipped to unload the containers.
3. Flatcars arrive at the siding, which has a limited capacity, and:
   a. Wait until the containers are offloaded
   b. Wait until containers are onloaded
   c. Wait until they can be transferred to the outbound yard
4. Containers arrive and depart the terminal parking lot, which has a limited capacity, via:
   a. The siding
   b. The front gate (over-the-road)
5. Trains depart the yard area with container type flatcars for various destinations on the rail line.

   The flatcars wait in the outbound yard area until they are coupled to a train for departure.

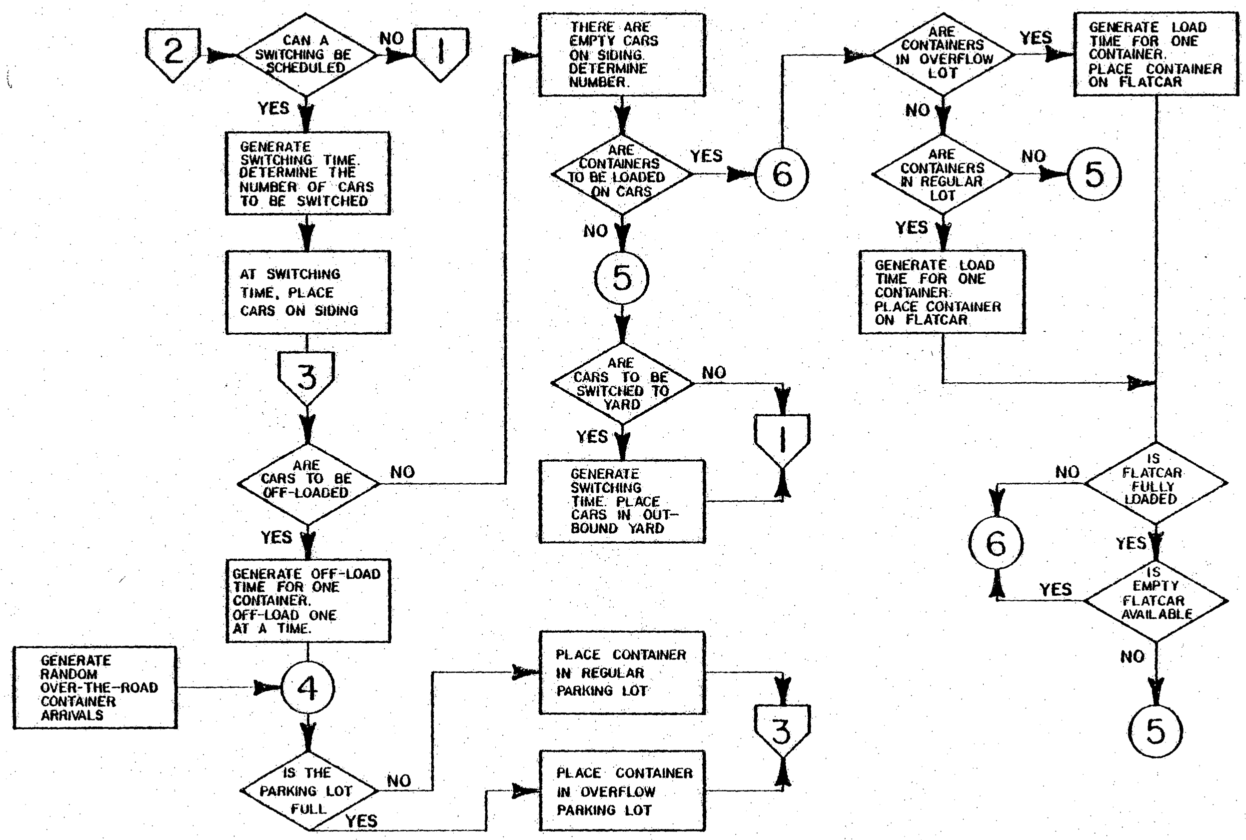
A flowchart of this operation is presented in Figure 3.

Railroads operate twenty-four hours a day, 365 days a year. Trains generally arrive and depart the terminal areas at published



INTERMODAL TERMINAL OPERATIONS

FIGURE 3



INTERMODAL TERMINAL OPERATIONS

FIGURE 3 (CONTINUED)
]

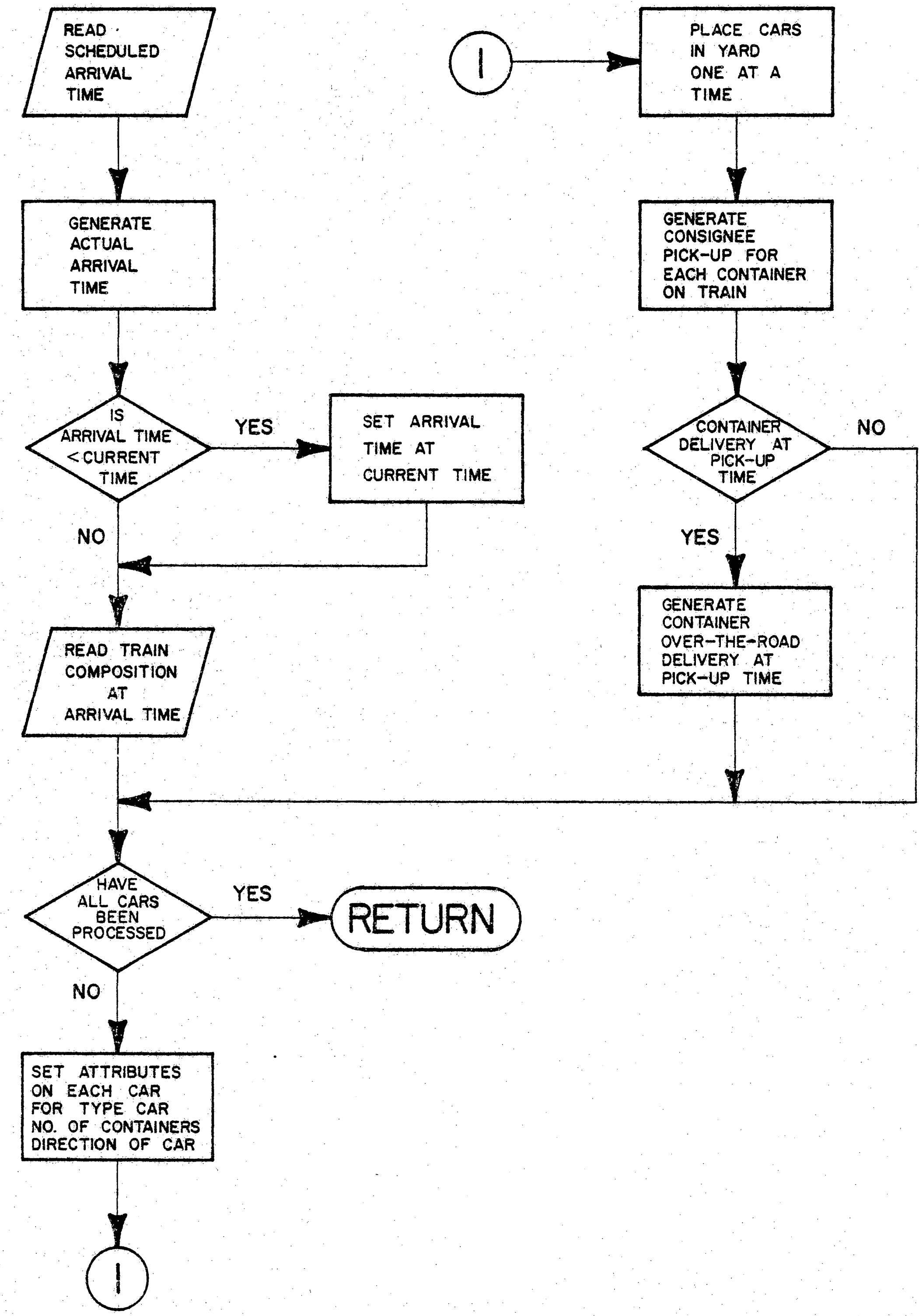
scheduled times. Each terminal has a finite capacity, and at any point in time there are a given number of cars, trains and containers in the terminal area with associated activities in progress. A simulation of the terminal's operations should make provision for these conditions. As Figure 3 illustrates, the terminal's initial conditions are provided for, and the daily arrival/departure schedules for the period of the simulation are filed to be called at the specified times in the simulation.

A basic feature of the terminal operation is associated with time. Not only do trains arrive and depart at specified times, but every activity within the terminal is related to the time of day. Therefore, the simulation should account for time in a manner that enables each event and associated activity to be accomplished and accounted for in a realistic sense. The twenty-four hour clock can be reasonably duplicated by establishing a basic time period for the simulation as one hour. Minutes and seconds are then provided for by dividing the one hour time period into decimal tenths and one-hundredths. For example, an event scheduled at time 10.12 would indicate that the event is scheduled to occur at seven minutes and twelve seconds after the tenth hour. Days can be recorded by incrementing a day counter by one at the end of each twenty-four hour period, and resetting an hour counter to zero.

Once the methodology for the clock is established, all activities and events can be modeled such that they can be scheduled, initiated, or concluded at specified times during the day. Events can also be generated to occur during some future time period. The model can then be designed in a manner that allows for each possible event that is scheduled to occur during a given hour to be checked for at the beginning of each clock hour of a specific day. The simulation is then terminated after a pre-determined number of days, or hours, of operation.

Figure 3 provides the basic logic for a model of the intermodal terminal operation and incorporates the time methodology required for the simulation. It should be noted that several of the branches that eminate from decision blocks in the flowchart initiate sub-programs with related, but independent, actions. For example, should a train be scheduled to arrive with flatcars loaded with containers during the current hour on a given day, they must be placed in the inbound yard. This process is not as straight forward as it might appear. There are, in general, three types of flatcars; those that can carry only over-the-road type trailers, those that can carry only containers without undercarriage, and those that can carry both types of containers. Furthermore, the cars can be fully loaded, partially loaded, or empty. Finally, the cars may be connected in the train such that they are facing the wrong direction (reversed) to be moved without special switching onto certain intermodal sidings. Another feature of the arrival is that trains seldom arrive at the exact scheduled time. Therefore, the model should provide for this time deviation. Figure 4 illustrates one method of incorporating the arrival event logic into the model.

Subroutines for switching, siding loading and unloading, parking lot activity, and departures can be developed in a similar manner. However, the logic employed is ultimately dependent upon the simulation



TRAIN ARRIVAL SUB-ROUTINE

FIGURE 4

language used to model terminal operations. A detailed description of the model, including all subroutines, is presented in Chapter III. The selection of the language for this research project is discussed below.

Selection of the Simulation Language

A major objective of this research is to develop a simulation model that can be used by intermodal managers to evaluate their terminal operations. With this goal being a primary consideration, the model is constructed and documented in a fashion that will enable managers to employ the model with minimum knowledge in computer programming and simulation. To assume no simulation experience on the part of users is, of course, naive. Nevertheless, the model is presented as simply as possible. The complexity of the intermodal terminal system would seem to inhibit the attainment of this goal. If, for example, the simulation were written entirely in FORTRAN, a knowledgeable individual in both the language and simulation would find the model difficult to apply without a great deal of preparation. Fortunately, several simulation languages exist that reduce programming requirements.

Four simulation languages that could be used to model this system are SIMSCRIPT, GASP IV, GPSS, and Q-GERT. The first two are general purpose simulation languages. GPSS is a "block" orientated special purpose language, while Q-GERT is "network" orientated. Each language provides special routines that promote simulation programming efficiency. Of these, Q-GERT appears to possess more features that can be incorporated to accomplish the objective discussed above.

Any simulation language used for the intermodal terminal model will require extensive supplemental FORTRAN programming. Q-GERT, as developed by Pritsker [10 and 12] is a FORTRAN based network orientated language that possesses many routines that automatically handle queues and schedule events that must be written by the user in the other languages, including GASP IV. Q-GERT also provides a great deal of user flexibility by allowing user written input, output, and FORTRAN subroutines. These features, overall, tend to reduce programming requirements.

With the exception of the random number generator, Q-GERT is virtually machine independent, permitting implementation on a wide range of computers. The language's graphical network model documentation procedures enable models to be presented in a way that should be comprehensible to managers with limited simulation experience. Additionally, experience gained by the author with certain characteristics of Q-GERT in other applications indicate that some features lend themselves to the type of model that is desired to be developed [44 and 45]. These features could prove important to this simulation model. For example, it is possible in Q-GERT to set server utilization times for waiting lines such that service in progress will not be completed until the occurrence of some other event in the program. This feature is exactly what is needed in an intermodal terminal model in order to automatically direct events to occur, and activities to be initiated, at specific (scheduled) times. Therefore, the Q-GERT Simulation Language has been selected as the language for the model.

APPENDIX A, describing the special features of Q-GERT that are incorporated into the model, is provided for prospective users of the model. This appendix contains all of the language's symbols and notation that are required to follow the development of the model as presented in the next chapter.

Summary

A description of the essential elements of intermodal terminal operations was presented. A flowchart of the logic required to model the system was also included. Particular interest was focused on the establishment of the methodology used to account for time in the model. An example of the logic developed for associated subroutines was also illustrated.

The selection criteria used to choose the simulation language for the intermodal terminal system was also discussed. The network orientation feature of the Q-GERT language provides the communication considered essential for an adequate understanding of the model by intermodal managers with some simulation background. The Q-GERT symbols and notation required to follow the development of the model are described in APPENDIX A.

CHAPTER III

CONSTRUCTION OF THE SYSTEM MODEL

The discussion that follows first defines the system in terms of its important components and their interrelationships. The formal model logic is then presented. Several of the terms that were introduced in the previous chapters form the basis for this presentation.

Model Specifications

The model specifications for the system to be modeled include definitions in terms of entities (system components), events, attributes, variables, relationships and formal logic. Specifically, the operations of the intermodal terminal operations of a railroad are limited to the interface for containerized freight between rail and over-the-road transport. Therefore, the system under study is defined as the over-the-road/rail operations of a railroad terminal.

System Components

The entities and their attributes associated with the system are described in Table 1.

TABLE 1

Entities and Their Attributes
of an
Intermodal Terminal System

Entities	Attributes
Terminal	In-bound Yard
	Out-bound Yard
	Siding
	Ramp
	Parking Lot
	Distances between in/out-bound yards and siding
	Distance between ramp and parking lot
	Switching schedule
In-bound Yard	Number of flatcars
	Type of flatcars
	Direction of flatcars
	Number of containers/trailers per flatcar
Out-bound Yard	Number of flatcars
	Destination of flatcars

TABLE 1 (continued)

Entities	Attributes
Siding	Capacity Number of flatcars Type of flatcars Number of containers/trailers on flatcars On/offload status
Ramp	Type of container handling equipment
Flatcars	Type Number of containers/trailers Facing direction in yard Destination
Container/Trailer	Type Parking lot status 1. Outbound rail a. Destination b. Number allowed per flatcar 2. Over-the-road pick-up

TABLE 1 (continued)

Entities	Attributes
Parking Lot	Capacity
	Type of containers
	Number of containers
	Containers that arrived by rail
	Containers that arrived over-the-road
	Container destination
	Rail shipping restrictions

The essential components of the Intermodal Terminal Model are the flatcars and container/trailers that are to be processed through the system. The processing is a function of the physical characteristics of the terminal facility, coupled with the type of equipment employed for the process.

Flatcars form queues at the inbound yard, waiting to have containers removed. They are then moved to the siding where they again form queues to be offloaded, onloaded, and switched to the outbound yard. Queues are also formed by flatcars at the outbound yard until they are removed from the system by a scheduled train departure.

Two types of queues are formed by containers at the parking lot. One queue represents containers that have arrived in the terminal by rail that must wait for an over-the-road tractor pick up. The second queue consists of containers that are waiting for rail transshipment.

System Activities

The activities of the system that are represented in the model are described in Table 2. It should be noted that the train arrival and departure activities associated with the system are omitted. The arrival and departure of trains are treated as scheduled events that add flatcars to, or delete flatcars from, the system. As such, scheduled train arrivals and departures perform functions similar to source and sink nodes normally used in more conventional simulation models to generate or remove transactions.

TABLE 2

Activities Associated with an Intermodal Terminal Simulation

Flatcars are switched to the siding

Reversed flatcars in the inbound yard are turned before being switched to the siding

Containers are removed from flatcars and placed in the parking lot

Containers are removed from parking lot and placed on flatcars

Flatcars are switched to the outbound yard

Containers are picked up by tractor at the parking lot

Containers are delivered by tractors to the parking lot

System Events and Associated Activities

Events associated with the systems activities are presented in Table 3. As noted above, two exceptions to listing in Table 3 are the arrival and departure events for the yard. These events add or delete flatcar transactions during the simulation and occur at specified times during a run. Consequently, their associated activity times are not accounted for in the model.

The scheduled arrival of flatcars at the inbound yard (event) sets the conditions necessary for the processing of flatcar transactions through the system. The arriving flatcars enter queues according to type of car and loading configuration. If the siding is empty, flatcars waiting in the inbound yard queues will be switched to the siding (activity) during specified time periods. A user input defined set of priorities determines which cars, if available, will be switched from the inbound queues first and which cars will be switched second. The number of cars switched to the siding is limited to the maximum car capacity of the siding. It is possible to switch empty cars from the inbound yard queues, given that there are no loaded cars available, and there are containers waiting for shipment in the parking lot. Reversed cars that are selected for switching are turned (activity) before being placed on the siding.

Once all the flatcars that are to be switched arrive at the queue representing the siding (event) and the time of day is within specified hours, their associated containers can be offloaded (activity). The containers are moved from the car via the ramp and placed in the parking lot queue. When all of the containers have been removed from

TABLE 3

Events Associated with Activities for an Intermodal Terminal Simulation

---

Flatcars arrive at the inbound yard

Flatcars begin switching movement to siding to be offloaded

Reversed flatcars in the inbound yard are turned before switching to siding

Flatcars arrive at siding

Containers start offloading at siding

Containers arrive at parking lot from ramp

All flatcars at siding are offloaded

Containers arrive parking lot over-the-road

Containers depart parking lot over-the-road

Containers depart parking lot for ramp

Containers begin loading on flatcars

All flatcars to be loaded at siding are loaded

Flatcars are switched from siding

Flatcars arrive at outbound yard

Flatcars depart outbound yard

---

the cars (event), then outbound containers can be onloaded (activity). The outbound containers complete loading (event) and the cars are then ready to be switched to the outbound yard (activity), again during specified hours. The cars arrive at the outbound yard queues according to the destination assigned to their loaded containers (event) and depart the system at the scheduled departure time (event) of trains outbound in predetermined directions. Special handling rules are provided for empty flatcars arriving at the outbound yard queues.

Related activities and events are associated with the parking lot. When a container arrives at the inbound yard (event), a tractor is scheduled to pick up the container after it arrives in the parking lot (activity). The tractor is placed in a queue upon arrival at the parking lot (event), is coupled to a matching container (activity) and the tractor/trailer combination departs the system (activity). Often, arriving tractors also deliver other containers for shipment (event). These containers are also decoupled from the tractors and placed in the parking lot queue (activity) for future rail transportation. Random over-the-road container arrivals (event) can also occur. However, the tractors associated with these arrivals are not entered into the system. Only scheduled tractor arrivals can pick up containers.

A special event and activity that provide the basis for real time scheduling of all the terminal events and activities discussed above is the realization of each clock hour (event) during the simulation and the scheduling of the next clock hour to occur in exactly one simulation time unit (activity). For example, the simulation begins with simulation time set at zero (event), which represents the beginning of the first hour of the first day of the simulation; the beginning of the next hour is scheduled (activity), and simulation time is stopped (event). The daily train arrival and departure schedules are checked and these events are scheduled into the simulation at appropriate times. The status of the entire system is then checked to determine what activities could be scheduled to begin during the hour. These activities if any, are scheduled and simulation time restarted. Simulation time is advanced to the next scheduled event occurrence time and then it is stopped once more. Should the event realized be one that changes the state of the system, the associated activities are scheduled before simulation time is allowed to advance. If, however, the event realized was the occurrence of the next unit of simulation time, the status of the entire system is checked before simulation time is permitted to continue. Arrival and departure train schedules are read into the system at the beginning of each new simulation day. This process continues until a predetermined simulation time is reached. At that point, the simulation is terminated.

A more detailed discussion of events and their associated activities, illustrating the way they mesh into the system, is given in the model logic section.

System Parameters

The system parameters are dependent on the physical characteristics of the terminal in addition to the train arrival and departure schedules. A switch from the inbound yard to the siding, for example, cannot be scheduled unless there are cars in the yard. This condition may be dependent on the scheduled arrival of a train. Furthermore, a switch cannot be made to the siding if there are cars on the siding. Therefore, the scheduled time to switch to the siding is dependent on prior events. However, time to switch, a parameter, is dependent on the distance the siding is located from the yard. The parameters of the model are listed in Table 4.

The probability distributions and their associated parameters listed in Table 4 will not remain the same within the model for simulations of different terminals. This situation is apparent since no two terminals possess exactly the same physical characteristics. This will also be true for different simulations of the same terminal. In this instance, the parameters of the model are consistent from run to run within a simulation, but several of them can be varied from simulation to simulation. For example, the scheduled arrival time of trains will remain the same through several runs of a simulation, but may be changed for a different simulation of the same terminal.

The manner in which the model's time distributions and parameters are provided for in the Q-GERT language is discussed in the model logic section of this chapter. The derivation of the distributions for a simulation of a specific terminal operation is presented in the chapter on implementation.

System Variables

The variables of the system fall into two general classifications; those that are provided by the programmer, and those that are a part of the Q-GERT language. Additionally, there are Q-GERT provided subprograms

TABLE 4

System Parameters for an

Intermodal Terminal Simulation

Parameter Name	Parameter Measure
Train arrival time	Schedule
Switching time from inbound yard to siding	Probability Distribution
Container unloading time from a flatcar (Removing container from flatcar and spotting it in the parking lot)	Probability Distribution
Container spotting time in parking lot (Placing an over-the-road container arrival in the parking lot)	Probability Distribution
Container pick-up time for over-the-road departure	Probability Distribution
Time to reverse flatcars	Probability Distribution
Container loading time on flatcars (Removing container from parking lot and placing it on flatcar)	Probability Distribution
Switching time to outbound yard	Probability Distribution
Train departure time	Schedule

that are used extensively in the supplemental FORTRAN programmed portion of the model. The variables and subprograms required for the model are defined below.

There are approximately one hundred fifteen (115) programmer defined FORTRAN variables incorporated into the user function subroutine of the model. Additionally, some of these variables may be treated as parameters. An illustrative example of these variables is listed in Table 5.

A list of selected Q-GERT variables with their definitions is provided in Table 6. These are variables that can be used directly in the programmer written subroutines of the model.

In Q-GERT, FORTRAN subroutines written for a model are defined as User Functions (UF). Table 7 presents a list and description of the Q-GERT subprograms that are utilized in the FORTRAN subroutines for the intermodal terminal model.

Two additional FORTRAN subprograms are also included in the model. Subroutine UI is employed to initialize the programmer defined variables and/or parameters and to create the initial conditions required for the model. Subroutine UO is used to perform the end-of-run computations and to output the results of the simulation. These subroutines are described in detail in the next chapter.

Model Logic

One desirable feature of the Q-GERT simulation language used to model the intermodal terminal system is that the model's graphical representation is in the format of a network. Thus, one can visualize the

TABLE 5

An Example of Programmer Provided Variables for the Intermodal Terminal Simulation Model in the Q-GERT Simulation Language

Variable	Definition
ACTUAL	Records the value of a random variable generated from a Q-GERT provided distribution. Type of distribution with associated parameter must be specified as program input.
ARR	ARR = TRANA + ACTUAL. Records the time a flatcar is to arrive in the inbound yard.
ATIME(I,J)	I=1, 365, J=1,20. Records a scheduled train arrival time for up to twenty trains per day for 365 days. Enables train flatcar and container composition to be automatically read at specified times during simulation. Initialized at 25.0 unless otherwise specified as train schedule input.
DTIME(I,J)	Essentially the same as ATIME(I,J) except that variable provides simulation with scheduled train departures. Initialized at 25.0 unless otherwise specified.
IDAY	Records current day of the simulation. Initialized at 1.
IDEPT(I,J)	I=1, 365, J=1,20. Records a code that contains the direction of a departing train and the maximum number of flatcars that can be coupled to the train scheduled to depart at time DTIME(I,J). Initialized at 0 unless otherwise specified as train schedule input.
IHOUR	Records current hour of day during the simulation. Initialize at 0.
ISER	Used to identify a specific server, by number, in the model.
ISS	An indicator variable for server status. 1 indicates server is busy, 0 otherwise.
INQ	The current number in a specific queue node.

TABLE 5 (continued)

Variable	Definition
IQ	Used to identify a specific queue node, by number, in the model.
ISWIN	An indicator variable. 1 indicates a switch from the yard to the siding is in progress. 0 otherwise.
ISWITC	An indicator variable. 0 indicates that switching is permissible. 1 indicates switching in progress.
ISWOUT	An indicator variable. 1 indicates a switch from the siding to the yard is in progress. 0 otherwise.
ITP1(I,J)	I=1, 365, J=1,20. Records number of "Trailer only" type flatcars on train scheduled to arrive at time ATIME(I,J). Initialized at 0 unless otherwise specified as train schedule input. A value other than 0 is a code that indicates the number of flatcars with two trailers, one trailer, or no trailers (empty). A provision is also made to identify the number of reversed cars.
ITP2(I,J)	The same as ITP1(I,J) except that "container only" type cars are provided for.
ITP3(I,J)	The same as ITP1(I,J) except that "container/trailer" type cars are provided for.
ITP1E, ITP1W ITP1S, ITP1N	Initialized number of containers in parking lot that require type 1 flatcars and East, West, South or North train departures.
ITP2E, ITP2W ITP2S, ITP2N	Initializes number of containers in parking lot that require type 2 flatcars and East, West, South, or North train departures.
ITP3E, ITP3W ITP3S, ITP3N	Initializes number of containers must use type 3 flatcars and East, West, South, or North train departures.
IWEST, IEAST ISOUTH, INORTH	Initialize number of flatcars in outbound yard waiting for train departure in a specific direction.
LOADOF	An indicator variable. 1 indicates inbound are being offloaded, or available for offloading at the siding. 0 indicates no activity.

TABLE 5 (continued)

Variable	Definition
LOADON	An indicator variable. 1 indicates that all cars at the siding are empty. 0 otherwise.
LOADOU	An indicator variable. 1 indicates outbound cars are loaded, or being loaded at siding. All other siding activity blocked. 0 indicates no activity.
MXPARK	A decision variable. Initialized at the maximum number of containers allowed in parking lot.
MXRAMP	A decision variable. Initialized at the maximum number of flatcars permitted at the siding.
PICUP	Records the value of a random variable generated from a Q-GERT provided distribution. Type of distribution with associated parameters must be specified as program input. This value is used to schedule a container pick-up at the parking lot that is associated with a container arrival in the inbound yard.
TIMESW	A random variable generated from a Q-GERT provided distribution. Type of distribution with associated parameter must be specified as program input. This value is used to schedule the switching activities duration.
TRANA	Reads ATIME(IDAY,M), IDAY=IDAY; M=1,20, into simulation if train scheduled to arrive during current hour.

TABLE 6

Selected List
Q-GERT Defined Variables used for the
Intermodal Terminal Simulation

Variable	Definition
ATT	The values of the entire array of all defined attributes of the current transaction.
ATT(I)	The value of attribute I of the current transaction.
NDE	The number of attributes associated with a transaction.
NREL(NODE)	The current number of transactions in a Q-node.
NRUN	The current run number for a simulation.
NRUNS	Total number of runs for a simulation.
PARAM(I,J)	The Jth value of parameter set I.
TBEG	Time at which data collection is to begin during a simulation.
TNOW	The current time of the simulation.

TABLE 7

Q-GERT Subprograms Used in UF

Subprogram	Definition
Subroutine GETAT(ATT)	The statement CALL GETAT(ATT) returns the attribute vector of the current transaction.
Function ISTUS(IQ,ISER)	The statement ISS=ISTUS(IQ, ISER) returns the status of server ISER associated with queue node IQ.
Subroutine PTIN(IQ,TIME, TNOW, ATT)	The statement CALL PTIN (IQ,TIME,TNOW,ATT) places a transaction in an events file at time TNOW, with attributes ATT, to be placed in queue node IQ at the TIME specified.
Function REMST(ISER)	The statement RTIME=REMST(ISER) returns the service time remaining on server ISER.
Subroutine STAGO(ISER, NODE,TIME,0, ATT)	The statement CALL STAGO(ISER,NODE,TIME,0,ATT) stops service on ISER and places transaction with attributes ATT in NODE at TIME.
Function XNINQ(IQ)	The statement INQ=XNINQ(IQ) returns the current number in queue node IQ at time TNOW.

flow of transactions through the system. Supplemental FORTRAN programming is not eliminated; however, the Q-GERT procedures are such that a user of the model need not be concerned with the detailed FORTRAN programming in order to understand the model's logic and the manner in which it functions. All FORTRAN programming associated with the model is accomplished through the use of User Functions that can be called when specified events occur during the simulation. These user functions are, in reality, programmer written subroutines that enable the modification of conditions in the system at any selected point, or time, during a simulation run. Extensive use is made of this Q-GERT option for the intermodal terminal model.

The model logic and associated supplemental programming are complex. Therefore, in the discussion that follows, the model is divided into subsections to facilitate understanding. The Q-GERT symbols and notation required for the model are defined in APPENDIX A. A complete graphical representation of the network is presented in APPENDIX B.

Q-GERT Network Model (Simplified)

The Q-GERT network model for the intermodal railway terminal requires three supplemental FORTRAN programs in order to operate. These are:

1. User Function (UF)
2. Subroutine UI
3. Subroutine UO

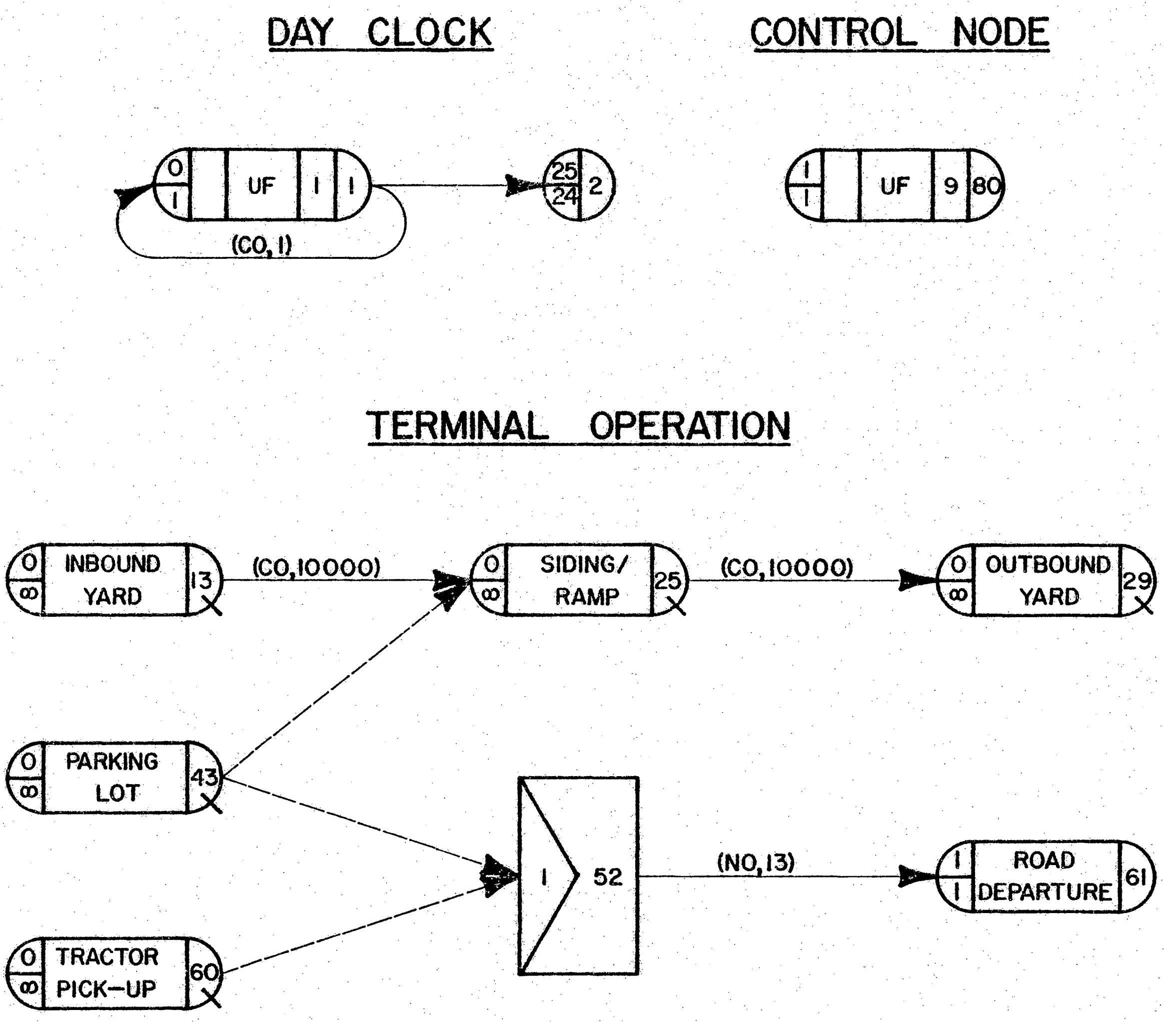
The user functions required for the model contain the logic needed to process transactions through the system. They are called when specific events occur during the simulation. The user functions will be described in sufficient depth in this section to clearly identify the purpose they serve in the model. However, a comprehensive discussion of their programming logic is deferred until after the Q-GERT logic has been developed.

Subroutine UI provides the means to initialize the system at the beginning of a simulation run. This subroutine is also used to read in the train arrival and departure schedules, including the flatcar and container composition of each train, for an entire simulation.

Subroutine UO is used to compute and output the results of a simulation. Q-nodes and their servers are incorporated into the model in a manner that does not permit transactions to flow in the system until they are directed to move when specific events occur. Therefore, the statistics generated for server utilization by the standard Q-GERT output program are not representative of service times in the system. Additionally, parking lot, inbound/outbound container, and inbound yard statistics are not automatically generated by the standard output program. Consequently, pertinent observations are taken during the simulation as changes in the system occur. These observations are then used to compute several of the output statistics for the simulation and to calculate the values of the output variables.

Subroutines UI and UO are discussed in the simulation computer program sections. The Q-GERT network logic for the model is detailed below.

A simplified Q-GERT network of the intermodal terminal model is illustrated in Figure 5. This example identifies the major components



Q-GERT INTERMODAL MODEL (SIMPLIFIED)

FIGURE 5

of the system. They are:

1. The Day Clock
2. A Control Node
3. The Terminal Operation

All of the components are interrelated and dependent on simulation time, yet they must function in a "real time" sense. This is necessary in order to approximate the flow of flatcars and containers through the actual system as closely as possible with the simulated system. Each of these components possess at least one special feature that controls simulation time. For example, the realization of any event that initiates a user function automatically halts simulation time at the point the user function is called. Q-GERT is a discrete next event simulation language; therefore, control of the simulation will not return to the main Q-GERT program, and simulation time advanced, until all of the logic incorporated in the user function has been accomplished.

The day clock is represented by nodes 1 and 2 in Figure 5. Node 1 is a source node. That is, this node does not require an initial incoming transaction to be realized. Current simulation time (TNOW) is set at zero at the start of a simulation. Node 1 is automatically realized at this time, and two transactions emanate from the node. One transaction is scheduled to again realize the node in a constant one unit of simulation time. The second transaction is sent with a zero time delay to node 2. Node 2 is a sink node. A sink node can end a simulation after a predetermined number of realizations. Node 2 requires twenty-five transactions before it is first realized and twenty-four transactions for each realization thereafter. The model's hour clock is then represented by node 1, and one unit of simulation time represents one unit of clock time. The accumulated days of a simulation are recorded at node 2 (the first transaction reaching node 2 at simulation time zero is, in effect, not counted).

User function 1 is also called each time node 1 is realized, and TNOW is stopped until control of the simulation is returned to the main Q-GERT program. This includes time TNOW=0 when node 1 is first realized. When control of the simulation is in user function 1 at the start of each new hour during the simulation, the status of the entire system is checked. Events that can be scheduled to take place during the current hour are identified and related activities scheduled. Activities for future events are also scheduled should the status of the system dictate. For example, train arrival and departure schedules are checked at the beginning of the first hour of each day. If a train is to arrive during the day, transactions with associated attributes are generated for each flatcar in the train and they are entered into the simulation in the inbound yard at the program generated arrival time during the day. Simultaneously, a tractor pick-up is scheduled for the parking lot for each container on each flatcar. These future pick-up events will not occur until at least twenty-four hours have elapsed, and could, in extreme circumstances, take as long as thirty days.

Control node 80 performs a special function within the model. This node is realized at the direction of the logic contained in several of the user functions. The realization of node 80 will pass control of the program to user function 9. The logic in user function 9 then directs the program to specified statement numbers in other user functions. This is accomplished by a call to subroutine PTIN(80,0.0,TNOW, ATT) under specified conditions in a user function. Immediately after the call to subroutine PTIN control is returned to the main Q-GERT program. Any future events scheduled by previous user function logic are placed in the main program files, actions dictated to take place at TNOW accomplished, and a transaction with current attributes scheduled to arrive at node 80 in TNOW + 0.0 time units. In other words, the main Q-GERT program files can be updated based on the current status of the system as recorded in a user function, the current transaction attributes transferred to another point in the user function programs, and simulation time, TNOW, held constant. Therefore, simulation time does not advance in the model until it is directed to advance.

The terminal operation section in Figure 5 illustrates the general flow of flatcar and container transactions through the system. Another example of a special feature of the model logic is also introduced. Nineteen of the twenty-six Q-nodes defined for the system have single servers with service times longer than the duration of any simulation run. In this example, the servers following Q-nodes 13 and 25 have service times of a constant 10,000 time units. This procedure insures that transactions are held in specific Q-nodes until they are directed to move by the occurrence of a designated event during a simulation.

The seven nodes of the terminal operation in Figure 5 represent the major events that occur at an intermodal terminal. The modeling of these events is, of course, more complicated than shown here, but they demonstrate the overall logic of the model. The primary condition that must be met in the system requires that flatcars are waiting at the inbound yard. Containers can be in the parking lot, but this is not necessary. If there are no flatcars, then there is nothing to process. Any of the Q-nodes can be initialized to meet this, or other, conditions at the beginning of a simulation. However, for the purposes of this discussion, it is assumed that all Q-nodes are empty, except the parking lot. It is assumed that there are sufficient outbound containers in the parking lot to load the flatcars at the siding.

The major events of the model are:

1. Flatcars arrive at the inbound yard. Tractor pick-ups are scheduled for their containers, and they wait to be switched to the siding.
2. Flatcars arrive at the siding where they wait to be offloaded.
3. Inbound containers arrive in the parking lot from the siding where they wait for a tractor pick-up.
4. All flatcars are offloaded at the siding and they wait to be onloaded with outbound containers.
5. Tractors arrive at parking lot and pickup preselected containers. They often deliver other containers for outbound shipment.
6. Flatcars are loaded with outbound containers at the siding and wait to be switched to the outbound yard.
7. Flatcars arrive at the outbound yard and wait for scheduled train departures.

The basic model logic required to simulate these events is what is shown in Figure 5.

User function 1 at node 1 is initiated at time TNOW = 0 and all incoming train arrivals are scheduled for the day. As the trains arrive, their flatcars are placed in the inbound yard represented here by Q-node 13. User function 1 is also used to schedule a tractor pick-up for each container on an arriving train at the scheduled arrival time plus container processing time.

Flatcars, now waiting in Q-node 13, are moved to Q-node 25, the siding, when the conditions for switching are met. If there are more cars in the yard than there are positions at the siding, the excess remain in the yard and wait for a future switching activity. Offloading the cars at the siding can commence when the time of day is within the range of normal working hours for the siding crew. Each container on every flatcar is removed and placed into the parking lot, represented in Figure 5 as Q-node 43, as separate transactions. The transactions representing flatcars remain at the siding, but are now empty.

Empty flatcars at the siding set one of the necessary conditions for removing outbound containers from the parking lot and placing them on the cars. Normal crew working hours is the second condition. The onloading of containers begins when both of these conditions are met. The procedure is to first check the outbound containers in the parking lot to see if any match the type of cars that are on the siding. This identifies containers that are eligible for loading. The next step identifies those eligible containers that require special handling.

For example, most over-the-road trailers are loaded in pairs on the flatcars. Occasionally, a trailer has a unique destination that dictates it must be shipped individually on a flatcar. The final step before loading requires that containers being shipped in pairs be matched by destination. The containers are then removed from the parking lot one at a time and placed on matching empty flatcars after the matching process is completed. The onloading continues until all flatcars that can be loaded with containers are loaded. The transactions representing loaded flatcars at the siding are identified by container destination, and they are then ready to be switched to the outbound yard.

The loaded outbound flatcars are switched from the siding to the outbound yard at Q-node 29 when the conditions of the system indicate that a switching can take place. When the siding has been cleared of all flatcars, the next switching from the inbound yard to the siding can be scheduled. The cars wait at Q-node 29 until a scheduled train departure event with a matching destination occurs. The matching transactions are then removed from the outbound yard and depart the system.

Containers that were removed from flatcars at the siding were placed in the parking lot. They wait there until a matching tractor that was scheduled to pick-up the container when it entered the inbound yard arrives at the parking lot via Q-node 60. Matching node 52 joins these transactions together, accounts for pick-up time on the activity following node 52, and the tractor-trailer combination departs the system over-the-road through node 61.

The entire process continues, driven by the daily train arrival and departure schedule, for the number of days terminal operations specified for the simulation. The complete Q-GERT logic for the terminal operation portion of the model is presented below. The logic for the day clock and control node remains unchanged from the previous description.

Assumptions of the Model

The logic for the model is based on the following assumptions about the system:

1. Three types of flatcars are provided for in the model.
    a. Type 1 flatcars. Flatcars that can carry over-the-road type trailers only.
    b. Type 2 flatcars. Flatcars that can carry over-the-road type trailers, or containers without an under carriage.
    c. Type 3 flatcars. Flatcars that can carry containers without under carriage only.

2. Three types of containers are provided for in the model.
    a. Type 1 containers. Over-the-road type of trailers.
    b. Type 2 containers. Containers that can be equipped with an under carriage for over-the-road transport.
    c. Type 3 containers. Containers that can not be equipped with an under carriage.

3. All containers are assumed to be forty feet long.

4. Each flatcar can be empty, or loaded with one or two containers.

5. Each switching of flatcars from the inbound yard to the siding will consist of the maximum number of flatcars permitted at the siding, given that loaded flatcars are available in the yard. Empty flatcars will be switched if there are containers in the parking lot and suitable cars are available in the yard.

6. All arriving loaded flatcars at the siding will be offloaded.

7. All empty flatcars at the siding will be loaded, given there are suitable containers in the parking lot.

8. All flatcars at the siding will be switched to the outbound yard at the scheduled switching time. Empties will either be shipped out of the terminal, or returned to the inbound yard.

9. Each container arriving at the inbound yard will generate a scheduled pick-up at the parking lot. These scheduled pick-ups can also generate containers for rail shipment. However, random container arrivals for rail shipment can not have a container pick-up associated with them.

Model Transactions and Attributes

There are four types of transactions incorporated within the model. Three of these are tracked through the system. They are:

1. Flatcars

2. Containers
3. Tractors

The fourth transaction type is used to operate the day clock as previously described. The attributes associated with the flatcar, container and tractor transactions are as follows:

1. Attribute 1 is used to designate the type of flatcar, container, or tractor a transaction represents. The possible values of this attribute are:

   A1=1 Type 1 container, flatcar, or tractor
   A1=2 Type 2 container, flatcar, or tractor
   A1=3 Type 3 container, flatcar, or tractor

2. Attribute 2 indicates the loading of an arriving flatcar.

   A2=0 The flatcar is empty
   A2=1 There is one container on the flatcar
   A2=2 There are two containers on the flatcar

3. Attribute 3 indicates the direction of an arriving flatcar on the train.

   A3=1 The car is facing in the proper direction for the siding
   A3=0 The car is reversed

4. Attribute 4 indicates the outbound direction (destination) of a container in the parking lot. This value is assigned to the associated flatcar when container is loaded for shipment.

A4=1 East Bound

A4=2 West Bound

A4=3 North Bound

A4=4 South Bound

A4=5 Flatcar is empty

5. Attribute 5 designates the number of containers to place on a flatcar. Some containers have a unique destination and must be shipped individually. Most containers, however, are shipped in pairs.

   A5=1 One container per flatcar

   A5=2 Two containers per flatcar

Attributes 6 and 7 are also used in the model. However, their purpose is to pass information within the user functions. Their use will be illustrated in the user function section.

User Input Requirements

User provided input that is required for the operation of the model is:

1. Time duration to be simulated in days or hours.
2. Train arrival schedule and composition for period to be simulated.
3. Train departure schedule and capacity for period to be simulated.
4. Initial yard and parking lot conditions at beginning of simulation.

5. Transient time distributions from yard to siding and siding to yard for terminal operation to be simulated.
6. Service time distributions for container offloading and container onloading at the siding and parking lot for terminal operation to be simulated.
7. Container pick-up time distribution at parking lot for arriving containers.
8. Terminal ramp activity working time of day rules.
9. Terminal switching activity time of day rules.
10. Weekend work rules.
11. Selection priority by type of container at the inbound yard for switching to the siding. Priorities one and two must be designated. The options are as follows:
    a. Type 1 container.
    b. Type 2 container.
    c. Type 3 containers.
    d. The containers waiting longest in the yard.
    e. The type of containers with the largest number in the yard.

For example, priority one could be for type 1 containers and priority 2 could be for the type of container waiting the longest in the yard. Any type 1 containers in the yard would take switching precedence. If there were no type 1 containers, then priority would shift to the longest waiting. The method employed for reading in the required input for a simulation is outlined in the implementation chapter.

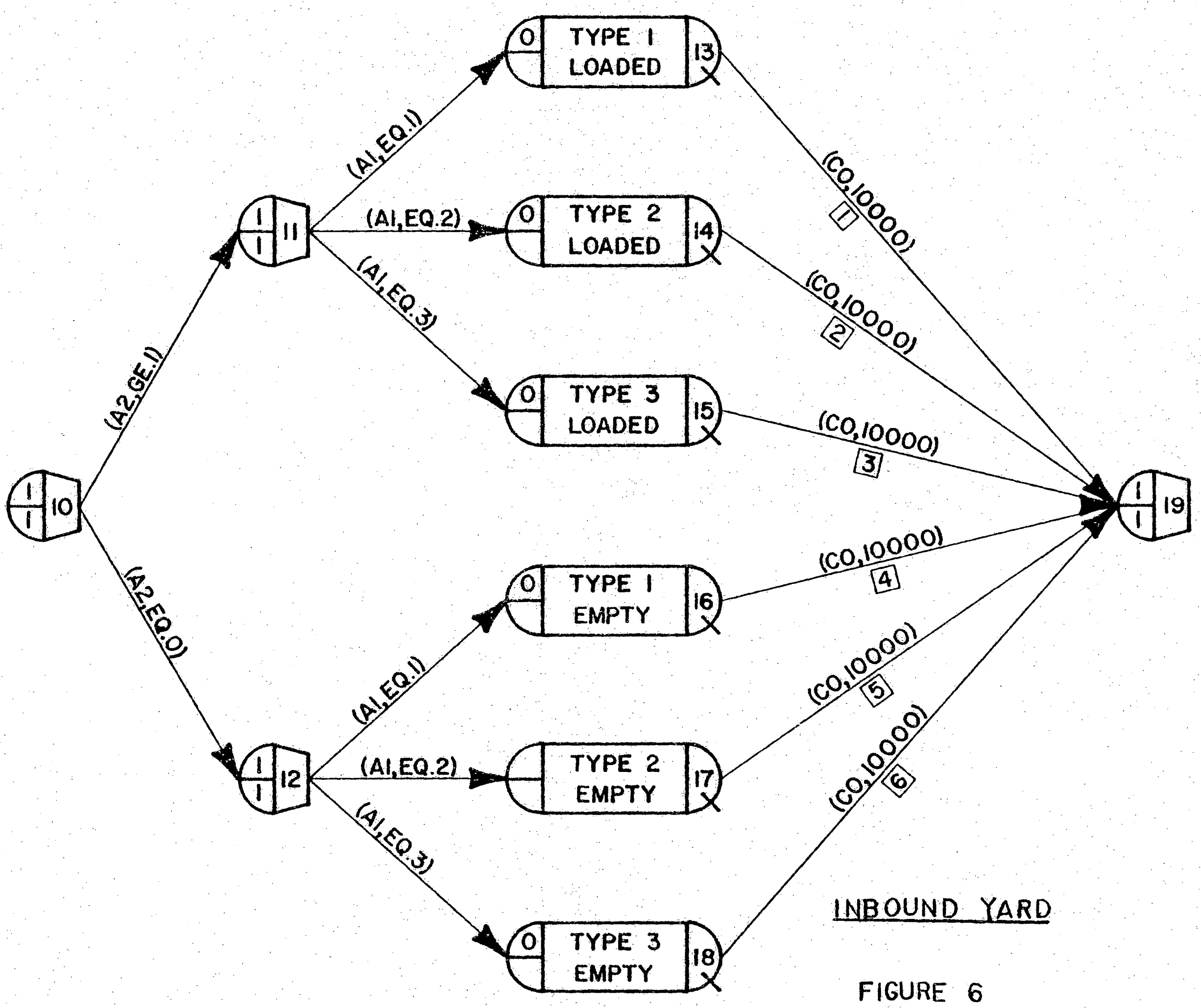
The Inbound Yard

The Q-GERT logic for the inbound yard is given in Figure 6. This section of the model consists of three conditional branching nodes and six Q-nodes. Node 19 is the input node for the siding. Its use is defined in the next section.

Nodes 10, 11, and 12 provide the logic for classifying incoming flatcars in the yard. A scheduled train arrival is read into the simulation in user function 1. This user function decodes the train's flatcar composition. The information generated by this process details the number and type of flatcars in the train, the number of containers per flatcar, and the direction the flatcars are facing in the train relative to the terminal's siding. These data are then used to generate a single transaction for each flatcar with attributes describing the car's characteristics. Each transaction is then scheduled to realize node 10 at the train's scheduled arrival time.

Attribute 2 is used to indicate the loading of an arriving flatcar. Transactions arriving at node 10 are routed to node 11 or 12 based on the value of this attribute. If the car carries one or more containers it is sent to node 11. It is sent to node 12 if it is empty. Nodes 11 and 12 perform identical functions. They are used to branch the cars according to type. Transactions arriving at node 11 are placed in Q-node 13, 14, or 15 based on the value of attribute 1. Node 12 provides the same logic for Q-nodes 16, 17, and 18.

The first transaction arriving at any of the Q-nodes passes directly through the node and initiates the node's corresponding service



INBOUND YARD

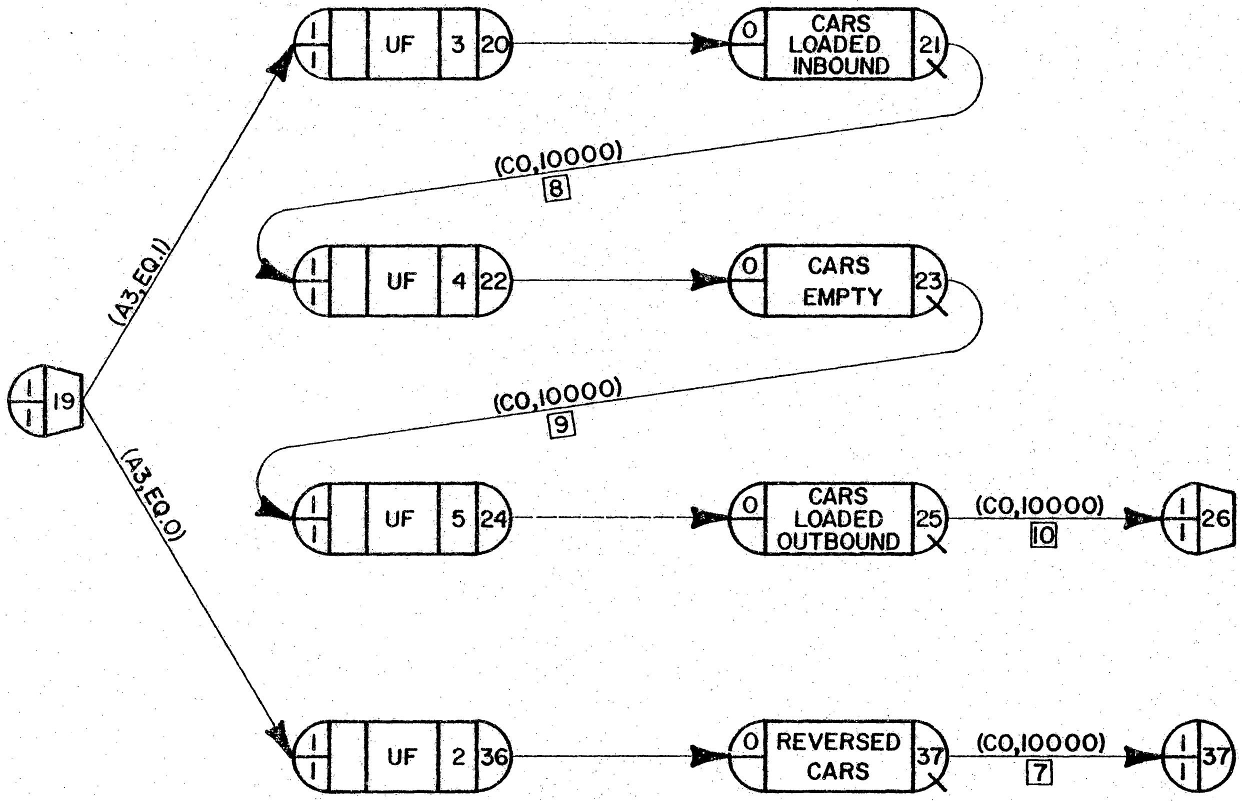
FIGURE 6

activity. Activity time for each of these is set at a constant 10,000 time units. This period of time is beyond that of total simulation time. Consequently, node 19 will never be realized by any transaction on the preceding service activities unless some action external to the network is provided. In addition, since the first transaction in a Q-node is placed immediately in service on the activity following the node, all other like transactions that enter behind the initial transaction are held in the Q-node.

The combination of decoding the composition of an arriving train to generate a transaction with the proper attributes for each flatcar in user function 1, and the branching logic employed at nodes 10, 11, and 12 effectively models the inbound yard classification function associated with rail terminals. The extended service times associated with the activities following each Q-node of the yard insures that all flatcars will remain in the yard until they are specifically called for at the siding. Therefore, the Q-GERT logic used to model the inbound yard adequately represents the functions of the yard in an actual system.

The Siding/Ramp

The Q-GERT logic for the siding/ramp operation is illustrated in Figure 7. Four user functions are required in this section of the model to facilitate the unloading and loading of flatcars and to switch the cars from the inbound yard to the siding. There are four network nodes that call the user functions, one regular node, a conditional branching node, and four Q-nodes required to model the siding/ramp.



<u>SIDING / RAMP</u>

FIGURE 7

The function of node 38 is to fulfill the Q-GERT requirement that every activity must have a starting and an ending node. This node is the ending node for activity 7. Node 26 is the input node for the outbound yard and its use is disucssed under that section.

It was stated in the previous section that node 19 could not be realized unless there was an action taken that was not a part of the Q-GERT network. This external action is dependent on two conditions. The first is the status of the siding. The siding has a finite capacity for the number of cars that can be on the siding. If it is occupied, then no other cars can be switched to the siding. The second condition is the time of day that switchings are allowed to occur at a terminal. For some terminals, there are no time limitations; for others, switchings can only occur during certain hours. These conditions can be specified for each simulation. Each time user function 1 is called during a simulation the status of the siding is checked. If it is unoccupied, then the time of day is checked to see if a switching can be scheduled. The discussion that follows assumes that these conditions have been met and that a switch to the siding is to be scheduled.

The first step in the switching process is to check the inbound yard for container loaded flatcars. If there are none, the yard is checked for empty cars. If there are no cars in the yard, no switching can occur. Assuming that there are empty flatcars only in the yard, the parking lot is checked for matching outbound containers. Empty cars are then switched to the siding as long as there are matching containers and there is room at the siding. Empty cars can not be switched to the siding if these conditions are not met.

Given that there are loaded flatcars at the inbound yard, the next step is to check the priorities for selecting cars, by container type, in the inbound yard that are to be switched. These priorities are read into the simulation as part of the input data. Two priorities must be designated. If no cars are in the yard that match these priorities, then a third priority is established by default. For example, assume that priority one is set for Type 1 containers and that priority two is set for Type 2 containers and that the siding capacity is ten cars. Assume further that there are two Type 1, seven Type 2, and five Type 3 loaded cars in the yard. The priority rules would insure that the two Type 1 cars would be switched first, then the seven Type 2 cars would follow. This movement would leave a vacancy at the siding. Therefore, one of the Type 3 cars would be switched to the siding. The inbound yard would contain only four loaded Type 3 cars after the switching, but this status would change should a train arrive before the next switching could occur.

The actual switching process is accomplished automatically once the first car to be switched is identified. For the purpose of this illustration assume that switching priority one is for Type 1 flatcars, the siding capacity is ten cars, and that there are twenty loaded Type 1 cars in the inbound yard. All of the cars are facing in the proper direction for the siding. Referring back to Figure 6, the status of the server on activity 1 would be busy and there would be nineteen transactions in Q-node 13. The status of the server associated with activity 1 would be checked in user function 1, indicating that there is at least one Type 1 loaded flatcar at the inbound yard. A time to switch (TIMESW) would be calculated, and the transaction currently on activity 1 would be removed through a call to subroutine STAGO(1,19,TIMESW,0,ATT). Since the server at activity 1 is now idle and there are transactions in Q-node 13, the server is immediately made busy by the next transaction in the queue. The call to subroutine STAGO, with related arguments, causes an event to be scheduled to occur. Simply stated, the transaction, complete with assigned attributes, that was on activity 1 is scheduled to realize node 19 at time TNOW + TIMESW. The events that follow are illustrated in Figure 7.

The major concentration of activity for an intermodal terminal is centered around the siding/ramp area. In the model, this activity begins when node 19 is realized. Attribute 3, the facing direction of the flatcar, for the arriving transaction is first tested. If the car is reversed, the transaction is routed through node 36, where user function 2 is called, and placed in Q-node 37. Q-node 37 is, in effect, an extension of the inbound yard and any car in this queue will be the first moved to the siding if a switching is in progress. In this example, the transaction representing the first car arriving at node 19 is facing in the proper direction for the siding. It is routed through node 20, where user function 3 is called, passes through Q-node 21 and causes the server on activity 8 to be made busy.

Control of inbound switching is now in user function 3. TNOW was halted when this function was called and the capacity of the siding was decreased by one for the transaction that is now on the siding. The status of activity 7 behind Q-node 37 is first checked in user function 3. If the server is busy, a previously reversed car has been turned and can now be placed on the siding. The service on activity 7 is stopped, a corresponding transaction inserted at node 20, keeping control in user function 3, and the transaction is routed into Q-node 21. If there are no turned cars waiting at Q-node 37, then the status of the server on activity 1 is checked. An idle server indicates there are no flatcars in the yard and the switching activity is stopped. In this example, the next transaction on activity 1 behind Q-node 13 is placed in node 19 with zero time delay. This process continues, reducing the siding capacity by one for each transaction passing through node 20, until the siding is fully occupied, indicating that the switching activity is complete.

Program control is passed to user function 1 at the completion of switching and the status of the system is checked to determine if an offloading activity can begin before control is returned to the main Q-GERT program. This activity can start if the switching is completed during normal working hours. A provision is provided for overtime work, but this must be stipulated prior to the start of the simulation. TNOW is allowed to increment when control is returned to the main program. User function 1 is called at the beginning of each hour. When normal working hours are indicated, the container offloading activity is initiated.

The container offloading process is started in user function 1 by removing the transaction on activity 8 and placing it in node 22.

This is accomplished by generating a time to begin the offload (TIMEOF) and calling subroutine STAGO(8,22,TIMEOF,0,ATT). This routes the transaction through node 22, calling user function 4, and through Q-node 23 to activity 9 at time TNOW + TIMEOF. The attributes of the transaction are first accessed in user function 4 by a call to subroutine GETAT(ATT). The parking lot queue a container is to be placed in, labeled INQ, is identified by the value of attribute 1, and the number of containers on the flatcar is determined by the value of attribute 2. The current status of the parking lot is then checked. If it is full, the container to be offloaded is placed in the overflow parking queue. If not, the capacity of the parking lot is reduced by one, unloading time (UNTIME) is generated, and each container is placed in the parking area by a call to subroutine PTIN(INQ,UNTIME,TNOW,ATT).

The call to subroutine PTIN generates a new transaction to be placed into Q-node INQ in TNOW + UNTIME time units each time it is made. The transaction's simulation mark time will be TNOW, and its attributes will be the same as those associated with the transaction that initiated the call to user function 4. There will be as many calls to subroutine PTIN as the value of attribute 2 indicates. It follows that if an empty car is indicated and attribute 2 equals zero, this call will not be made. When all containers on a particular flatcar have been offloaded, the status of the server on activity 8 is checked. If the server is busy, a call to STAGO(8,22,0.0,0,ATT) is scheduled and control is returned to the main program. This call reinitiates user function 4 at simulation time TNOW and the unloading process continues. When the server status check for activity 8 indicates that the server is idle, all flatcars have been offloaded and currently reside in Q-node 23. The loading of containers on flatcars can now begin when the enabling conditions of the system are properly set.

The initial system condition requirements for the start of a container onloading process is similar to those described for the off-load. The time must be during normal working hours and the offload must be complete. An onload time (TIMEON) is generated and the transaction in service on activity 9 behind Q-node 23 is removed and placed into node 24 through a call to subroutine STAGE(9,24,TIMEON,0,ATT). A call is made to user function 5 at node 24, but a transaction is not automatically routed to Q-node 25. At this point the model's logic becomes more complex. The logic required must insure that specific type containers be loaded on specific type flatcars. Additionally, when two containers are loaded on one flatcar, which is usually the case, the containers must also be matched by a common destination. If there are no suitable containers in the parking lot for the flatcar currently being processed, then the car must be switched to the outbound yard empty. The rail departure section of the parking lot is intricately involved in this process. Reference should be made to Figure 9, Q-nodes 43, 44, and 45, with the corresponding activities 11, 12, and 13 for the following discussion.

User function 5 is called when node 24 is realized at the start of the onloading activity. TNOW is halted and the attributes of the current transaction are obtained through a call to subroutine GETAT(ATT). A call to subroutine PTIN(80,0.0,TNOW,ATT) is then made and control is returned to the main program. This procedure accomplishes the following desired actions:

1. Control node 80, as previously described, is realized after the system is updated, but TNOW does not advance. This insures that the next transaction residing in Q-node 23 moves on to activity 9 and that the model's file is updated.
2. The transaction that initiated the loading activity is destroyed, but the transaction attributes are retained.
3. Control is retained in user function 5.

The value of attribute 1, the type of car, is then compared with the server status of a corresponding type of container in the parking lot. If the server is idle, then there are no containers of the type required for the flatcar in the lot. Attribute 4, the destination of the car, is then set equal to 5. This indicates that the flatcar is empty. Subroutine PTIN(25,0.0,TNOW,ATT) is called to place an empty car, which is now a new transaction, in the outbound Q-node 25. The status of the server on activity 9 is checked. If the server is busy, the transaction is removed, placed in node 24 and the process continues. If the server on activity 9 is idle, then there are no more cars to onload. Control returns to the main Q-GERT program and TNOW is allowed to advance. Should the status of a server in the parking lot for a matching container and flatcar be busy, then it is possible to load the car.

Loading a container on a matching flatcar requires that the special handling characteristics of the container be checked. This information is recorded as the value of attribute 5 for the container. The server number of a selected container is recorded as ITYPE. A call is then made to subroutine STAGO(ITYPE,80,0.0,0,ATT). This removes the transaction from activity ITYPE and places it in control node 80 as the current simulation transaction with zero units time delay. A call is then made to subroutine GETAT(ATT) and the attributes of the transaction recorded. If attribute 5 has a value of 1, then the container must be loaded by itself on the flatcar. The current capacity of the parking lot is increased by one, the time to onload the container (OLTIME) is generated, and a call is made to subroutine PTIN(25,OLTIME,TNOW,ATT). This generates a new transaction at Q-node 25, with the value of attribute 4, the containers destination, the same as it was for the container. The status of the server on activity 9 is then checked and the onloading continues if the server is busy.

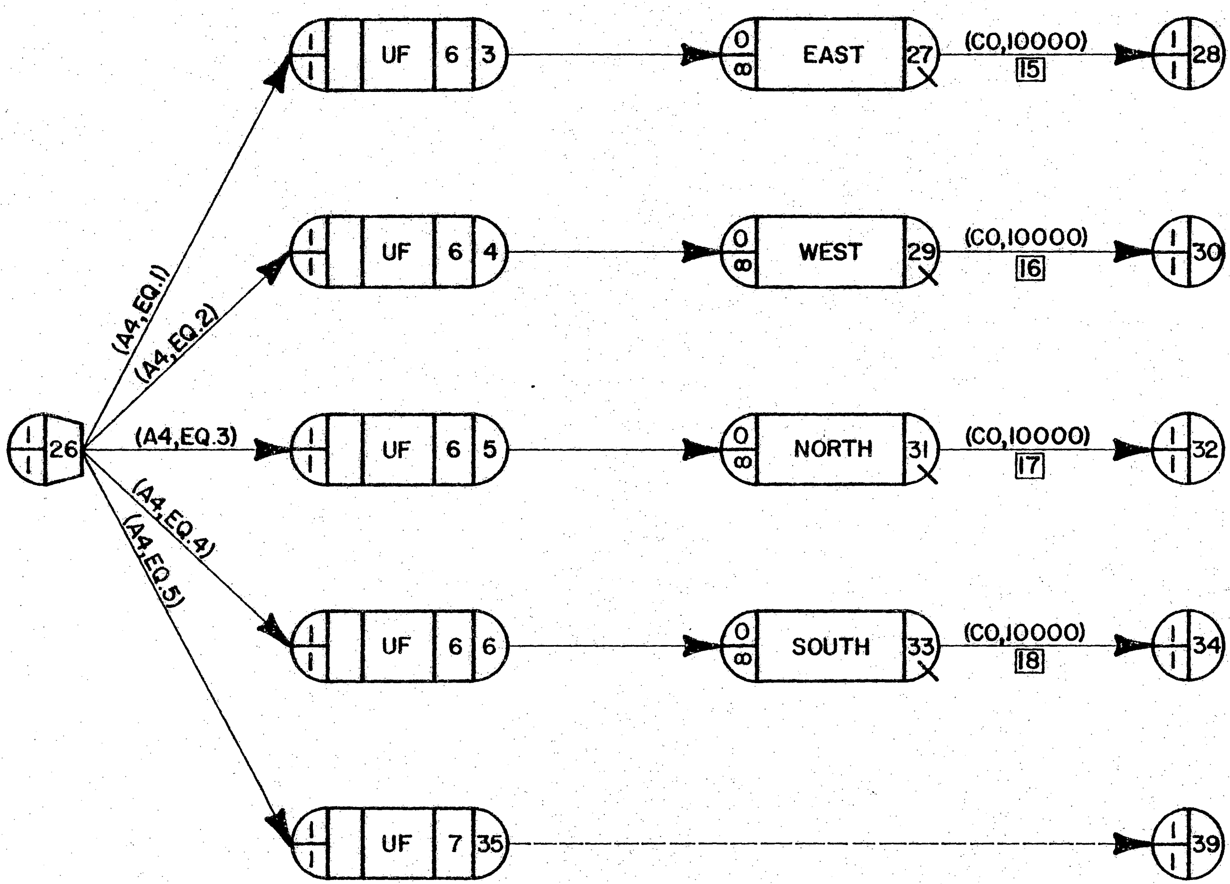
Two container matches must occur for units to be shipped together on the same flatcar. First, they must have the same destination. The second condition is that there can be no special handling characteristics for either container. This is accomplished by recording the values of attributes 4 and 5 of the current container to be loaded. These attribute values are then compared to the corresponding attribute values of each of the other preselected containers in the parking lot. If no match occurs, the original container is loaded individually as previously described. When a match does occur, the current parking lot capacity is increased by two and a transaction placed in Q-node 25.

The siding is checked for additional cars to be loaded and returns control to the main program when the loading activity has been completed. The flatcars can now be switched to the siding at any time the switching rules of the system will allow.

The Outbound Yard

The Q-GERT logic for the outbound yard is illustrated in Figure 8. There is one conditional branching node, five network nodes, four Q-nodes and five statistics nodes. The outbound switching conditions of the system are checked in user function 1. The time to switch to the outbound yard (TIMEOU) is generated and a call is made to subroutine STAGO(10,26,TIMEOU,0,ATT). The transaction on activity 10 is placed into node 26 at time TNOW + TIMEOU. This transaction is routed according to the value of attribute 4, the destination attribute. If the routing is to any node,but node 35, the car is loaded and user function 6 is called. The transaction proceeds to the appropriate outbound queue while the next transaction is removed from activity 10 and placed in node 26 through user function 6. This process continues until all cars have been switched from the siding.

An empty car is routed from node 26 to node 35 where user function 7 is called. The maximum desirable number of empty cars in the inbound yard is checked by type in this function. If the number of empty cars in the inbound yard of this type is below those specified, the car is routed to the proper Q-node at the inbound yard. Should there be more empty cars in the system than desired, the car is shipped out of the system via statistics node 39 and program control is shifted to user function 6.



OUTBOUND YARD

FIGURE 8

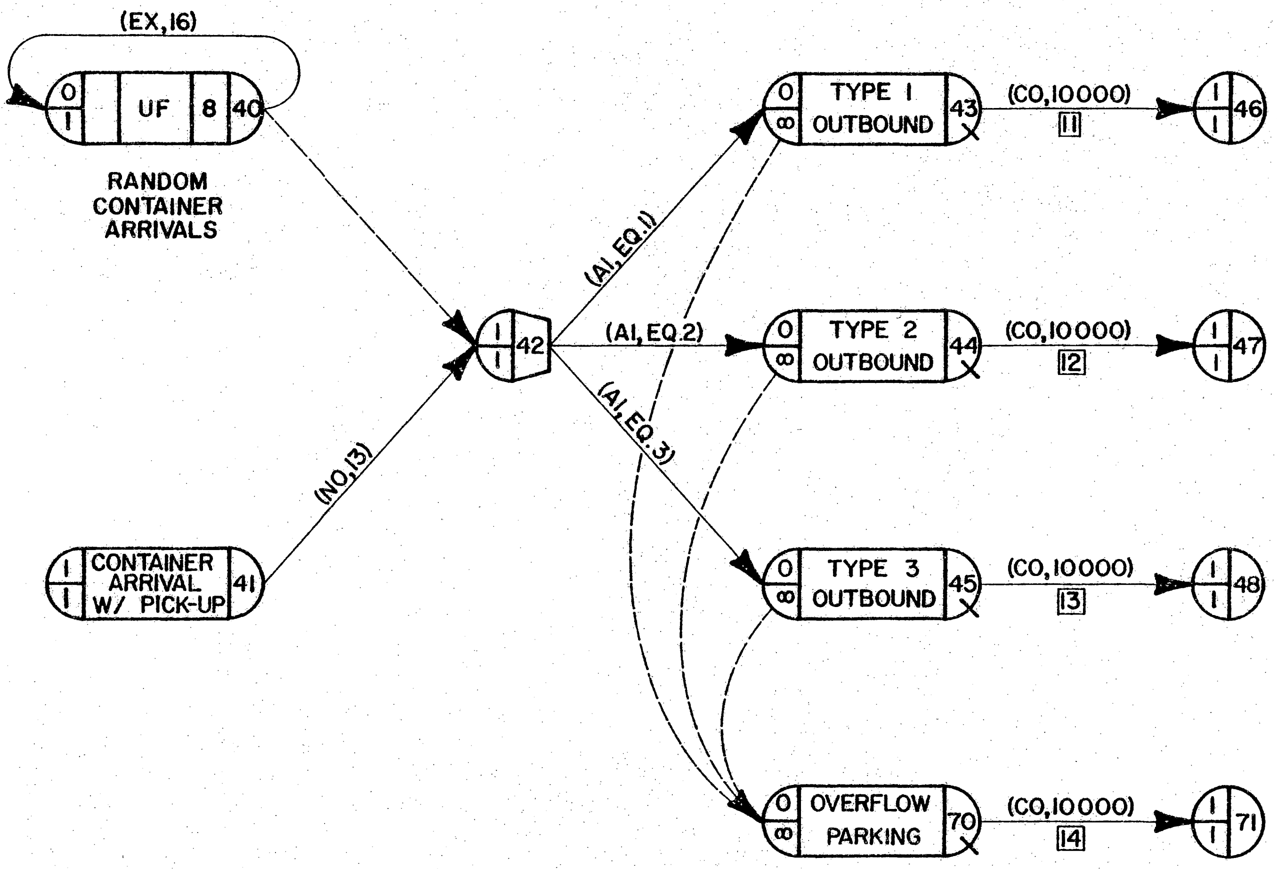
The occurrence of a scheduled train departure event will remove up to the maximum number of flatcars allowed on a train from the system for a given destination. The transactions depart the system through statistics nodes 28, 30, 32, and 34. Any cars in the outbound yard at the beginning of a day will be automatically classified for a scheduled departure. All matching cars arriving in the yard before departure time are eligible for the train, given that space exists. This feature, coupled with the branching activities at node 26, provides the outbound rail classification feature of the model.

The above discussion of the inbound yard, the siding/ramp, and the outbound yard completes the Q-GERT logic for the way flatcar transactions flow through the system in simulated time. Rail container departures were a necessary part of the logic in that they are used to generate the transactions that represent the flatcars in the outbound section of the siding. The discussion that follows will describe the way container transactions are accounted for in the system. These transactions are confined to the parking lot section of the model. It should be noted that the parking lot is one continuous area contiguous to the ramp where containers are mingled, without regard to method of arrival or departure. In this model, however, the parking lot is divided into two sections. One section is for containers that will depart the parking lot by rail and are referred to as outbound containers. The other section is for inbound containers that have arrived at the parking lot by rail and will depart the system via the road.

The Container Parking Lot, Rail Departures

Figure 9 illustrates the rail departure section of the container parking lot consisting of nine nodes, plus Q-node 70 which is shared with the road departure section for overflow container parking. Nodes 46, 47, 48, and 71 are incorporated to meet the Q-GERT requirement for nodes following activities. It should be noted that the parking Q-nodes 43, 44, and 45 balk to Q-node 70, yet they all have infinite capacities defined. The same situation is true for the road departure section of the lot. A total of six Q-nodes, not including the overflow Q-node, are required to model the parking lot. Each of these nodes could, conceivably, contain the maximum number of containers permitted in the lot and if the Q-nodes were defined in this way, the model's parking lot could easily operate over capacity. To divide the parking capacity among the Q-nodes would also be unrealistic. The method used to resolve this dilemma uses a variable labeled MXPARK. MXPARK is set equal to the maximum capacity of the parking lot on initialization. During the simulation, it is decremented by one each time a container is placed in the lot and incremented by one for each container that leaves the lot. This includes those containers that are placed in the lot on initialization. Therefore, MXPARK records the current remaining capacity of the parking lot. MXPARK is tested before each container is placed in the parking lot. If the value is positive, the container is allowed to enter its proper Q-node. If the value is zero, there are no parking spaces and the container is placed in the overflow lot.

The overflow lot performs a special function. It is used to



CONTAINER PARKING LOT (RAIL DEPATURES)

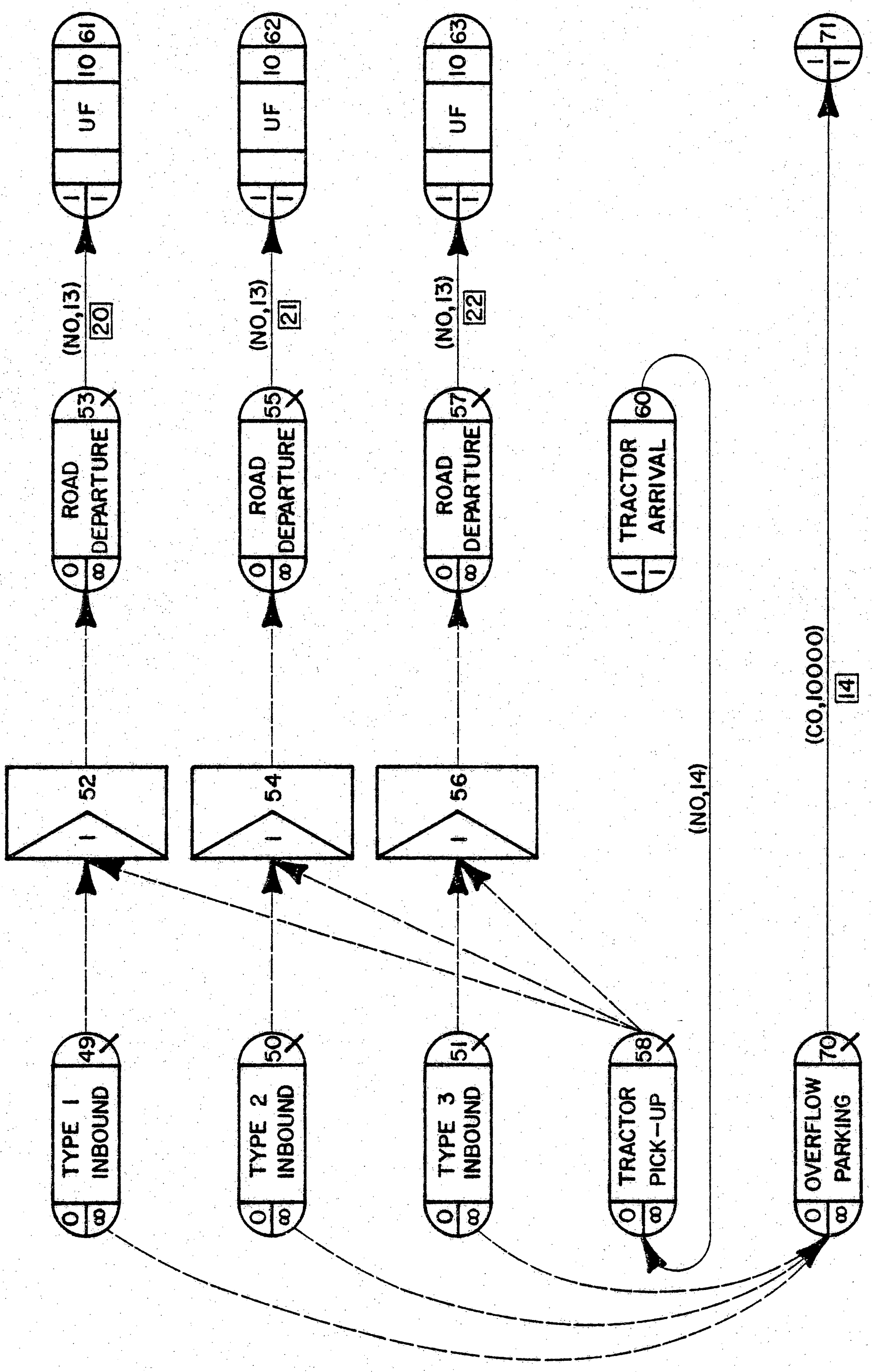
FIGURE 9

collect statistics on the adequacy of the parking lot for each simulation. Q-node 70 is checked at the end of each day. If there are no transactions in the queue, the observation and the value of MXPARK are recorded. The total number in the parking lot is computed. If there are transactions in the queue, the number is recorded, then MXPARK is tested to see if any can be moved to the regular parking lot. Those transactions that can be moved are placed in the lot and the residual, if any, remain in overflow parking for the following day's operations. These daily observations are used to compute some of the output statistics. These are discussed in detail in the chapter on implementation.

The remaining operation of this section of the parking lot is rather straight forward. Containers arrive from two sources. Those entering at node 41 were identified and scheduled when the containers entered the inbound yard. The random container arrivals are generated at node 40 where user function 8 is used to determine their type, destination, and any special handling requirements. They are then entered at node 42 by a call to subroutine PTIN. At node 42 the containers branch according to type and enter their particular queues. They are removed from Q-nodes 43, 44, and 45 in accordance with the procedures described for loading containers on flatcars in the siding/ramp section.

Container Parking Lot, Road Departures

The road departure section of the parking lot is presented in Figure 10. There are two types of transaction associated with this section of the model. The container transactions are placed into Q-nodes 49, 50, and 51 during the flatcar offloading cycle. The way these



CONTAINER PARKING LOT (ROAD DEPARTURES)

FIGURE 10

transactions are generated was described in the offloading portion of the discussion of the siding/ramp operation. Matching tractor transactions were generated for the containers transaction as the flatcars, loaded with containers, were first placed in the inbound yard. The tractor transactions are scheduled to arrive at node 60 with a sufficient time delay to permit the processing of the container transactions through the system. That is, to move from the inbound yard to the parking lot. The value of attribute 1, for all types of transaction in the model, designates the type of containers that are permitted to be associated with the transaction. This discussion will illustrate how the tractors are mated to the containers, and are then removed from the system. The operation of balking feature from the container parking queues to the overflow Q-node 70 was discussed in the previous section.

Inbound container transactions are placed into Q-nodes 49, 50, and 51 based on the value of attribute 1 for the flatcar transaction from which they were generated. For example, assuming a transaction representing a type 1 flatcar at the inbound siding with the value of attribute 2 equal to two. Two transactions would then have been placed in Q-node 49 when the car was offloaded. The value of attribute 1 for each of these transactions would be set equal to one. Two tractor arrivals, also with attribute 1 values of one were scheduled to arrive, and will eventually pick up the two containers. Ordinarily, but not necessarily, the containers arrive at the parking lot first. The tractors then begin to arrive independently at node 60. The tractors are then routed to Q-node 58, with a time delay to allow processing by the terminal personnel, where they wait to be matched with a corresponding container.

The use of match nodes 52, 54, and 56 provide the required matching function. The Q-GERT logic of matching nodes is such that transactions in queues on their input side can be combined, based on attribute value, and routed as a single transaction to a node on their output side. A tractor arrives at Q-node 58, and the value of attribute 1 is one. A container residing in Q-node 49 will also have an attribute 1 value of one. Match node 52 will recognize this condition, remove each transaction from its respective Q-node, and place a single corresponding transaction in Q-node 53. If the server on activity 20, following Q-node 53, is idle, then the departing tractor-trailer combination will be processed out of the system through node 61. The purpose of user function 10 at node 61 is to increment the current capacity of the parking lot, MXPARK, by one. The processing of type 2 and type 3 containers is performed in the same manner, with the value of attribute 1 set at two and three respectively.

Data Input for the Intermodal Terminal Q-GERT Model

The Q-GERT defined data input required to program the model is listed in Figure 11. There are a total of one hundred forty-six (146) data statements used to represent the logic as described in the previous section.

User Function Logic

The logic employed to schedule and coordinate the flow of transactions through the intermodal terminal Q-GERT network model requires

```
GEN,HAMMESFAHR,INTMOD YARD,2,18,1981,13,1,64,1511.99,100,,,7*
SOU,1,0,1,*                   DAYCLOCK
VAS,1,1,UF,1*                 CHECK SCHEDULE
ACT,1,1,CO,1.*                ONE HOUR OF TIME ON CLOCK
ACT,1,2*
SIN,2/END SIM,25,24*
REG,10,1,1,A*
ACT,10,11,(9)A2.GE.1*
ACT,10,12,(9)A2.EQ.0*
REG,11,1,1,A*
ACT,11,13,(9)A1.EQ.1*
ACT,11,14,(9)A1.EQ.2*
ACT,11,15,(9)A1.EQ.3*
REG,12,1,1,A*
ACT,12,16,(9)A1.EQ.1*
ACT,12,17,(9)A1.EQ.2*
ACT,12,18,(9)A1.EQ.3*
QUE,13/YARD1L*
ACT,13,19,CO,10000,1/IN T1 L*
QUE,14/YARD2L*
ACT,14,19,CO,10000,2/IN T2 L*
QUE,15/YARD 3L*
ACT,15,19,CO,10000,3/IN T3 L*
QUE,16/YARD 1E*
ACT,16,19,CO,10000,4/IN T1 E*
QUE,17/YARD 2E*
ACT,17,19,CO,10000,5/IN T2 E*
QUE,18/YARD 3E*
ACT,18,19,CO,10000,6/IN T3 E*
REG,19,1,1,A*
ACT,19,20,(9)A3.EQ.1*
ACT,19,36,(9)A3.EQ.0*
REG,36,1,1*                   CAR IS REVERSED
VAS,36,6,UF,2*
ACT,36,37,NO,10*
QUE,37/REV CARS*
ACT,37,38,CO,10000,7/TURN CAR*
STA,38,1,1*
REG,20,1,1*                   INBOUND RAMP SWITCH
VAS,20,6,UF,3*
ACT,20,21*
QUE,21/UNLOAD*                INBOUND CARS AT RAMP
ACT,21,22,CO,10000,8/CARS IN*
REG,22,1,1*
VAS,22,6,UF,4*
ACT,22,23*
QUE,23/EMPTY*                 EMPTY CARS AT RAMP
ACT,23,24,CO,10000,9/CARSEMT*
REG,24,1,1*
VAS,24,6,UF,5*
QUE,25/ONLOAD*                OUTBOUND CARS AT RAMP
ACT,25,26,CO,10000,10/CARS OUT*
REG,26,1,1,A*
ACT,26,3,(9)A4.EQ.1*
ACT,26,4,(9)A4.EQ.2*
ACT,26,5,(9)A4.EQ.3*
ACT,26,6,(9)A4.EQ.4*
```

FIGURE 11

Data Input for the Intermodal

Terminal Q-GERT Model

```
REG,3,1,1*
VAS,3,6,UF,6*
REG,4,1,1*
VAS,4,6,UF,6*
REG,5,1,1*
VAS,5,6,UF,6*
REG,6,1,1*
VAS,6,6,UF,6*
ACT,3,27,(6)31*
ACT,4,29,(6)32*
ACT,5,31,(6)33*
ACT,6,33,(6)34*
ACT,26,35,(9)A4.EQ.5*
QUE,27/EAST*
QUE,29/WEST*
QUE,31/NORTH*
QUE,33/SOUTH*
REG,35,1,1*
VAS,35,6,UF,7*
ACT,27,28,CO,10000,15/EAST*
ACT,29,30,CO,10000,16/WEST*
ACT,31,32,CO,10000,17/NORTH*
ACT,33,34,CO,10000,18/SOUTH *
STA,28,1,1*
STA,30,1,1*
STA,32,1,1*
STA,34,1,1*
STA,39,1,1*
SOU,40,0,1*
VAS,40,6,UF,8*
ACT,40,40,CO,10000*
REG,41,1,1*
ACT,41,42,NO,11*
QUE,70/OVERLOAD*
ACT,70,71,CO,10000,14/OVERLOAD*
STA,71,1,1*
REG,80,1,1*
VAS,80,6,UF,9*
REG,60,1,1*
ACT,60,58,NO,12*
REG,42,1,1,A*
ACT,42,43,(9)A1.EQ.1*
ACT,42,44,(9)A1.EQ.2*
ACT,42,45,(9)A1.EQ.3*
QUE,43/TP1 OUT*
QUE,44/TP2 OUT*
QUE,45/TP3 OUT*
QUE,49/TP1 IN,(10)52*
QUE,50/TP2 IN,(10)54*
QUE,51/TP3 IN,(10)56*
QUE,58/TR ARR,(10)52,54,56*
QUE,53/ONE OUT*
QUE,55/TWO OUT*
QUE,57/THREE OU*
ACT,43,46,CO,10000,11*
ACT,44,47,CO,10000,12*
ACT,45,48,CO,10000,13*
STA,46,1,1*
STA,47,1,1*
STA,48,1,1*
MAT,52,1,49/53,58*
```

FIGURE 11 (continued)

```
MAT,54,1,50/55,58*
MAT,56,1,51/57,58*
ACT,53,61,NO,13,20/ONE OUT*
ACT,55,62,NO,13,21/TWO OUT*
ACT,57,63,NO,13,22/THREE OU*
STA,61,1,1*
VAS,61,6,UF,10*
STA,62,1,1*
VAS,62,6,UF,10*
STA,63,1,1*
VAS,63,6,UF,10*
PAR,1,0.0,-.5,.5,.167*        ASSUME NO. N&W SUPP. TRAIN ARRIVAL TIME
PAR,2,4.,2.,10.*              ASSUME EX. N&W SUPP. CONT. STREET TIME
PAR,3,.1485,.1275,.1725,.004* TESTED NO. N&W DATA. CAR SWITCH TIME
PAR,4,.22,0.0,4.*             TESTED PO. N&W DATA. TIME TO START OFFLOAD
PAR,5,,4.,7.*                 TESTED UN. N&W DATA. TIME TO START OFFLOAD
PAR,6,,0.0,9.*                TESTED UN. N&W DATA. TIME CONT. DELEVERD
PAR,7,.1025,.05,.3334*        TESTED EX. N&W DATA. CONT. OFFLOAD TIME
PAR,8,,8.,17.*                ASSUME UN. N&W SUPP. INIT. CONT. PICKUP
PAR,9,.105,.067,.2667*        TESTED EX. N&W DATA. CONT. ONLOAD TIME
PAR,10,3.,.5,5.5,.8333*       ASSUME NO. N&W SUPP. TIME TO REV. CAR
PAR,11,.2,.05,.4,.06*         ASSUME NO. N&W SUPP. LOT SPOT TIME
PAR,12,.5,.1,1.,.15*          ASSUME NO. N&W SUPP. TRACTOR PICKUP TIME
PAR,13,.2,.1,1.,.15*          ASSUME NO. N&W SUPP. TRACTOR DEPART TIME
PAR,14,,1.,4.*                TESTED UN. N&W DATA. NOTIFICATION TIME
COL,1/OVER PKR,2/DAYS OP,3/IN YARD,4/CARSPDAY,5/CONOFFPD,6/A INYARD*
COL,7/REG LOT,8/TOT PARK,9/INBD CON,10/OTBD CON,11/TOT SWT,12/INSWPCAR*
COL,13/OTSWPCAR,14/TIMEPCAR,15/OFTMPCON,16/ONTMOCON,17/CONOFPRN*
COL,18/CONONPRN,19/CONONPD,20/SWTMPWD,21/TMPINCON,22/TMPOTCON*
COL,23/TCARSPRN,24/TCONPROT,25/TCONPRNIN*
FIN*
```

FIGURE 11 (continued)

ten user functions. These FORTRAN written subroutines range from a simple, one statement procedure used in user function 2, to the very complex logic of user function 1. The description of the logic employed for the model that follows is limited to a general discussion which explains the function of each subroutine, and their interrelationships. The detailed logic for each subroutine is presented in the appendices. The FORTRAN program listing for Function UF(IFN) is in APPENDIX D.

Function UF(IFN)

A user function is specified each time a transaction realizes source nodes 1 and 40; regular nodes 2, 3, 4, 5, 6, 20, 22, 24, 35, and 80; and statistics nodes 61, 62, and 63 in the Q-GERT model. Every node thus realized sets IFN equal to the user function number designated at the node. Simulation time, TNOW, is stopped each time Function UF(IFN) is entered.

Branching to specific statements within Function UF(IFN) is determined by the value of IFN at a computed go to statement in the beginning of the subroutine. For example, if IFN=1, the subroutine is entered at a point that contains the logic for user function 1 of the FORTRAN programming. The system is initialized when TNOW, the current simulation time for a simulation run, is equal to zero. The model's clock time, IHOUR, is set at 00:00 hours of day one, when TNOW equals zero, representing the beginning of the simulation at midnight of the first day. Clock time, IHOUR, is then incremented by one each time simulation time, TNOW, advances to the next unit of time. IHOUR is reset to zero at the end of each twenty-four simulation time units. The current calendar day, IDAY, is incremented by one each time IHOUR equals twenty-four.

The time sequencing for the model is accomplished whenever user function 1 is specified during the simulation. This enables the status of the system to be checked and "daily" events to be scheduled at the start of each hour, or day, if required by the system. TNOW is suspended until all required tasks are accomplished. The manner in which the daily events are scheduled is discussed below.

User Function 1

User function 1 provides the logic required to schedule transactions through the system. This logic includes the input of train arrivals and departures, enforcement of car switching priorities, interpretation of working rules, the timing of all transaction movement through the terminal, parking lot monitoring, and the collection of observations used for the output statistics for a simulation. The logic in this function also provides the means to initialize the status of the system at the start of a simulation run (TNOW=0.0). This is accomplished by assigning attributes to transactions representing appropriate numbers and types of flatcars, containers, and tractors in the system.

User function 1 is specified at source node 1 in the Q-GERT network model. As previously noted, this node is realized at the start of a simulation run, and at the end of each constant one time unit for the duration of a run, providing the timing required for the system.

The simulation clock, IHOUR (initialized at zero), is incremented by one each time the FORTRAN program for the network is entered at user function 1. The value of IHOUR is then tested. If IHOUR is less 24, the simulation is in a current day, and the status of the system at time IHOUR is checked. If IHOUR is equal to 24, a current day has just ended, the status of the system is updated, the results of the day's operations are compiled and recorded, and the collector and indicator variables re-initiated for the coming day's operations. In the discussion that follows, the end of a day's logic is first presented, then the logic for daily operations follows.

When the status of the system indicates that a day has just ended, the current status of the overflow parking lot (Q-node 70) is checked. Transactions representing containers were placed in the overflow parking lot during the current day, if at any time during the day the regular parking lot is full. Should there be any transactions in the overflow lot, the server for activity 14 behind Q-node 70 will be busy. If the server is busy, overparking occurred during the day, and the status of the parking lot is updated before collecting overparking observations. If the server is idle, no overparking occurred during the day and this observation is recorded.

Given that overparking did occur during the day, the number of containers overparked is recorded by setting the variable IOVP(IDAY) equal to the number of transactions residing in Q-node 70, plus the transactions in service on activity 14. The remaining capacity of the regular parking lot (MXPARK) is then checked to determine how many spaces are available in the regular parking lot. If MXPARK is equal to zero, the regular lot is full, and the day's overparking, IOVP(IDAY), remains unchanged. If MXPARK is greater than zero, there is space available in the lot and overparked containers are moved from the overflow lot to the regular lot until the regular lot is full, or all overparked containers have been moved. This is accomplished through a call to subroutine STAGO(14,80,0.0,0,ATT) for each container that is to be moved. Subroutine STAGO removes the current transaction that is in service on activity 14 and places it into the regular parking lot via user function 9. The current capacity of the parking lot, MXPARK, is then decreased by one, and the value of the transaction's attribute 2 tested to determine the type of departure. The transaction is then placed into the proper section of the regular parking lot through a call to subroutine PTIN(NODE,0.0,TNOW,ATT).

Once the parking lot section of the model has been updated to reflect the current status of the system at the end of a day, IHOUR is re-initialized to zero for the coming day. The day's total parking, IPARKT(IDAY), is recorded and IDAY is then incremented by one. The daily working rules can then be tested to determine if the coming day is a non-working day or a normal working day.

The daily working rules are an important feature of the program logic. Railroads, generally, operate their trains seven days a week during the entire year. The loading and unloading terminal operations do not, as a rule, follow this pattern. These activities, including container pick-up and delivery, are usually on a five-day-a-week work schedule. This section of the program allows a user to specify the weekend work rules for a simulation that are in force at a particular terminal facility. The indicator variables ISAT and ISUN are used for this purpose. They are initialized at the start of a simulation in subroutine UI. The indicator variable IWORK is set to one, or zero, for each day of the simulation in accordance with the day of the week and the values of ISAT and ISUN. If ISAT or ISUN equal one, work is permitted. If either of them equal zero, no work is permitted. Once the work rule has been established for the coming day, the day's train arrival and departure schedules are read into the simulation.

Provision has been made for ninety days of operations with ten train arrivals and departures per day. Individual train arrival times, ATIME(IDAY,M), and departure times, DTIME(IDAY,M) are originally initialized in subroutine UI at 25.0 hours. Scheduled train arrival and departure times can then be re-initialized in subroutine UI to specific times for individual train arrivals or departures on a given day. These input data are then stored until called for by the simulation on an appropriate IDAY.

The arrival schedule is accessed by setting the variable TRANA equal to ATIME(IDAY,M) for each possible arrival during the current IDAY. If TRANA is less than 25.0, a train is scheduled to arrive during the day at the time indicated by the value of TRANA. When TRANA is equal to 25.0, no additional trains are scheduled to arrive during the day. For each scheduled train arrival, an actual train arrival time is generated (ARR). This time is randomly sampled from a probability distribution relating to the experience of a terminal under study. Therefore, a train's simulated arrival time is set equal to the scheduled time, plus or minus, a random deviation. The procedure provides for a more realistic train arrival pattern during a simulation.

An arriving train's loading configuration is read into the simulation at the train's arrival time during a run. The type, number, and container loading of the cars on a specific train was stored through subroutine UI, when the system was initialized, to be read into the simulation at the train's arrival time. For example, assume that a train is scheduled to arrive on the fifth day of a simulation, and that it is the second train to arrive for the day. If the train is made up of twelve type 1 cars such that there are five cars loaded with two containers, two cars with one container each, and no empty cars on the train, the train's configuration storage variable, ITP1(IDAY,M), indicating the day of arrival, and the train's container loading would be coded as follows:

```
ITP1(5,2)=12050200
```

Reading from left to right, the variable ITP1(5,2) first two digits (12) indicate that there are twelve cars on the train. The third and fourth digits (05) indicate that five of the cars are loaded with two containers each. Digits five and six (02) specify that there are two cars with one container each, and digits seven and eight (00) specify that there are no empty cars on the train. The variables ITP2(5,2) and ITP3(5,2) would be set equal to zero, indicating that there are no type 2 or type 3 cars on the arriving train. Each variable is then decoded at the train's arrival time on day five during the simulation, such that each car, of each type, can be scheduled to enter the inbound yard at the proper arrival time as a separate transaction. The attributes assigned to each of the transactions representing a flatcar are as follows:

Attribute	Possible Values	Definition
1	1, 2, or 3	Car type
2	0, 1, or 2	Number of containers per car
3	0 or 1	Facing direction of car

The cars (transactions) can then be individually placed into branching node 10, representing the entry point of the inbound yard of the Q-GERT model, through a call to subroutine PTIN(10,ARR,TNOW,ATT) at simulation time TNOW + ARR.

Some railroads notify a container's consignee when their container has arrived in the yard to provide them with an expected offload time for planning purposes. Others wait until the container is on the ground before notifying the consignee. An indicator variable, NGND, is incorporated into the model to specify which notification rule is in effect for a simulation. If NGND is initialized at one, notification will take place after a container has been offloaded. If NGND is equal to zero, notification will be accomplished when a container enters the yard on a flatcar. In either case, pick-up times must be scheduled and matching tractor transactions for the containers entered into the simulation at the scheduled times. If NGND is equal to zero for a simulation, this scheduling is accomplished through a routine at train arrival time in user function 1. Should NGND be equal to one, the process is delayed until containers are offloaded from their flatcars and placed in the parking lot. The notification process is described in user function 4.

Scheduled train departures are read into the simulation in a manner similar to the arrival schedule. All possible departure times, DTIME(IDAY,I), are first initialized at 25.0. Specific departure times are then entered. The coding required for the configuration of a departing train consist of the outbound direction of the train, and the maximum number of flatcars permitted on the train. For example, given that a train is departing West-bound with a capacity of twenty-five cars, the coding would be as follows:

```
IDEPT(IDAY,I)=4025
```

Reading from left to right, the first digit of the code (4) indicates the direction of the train, and the last three digits (025) indicate the maximum intermodal flatcar capacity permitted on the train.

The train departure schedule process is accomplished at the start of each new day, and is again checked just prior to each train departure during the day. Cars waiting at the outbound yard for a train departure at the beginning of the day are given priority on the day's scheduled outbound trains. After all possible waiting cars have been scheduled to depart during the day, cars arriving at the outbound yard during the day, but prior to departure time, will also be placed on a departing train, given that there is room remaining on the train at departure time.

The procedures described above are accomplished at the end of each day during a simulation run. They provide for the sequencing of events for the coming day's operations. That is, the status of the parking lot is updated, working rules for the new day are established, and train arrival and departure schedules are entered into the simulation. The discussion that follows describes the actions that are taken at the beginning of each hour during the simulation.

The indicator variable, INIT, is set equal to zero in subroutine UI. This permits the system's starting conditions to be entered into the system before the current status of the system is first checked. Once the system has been initialized, the work day indicator variable, IWORK, is tested. If IWORK is equal to one, a normal work day is indicated, and the status of the siding is checked. If IWORK equals zero, no work is permitted and control is returned to the main Q-GERT program. Each simulation run begins with the first day (IDAY=1) assumed to be Monday, a working day, and the time of day set at 00:00 hours (IHOUR=0). Therefore, IWORK=1, and the first action, after system initialization, is to test the siding operation.

Whenever the siding is empty, the indicator variables ISWIN and ISWOUT will both be equal to zero. If either is equal one, there are cars on the siding and inbound switching to the siding from the yard is not permitted. Assuming that both variables are equal to zero, a switch to the ramp can be scheduled, given the proper switching rules. The switching rules for this model are straight forward, but appropriate for many intermodal terminals (they can be modified for more complex systems). Only one inbound-outbound switching is allowed per day for the siding. Further, any switching must occur between the hours of 6:00 PM of the current day and 8:00 AM of the following day. Since any car switching onto the siding must be accomplished prior to 8:00 AM for any day, any cars on the siding at 8:00 AM will remain there until 6:00 PM the same day. Therefore, a test of IHOUR in user function 1 before scheduling a switch, coupled with the status of the indicator variables ISWIN and ISWOUT, is sufficient to insure that switching be limited to one inbound and one outbound switching per day.

The scheduling of a switch to the siding is accomplished by first checking the status of the inbound yard to see if loaded cars are available. If there are loaded cars, they are selected for switching in accordance with the priority switching rules for the simulation. If there are no loaded cars in the yard, empty cars are located if they exist. The outbound (rail departure) parking lot is also checked for matching containers if empty cars are to be switched. If there are none, no switching can be scheduled, and control is returned to the main program. Assuming there are loaded cars in the inbound yard, they are selected for switching based on the values of the variables IPRI1 and IPRI2, which are initialized in subroutine UI, and establish the following priorities:

IPRI1,IPRI2=(0,1,2,3,4)

Priority	Definition
0	The type of car with the longest number in the yard
1	Type one cars
2	Type two cars
3	Type three cars
4	The type of car waiting the longest in the yard

IPRI1 will take precedence over IPRI2. If neither IPRI1, nor IPRI2 can be satisfied, then priority is given to the type of car waiting the longest in the yard.

The first car selected to be switched from the inbound yard initiates the procedure to move all other cars that can be switched to the siding. A variable, IQ, is set equal to the activity number behind the Q-node containing the cars to be switched. The time required to switch (TIMESW) the first car is randomly sampled from a switching time probability distribution. The indicator variable NSWIN (set at zero at the start of each day) is incremented by one to record the number of the switch, a collector variable, TSWIN(IDAY,NSWIN), is set equal to TIMESW, and the indicator variable ISWIN is set to one. The total number of cars currently in the inbound yard is calculated and the value recorded. A call to subroutine STAGO(IQ,19,TIMESW,0,ATT) then removes the transaction currently in service on activity IQ, and places it in branching node 19 at time TNOW + TIMESW with its associated attributes. Branching node 19 routes the transaction to regular node 36 if the car is reversed. This will specify user function 2 to be called. If the car is not reversed, the transaction is sent to regular node 20 where user function 3 is specified. Specifying either user function 2, or user function 3, results in moving the next car to be switched from the yard to the siding. The logic for moving these additional cars will be discussed in the sections for these user functions.

At any time the indicator variables ISWIN, or ISWOUT, are equal to one, the siding will be occupied. In this event, the program logic determines if the cars at the siding can be offloaded, onloaded, or switched to the outbound yard. The first step in this part of the program is to test the time of day. The daily working rules for loading and offloading containers stipulate that normal working hours are between 8:00 AM and 6:00 PM. If IHOUR is outside this range, the status of the siding is tested to determine if cars can be switched from the siding to the outbound yard. Should IHOUR be within the range of normal working hours, the check is to determine the container activity to schedule at the siding.

Siding activities are sequential. Cars enter the siding, are offloaded, onloaded, then switched to the outbound yard. Consequently, if the siding is occupied, the first check is for an offloading activity. The indicator variable LOADOF is set equal to one when offloading begins. LOADOF is set to zero at the completion of the activity. Therefore, if LOADOF equals one, an offload is in progress and control is returned to the main program. If LOADOF is equal to zero, it is possible to schedule an offload activity. The current status of the siding is checked to determine if there are cars waiting to be offloaded. The nature of the server on activity 8 behind Q-node 21 indicates this status. A busy server on this activity indicates there are cars waiting to be offloaded. The first container to be offloaded is scheduled to be removed from a car by a call to subroutine STAGO(8,22,TIMEOF,0,ATT). This action removes the transaction in service on activity 8 and places it in regular node 22 at time TNOW + TIMEOF. User function 4 is specified as the transaction passes through node 22. All other cars to be offloaded are processed by the logic programmed in user function 4.

An idle server on activity 8 indicates that the cars at the siding have been previously offloaded. In this event, the status of the siding is checked to determine if there are empty cars that can be onloaded. The indicator variable LOADON is then tested. If LOADON is equal to one, an onload is in progress and control is returned to the main program. Given that there is no offload in progress, the status of the server on activity 9 is checked. Should the server be idle, the cars on the siding have been previously loaded and they are waiting for the proper time of day to be switched to the outbound yard. Therefore, control is returned to the main program. When the server on activity 9 is busy, an onloading process can begin. LOADON is set equal to one, and the minimum time remaining to begin an onload operation before the end of normal working hours, TIMEON, is determined. The onloading activity is initiated by a call to subroutine STAGO(9,24,TIMEON,0,ATT). The transaction in service on activity 9 is removed and placed into node 24 at TNOW + TIMEON. User function 5 is specified at node 24 and the container onloading process is completed through the logic programmed in this user function.

Once all the flatcars on the siding have been offloaded and reloaded, a switch to the outbound yard is possible. The status of the server on activity 10, behind Q-node 25, indicates this condition. A switch to the outbound yard can be scheduled if the server is busy. To initiate a switch, the indicator variable LOADOU is set equal to one; time required to switch, TIMEOU, is determined and a call to subroutine STAGO(10,26,TIMEOU,0,ATT) is accomplished. When node 26 is realized at time TNOW + TIMEOU, the transaction branches in accordance with the value of attribute 4. Attribute 4 designates the outbound direction of the car being switched. After branching occurs, user function 6 is specified as the transaction is routed to its proper outbound Q-node. User function 6 is then employed to transfer the remaining cars on the siding to the outbound yard. The logic for this user function is detailed in the user function 6 section of this chapter.

User Function 2

Transactions representing flatcars that are switched from the inbound yard to the siding are tested at conditional branching node 19 to determine if a car is facing in the proper direction to be moved to the siding. The branching is based on the value of attribute 3 of the transaction. If the value of the attribute is one, the car is placed on the siding through node 20. Should the value of the attribute be zero, a reversed car is indicated, and a delay time is encountered to turn the car prior to placing it on the siding.

Flatcars that are routed to be turned in the manner described above pass through node 36 where user function 2 is specified. The purpose of this user function is to indicate that the transaction was not placed on the siding, but requires additional routing. However, model logic also requires that the next car in the yard be moved from the yard to the siding as long as there is space available on the siding. User function 2, therefore, simply transfers program control to a section in user function 3 beyond the point where the number of cars switched to the siding ISWITC, is not incremented. The next car to be switched from the yard is then identified and processed through the logic provided in user function 3. This user function is described below.

User Function 3

User function 3 is specified at node 20 each time a transaction is routed from the inbound yard of the model to the siding. The purpose of this user function is to initiate the switching of the next car in the yard to the siding. A variable, ISWITC, is used to indicate the current number of cars switched to the siding during an in-switching operation. When the value of ISWITC is equal to the maximum number of cars permitted on the siding, the switching operation is stopped.

Cars can be moved onto the siding from two sources. The first source that is checked is the server following Q-node 37, on activity 7, in the system network. If this server is busy, then a previously reversed car has been turned and is ready to be placed on the siding. In this event, a call is made to subroutine STAGO(7,20,0.0,0,ATT) and simulation control is returned to the main program. This action causes the transaction currently in service on activity 7 to be placed directly on the siding via node 20 with a zero time delay through the main Q-GERT program. The realization of node 20 using this method allows user function 3 to be immediately respecified. ISWITC is incremented by one, and a test is again made for reversed cars. When there are none, the regular inbound yard is checked for cars of the proper priority that were first identified as available for switching in user function 1.

Cars are moved from the inbound yard via user function 3 through a call to subroutine STAGO(IQ,19,0.0,0,ATT) as long as the server on activity IQ is busy and the variable ISWITC is less than the maximum capacity of the siding. If the server, IQ, should become idle, indicating no remaining transactions in the preceding Q-node, and there is still room on the siding for additional cars, program control is returned to user function 1 to determine if there are lower priority cars available that can be switched. The process continues until the switching activity is complete. The indicator variables are then set to indicate that cars are on the siding and that they are ready to be offloaded. Program control is then returned to the main program.

User Function 4

User function 4 is called when containers are offloaded from a flatcar. This is accomplished by removing a transaction in service on activity 8, behind Q-node 21, and routing it through node 22 by a call to subroutine STAGO(8,22,0.0,0,ATT) in user function 1. A call to subroutine GETAT(ATT) in user function 4 returns the values of the attributes associated with the transaction. Attribute 1 identifies the type of containers. The value of attribute 2 represents the number of containers on the car. This information is used to generate separate transactions representing containers to be placed in the over-the-road departure section of the parking lot. Additionally, a container pickup (tractor) transaction is also generated for each container placed in this section of the parking lot if the indicator variable NGND is equal to one for the simulation. Should the regular lot become full at any time during the process, containers that are removed from the cars are placed in the overflow parking lot.

The cars at the siding are offloaded individually, removing one container at a time. An offload time is generated for each container, and the total offload time is accumulated by a collector variable for each simulation run. As a car is offloaded, it is moved into Q-node 23, and the status of the server on activity 8 is determined. If the server is busy, there is at least one other car to be offloaded. The process is reinitiated by a call to subroutine STAGO(8,22,0.0,0,ATT) and continues until all cars have been offloaded. When offloading is completed, indicator variables are set to indicate that there are cars available to onload and control is returned to the main program.

User Function 5

User function 5 is initiated each time containers are processed for loading. The sequence is begun when a transaction in service on activity 9 is removed from service and placed into node 24 by a call to subroutine STAGO(9,24,TIMEON,0,ATT) in user function 1. The attributes of the transaction are accessed by a call to subroutine GETAT(ATT). The value of attribute one identifies the type of container that must be accessed in the rail departure section of the parking lot to load onto a particular type of flatcar. If containers of that type are available in the lot, they are loaded on the car. Otherwise, the cars remain empty.

The logic contained in user function 5 tests the values of attributes 4 and 5 associated with the transactions representing containers in the parking lot. The value of attribute 4 represents the outbound direction of a container from the yard. The value of attribute 5 stipulates a container's special handling characteristics. Ordinarily, two containers are loaded on flatcars for outbound shipment, and these must be matched by direction of departure. Occasionally, however, only one container is loaded on a car. The test for the container loading conditions is such that transactions cannot be moved directly from node 24 in the model to Q-node 25. Therefore, special transaction handling procedures have been developed for user function 5. These procedures, in essence, destroy each transaction entering node 24 after the transaction's attributes have been retrieved. Then, a new transaction, with the mark time of the original transaction, is created and inserted into Q-node 25 after the proper matching has occurred. The value of attribute 4 for this new transaction is set to the value representing the destination of the loaded containers.

The logic in user function 5 is first used to match a section of the parking lot that represents the type of containers that are required to be loaded on the car type being processed. The corresponding transactions in the parking lot are then removed individually. The value of attribute 5 is tested for the first transaction removed. If the value of the attribute is one, a single transaction with the container's attributes is placed into Q-node 25 after an onloading time delay, and the current capacity of the parking lot is increased by one. On the other hand, should the value of attribute 5 for the initial transaction that was removed be equal to two, the parking queue must be searched for another transaction with a matching destination. If no match exists, the container is loaded individually and the parking capacity is increased by one. If a match does occur, the time to load two containers is generated, one container transaction is destroyed, and the other inserted into Q-node 25, now representing a flatcar, with the proper attributes and mark time. The parking capacity is increased by two, reflecting the additional parking made available by the loading of two containers.

After each flatcar has been processed for loading by this user function, the status of the siding is again tested to determine if other cars are available for loading. If the server on activity 9 is busy there is at least one additional car to be processed. Subroutine STAGO (9,24,0.0,0.ATT) is recalled, and the onloading continued until there are no cars remaining to be loaded. When all cars have been processed, indicator variables are set to indicate there are cars at the siding waiting for outbound switching. Program control is returned to the main Q-GERT program, and simulation time is allowed to advance.

User Function 6

When the status of the system indicates that a switching to the outbound yard can be initiated, user function 6 is employed. This user function is specified whenever a transaction realized node 3, 4, 5, or 6. The operation is started by a call to subroutine STAGO(10,26,TIMEOU, 0,ATT) in user function 1, which removes the transaction in service on activity 10 and inserts it into conditional branching node 26.

Transactions realizing node 26 are routed to following nodes based on the value of attribute 4. This attribute records the outbound direction of a flatcar, or indicates that the car is empty. The possible values of attribute 4 are as follows:

Value	Definition	Outbound Q-node
1	East bound	27
2	West bound	29

Value	Definition	Outbound Q-node
3	North bound	31
4	South bound	33
5	An empty car	Regular node 35

User function 6 is specified for all attribute values except the value of five. If the value of the attribute is five, user function 7 is specified for the special handling of empty flatcars. When either of these user functions is specified, however, cars are sequentially removed from activity 10 as long as there are cars on the siding, and routed to their respective outbound Q-nodes to wait for a train departure from the system.

User Function 7

User function 7 performs a unique service for empty flatcars that are switched to the outbound yard, in addition to switching a following car on the siding to the yard. Empty cars can be retained in the system instead of being shipped to another destination. The modeler selects the maximum desirable number of empties that are to be permitted in the system. User function 7 is then used to determine the number of empty cars that are located in the inbound yard each time the function is specified. If the number is below that specified, the car is returned to the proper Q-node in the inbound yard for future use in the system. If the number of empty cars in the system is above that specified, the car is shipped out of the system.

User Function 8

User function 8 is provided to assign values to attributes 4 and 5 for random over-the-road container arrivals of the system. Once these values have been determined, they are associated with the arriving transaction for the rail departure section of the parking lot. The transaction is then placed into conditional branching node 42 where it is routed to the proper parking Q-node to await a flatcar for loading outbound.

User Function 9

This user function performs two extremely important activities for the model. First, when user function 9 is specified, control of the program is in the main Q-GERT program. This is accomplished by calling subroutine PTIN(80,TIME,MARKTIME,ATT) at various locations within Function UF(IFN), followed by a RETURN statement. The call places the current simulation transaction into regular node 80 and the return statement enables the main Q-GERT program to update all files to the current simulation time. The only purpose of node 80 in the network is to specify user function 9. The logic in this user function is such that every time the function is entered at user function 9 the attributes of the current transaction are retrieved, and program control is then immediately passed to the point within Function UF(IFN) immediately following the point where the PTIN call was originated. Therefore, current transaction attributes can be retrieved when required, or a zero time delay initiated, without violating ANSI FORTRAN specifications that forbid recursive calling within functions.

The second activity provided by user function 9 is to place arriving over-the-road containers in the parking lot that enter the system with scheduled tractor container pick-ups. This method of placing outbound (rail departure) containers in the system is employed when the system specifies that container consignees are notified to pick up a container arriving by rail when the flatcar enters the inbound yard. The main purpose of this logic is to decrement the current capacity of the parking lot by one for each container arriving over-the-road.

User Function 10

The purpose of user function 10 is to increment the current status of the parking lot by one for each container departing the parking lot over-the-road. This user function is also used to schedule the return to the parking lot any container that departs the system over-the-road that is to be returned after it has been offloaded by the consignee. This is accomplished by a call to subroutine PTIN(80,DELAY, TIME,TNOW,ATT) as the container departs the system via statistics nodes 61, 62, or 63.

Supplemental Computer Programs

There are three computer subroutine FORTRAN programs required for the intermodal terminal simulation, in addition to the Q-GERT data needed to describe the model. These are Subroutine UI (user input), Function UF(IFN) (user function), and Subroutine UO (user output). Together, the entire model requires approximately 16 seconds on an IBM 360/165, Model 2 computer to compile and link the model's subroutines with the Q-GERT Analysis Program. The execution step uses approximately

13.2 seconds of central processing unit (CPU) execution time for each simulation run of the intermodal terminal system. The user function subroutine was described in the previous section. The requirements for subroutines UI and UO are described below.

Requirement for Subroutine UI

The primary purpose of subroutine UI is to read into the simulation the values of the variables needed to define the system to be studied, to establish switching priorities and weekend working rules, and to provide the vehicle to read in and store train arrival and departure schedules until called for during the simulation. Although this subroutine is used to establish the initial values for variables to describe the beginning status of the system, it is not used, in the conventional manner, to initialize the status of the system. This feature in the system initialization is accomplished through a special routine incorporated into function UF(IFN). An example of the FORTRAN program listing for subroutine UI is included in APPENDIX C.

The indicator variables INIT, and INPRO, are set at zero in subroutine UI to enable the programming logic in function UF(IFN) to place the desired number of transactions, with the proper attributes, in selected Q-nodes at the beginning of each simulation run. The quantities and types of transactions to be generated for the initial terminal status are determined by the values set for the input variables associated with the parking lot, and the inbound and outbound yards. Additionally, the variable IDAY is set equal to one, and IHOUR is initialized at zero. These four variables, INIT, INPRO, IDAY, and

IHOUR set the conditions necessary for the simulation to operate. The remaining variables initialized in this subroutine establish the conditions of the system that are sufficient for the model to represent the specific terminal under study.

The values of the variables MXRAMP and MCAP essentially define the size of the system. The model assumes that there will always be adequate room for flatcars in the inbound and outbound yards. Therefore, the initial value of MXRAMP sets the maximum number of flatcars that are permitted at the siding at any point in time; a physical limitation at any terminal. Likewise, the initialized value of MCAP establishes the maximum number of containers that can be spotted in the regular parking lot. A system variable, MXPARK, is set equal to MCAP on initialization, and records the current capacity of the parking lot at any point in time during a simulation run.

Inbound flatcar switching priorities and weekend work rules for a simulation are also established on system initialization. The variables IPRI1 and IPRI2, as previously discussed, designate which cars will be moved first from the inbound yard. The working rule indicator variables ISAT and ISUN, provide the option to shut down the siding and parking lot operation on weekends. If either of these variables are initialized at zero, there will be no work on the days indicated during the simulation. This is accomplished by setting the work day indicator variable, IWORK, equal to zero at appropriate times during a run. IWORK is always initialized at one.

A container consignee notification indicator variable, NGND, is also provided for in the model. If NGND is initialized at one, notification will take place after a container is offloaded from a car in the parking lot. If the variable is set at zero, notification will take place when the flatcar loaded with the container enters the inbound yard.

There are several indicator variables that are always initialized at zero for a simulation. They are used to aid in the collection of statistics during a run. The more important ones are NSWIN, NOFF, NON, and NOUT. These variables, used in conjunction with the current value of IDAY, establish the number of switches to the inbound yard, container offload activities, container onload activities, and outbound switches for each day during a simulation. Other variables are used to accumulate the numbers of flatcar transactions, by type, for a given run.

The variables ATIME(I,J) and DTIME(I,J) are used to initialize the scheduled arrival and departure times of individual trains for an entire simulation run for train J, on day I. Therefore, each train scheduled to arrive into, or depart from, the system is read into the simulation upon initialization and stored until called for by the program. Further, for an arriving train, the train's composition is also provided for by coding the variables ITP1(I,J), ITP2(I,J), and ITP3 (I,J), as described in the preceding section. The code records the number and container loading of type one, two, or three flatcars. For each departing train, the variable IDEPT(I,J) must be provided and coded such that the departing direction and maximum capacity of each train can be read into the simulation at the proper time. This procedure was also discussed in the previous section.

The FORTRAN coding required to initialize a simulation requires that input values be determined for forty-eight variables to define the system. In addition, the arrival and departure schedule must be provided for. The number of variables requiring user supplied values will vary depending upon the number of days to be simulated, and the number of train arrivals and departures per day. For example, the simulation of one system for a period of sixty-three days required one hundred fifty FORTRAN statements to define the arrival schedule, and thirty-two additional statements to provide for train departures.

Requirements for Function UF(IFN)

There are approximately seven hundred eighty (780) FORTRAN programming statements required for function UF(IFN). This part of the model, as described in the section on user function logic, provides the timing required for the system and enables the collection of observations about the system each time the status of the system changes. The observations gathered during a run of a simulation in this subroutine are then accumulated in subroutine UO. APPENDIX D contains the complete FORTRAN program listing for function UF(IFN). A discussion of the collector variables used in this subroutine is given below.

The container parking facility is of prime interest when evaluating an intermodal terminal system. There are three programmer defined collector variables provided to accumulate daily observations about the parking lot. These are IOUP(IDAY), MOVE(IDAY), and IPARKT (IDAY). IOUP(IDAY) is used to record any container overparking that occurs on any day during a run. MOVE(IDAY) records the number of containers moved from the overflow lot to the regular lot. The total number of containers in the regular parking lot each day is recorded as IPARKT(IDAY).

The variables TSWIN(IDAY,NSWIN) and ISWINT(IDAY,NSWIN) are used to collect data on the inbound switching operation of the terminal facility. TSWIN(IDAY,NSWIN) accumulates the total switching time from the inbound yard to the siding for each switch that has taken place during the day. The total number of cars switched to the siding is accumulated by ISWINT(IDAY,NSWIN). Outbound switching times from the siding to the outbound yard are gathered by the variable TSOUT(IDAY, NOUT) and NSWOUT(IDAY,NOUT) is used to record the number of cars switched to the outbound yard. Container onloading and offloading service times at the siding are accumulated for each operation during the day by the variables TON(IDAY,NON) and TOFF(IDAY,NOFF) respectively. The number of containers loaded, or unloaded, are recorded by LCON(IDAY,NON) and LCOFF(IDAY,NOFF).

The observations gathered through the use of each of the variables described in this section represent the daily operations within the system. The information is compiled in subroutine UO at the end of each run to be used to calculate the output statistics for the simulation. Subroutine UO is presented in the next section.

Subroutine UO

The standard Q-GERT output statistics provided at the end of a simulation by the Q-GERT Analysis Program for this model cannot be easily interpreted by an individual not completely familiar with the model and the programmer written FORTRAN logic. Furthermore, because of the unique way servers associated with activities following Q-nodes are employed in the model, the standard output statistics concerning server utilization can be misleading. This is also true of the output statistics compiled for the sink and statistics nodes at the end of a simulation. This is not to imply that the statistics generated by the program are useless. Quite the opposite is true. The intent is to warn potential users of the model to be extremely cautious when attempting to interpret the results of a simulation based on the standard output statistics generated by the analysis package. Fortunately, this problem can be minimized through the use of the programmer written output subroutine option provided with the Q-GERT Analysis Program.

All Q-GERT variables, subroutines and functions, in addition to those provided by the programmer, can be accessed in subroutine UO. This subroutine is automatically called by the main Q-GERT program just prior to the compilation of the standard output statistics at the end of each run. Therefore, all data collected by the Q-GERT program during a run is available for calculations in subroutine UO at the end of a run. For example, the average waiting time of all transactions passing through a Q-node during a run can be accessed through the Q-GERT function AVEWT(NODE), and the average number of transactions in a Q-node for the run can be accessed through function

TINIZ(NODE)/TNOW. This Q-GERT feature, coupled with the programmer collected data in function UF(IFN), enables the compilation and formatting of the simulation output in a more readable form. For the Intermodal Terminal Model, the output subroutine requires three hundred forty-eight (348) programmer written FORTRAN statements. A listing of subroutine UO is included as APPENDIX E. An example of the programmer formatted output for a simulation is discussed in the chapter on implementation.

Validation and Testing

The validation and testing of the intermodal terminal simulation model has evolved through several successive steps. In the earlier phases of this process, the results of a simulation run were compared to precomputed results that were based on hypothetical input data. When the model was more fully developed, comparisons were made between simulation results and the historical records of an existing intermodal terminal facility. These procedures do not, of course, insure model validity, but they do provide a reference from which to establish confidence in the model. It would be more desirable to compare the results of an intermodal terminal simulation to analytical results from the same system. However, Van Horn has observed that such statistical validation is not always possible [86]. This situation is applicable to the validation of this model where analytical techniques for the intermodal terminal system have not been developed. Therefore, the validation of the model relies first on the approach suggested by Naylor and Finger [69], using the working procedures suggested by Law [29 and 30]. Then, confidence in the validity of the model is established by inference. That is, if all the critical parts of the model are assumed to be valid, then by inference, the entire model should be valid. The critical components of the model are the arrival and departure section, the siding operation, and the switching function. This last phase of validation is accomplished by comparing simulation results with historical observations for an intermodal system.

The original construction of the model was based on interviews with experts who provided information on intermodal operations, and experience gained through the observation of an actual terminal system. The components of the system were then incorporated into the model such that the system developed represented the primary elements of an intermodal terminal. The model was then tested using constant transaction inputs for the system, coupled with constant activity service times, to confirm that transactions flowed through each component of system as intended. After each section had been individually tested, verification of the overall performance of the model was accomplished by comparing the results of a simulation run with precalculated outputs for the simulation, given the constant input data and service times. The results of this test of the model indicated that the model did function as expected.

Having established that the model would perform as expected, procedures were implemented to establish its validity. The first phase of this process was to determine if the model did adequately represent intermodal railway terminal operations. Therefore, a meeting was held with several individuals that are expert in intermodal operations. The model was presented to them and each component discussed in detail. At the conclusion of the presentation, it was unanimously agreed that the model was representative of the intermodal terminal operations of their railroad. That is, the flow of transactions through the simulation model were representative of the movement of flatcars, containers, and tractors within a terminal system, and that the essential functions and events of the terminal had been provided for in the model. Consequently, there is reasonable evidence to suggest that the model developed during this research has a high degree of face validity.

The final phase in the validation of the model involves the comparison of the results of historical operations for an actual system with the results of a simulation of the same facility during the same period. To make this comparison, the operating records of the Norfolk and Western Railway facility at Roanoke, Virginia, were obtained. The system's parameters, as previously defined in Table 4, and their associated probability distributions were estimated from data extracted from these records and incorporated into the simulation model. The facility's train arrival and departure schedules for the fifty-nine day period, beginning on January 26, 1981, and ending on March 25, 1981, were then entered into the model as input data. The system was simulated for one hundred replications of the network to obtain the variability of the estimated mean values of the system's variables.

The output results of the simulation were compared to the actual operating results of the system during the same time period. For example, the number of flatcars and containers in the system were accurately accounted for by the simulation model. Likewise, the number of container offloads, and onloads, accomplished during the simulation were representative of the actual operation. Switching times and associated cost were also reasonably predicted by the simulation. Each of these tests, presented in detail in the example given under implementation in the next chapter, indicate that the model has a high degree of validity for the Roanoke terminal facility. It should be noted, however, that the model should be validated for each facility that it is intended to represent prior to its use in a decision analysis process.

Summary

The entities, events, activities, attributes, variables and interrelationships of an intermodal terminal facility were defined. The Q-GERT model logic was then presented, with detailed descriptions of the system's major components. These include the inbound yard, siding, outbound yard and the inbound and outbound sections of the parking lot. Illustrations were presented to demonstrate the manner in which transactions flow through the simulation network that represent flatcars, containers and trailers in an actual system.

The three FORTRAN written subroutines that are an integral part of the model were also discussed. The first is User Function

(UF), consisting of ten sections that provide the timing for a simulation, initiate activities, and enable the movement of the three separate types of transactions through the system. Subroutine UI is used to initialize the system's variables and establish the simulation's initial conditions. A most important function provided by this subroutine, however, allows the user to read into the program the train arrival and departure schedules. These schedules are required to drive the system. Subroutine UO is incorporated into the model to collect the results of each simulation run in order to calculate and print the estimated mean values of the system's variables and their associated variances as output for the simulation. This subroutine also enables the user to format output reports such that emphasis can be placed on particular variables of interest.

Model validation and testing was discussed. However, presentation of the comparison of a simulation's output results with a system's historical record of operations is presented in the following chapter. It was emphasized that validation should be accomplished for each terminal to be simulated.

CHAPTER IV

IMPLEMENTATION

A four phase process is recommended for the implementation of the intermodal railway terminal simulation model. This process includes: (1) obtaining the characteristics of the terminal to be modeled and the collection of pertinent data to define the facility, (2) modification of the simulation model to conform to the terminal under study, (3) comparing the simulation model output with known operational results of the system to establish the adequacy of the model and, (4) variation of selected system parameters in the model to estimate the expected change in the system under similar conditions. Phases three and four are illustrated through the use of an example of model implementation.

Phase I - System Characteristics

No two intermodal terminals are exactly alike. In general, however, they all possess certain common characteristics. These similar characteristics, discussed in Chapter II in terms of entities, attributes, events, and activities enable the modeler to define a particular terminal system by setting the model's input variables equal to the values inherent within the system without modification to the Q-GERT program. On the other hand, some of a system's characteristics are so unique that they will almost always require some modification in the Q-GERT input data statements, and an occasional change in the model logic. These characteristics are associated with the system's parameters that represent activity service times. The parameters are estimated by applying random sampling techniques, wherever possible, and are provided for during a simulation run by generating random variables from representative probability distributions.

The purpose of the first phase of model implementation is to identify a terminal's characteristics in order to specify the required values for the model's input variables and to define the appropriate probability distributions associated with the model. This section discusses model specification and the requirements for the input variables.

Input Variables

Primary sources of data for an intermodal terminal include the physical observations of the terminal, a review of the facilities' historical records, and interviews with individuals knowledgeable in the terminal's operation. The required information for the model that is gained through data collection includes switching rules, car selection, priorities, working rules, facility capacities, train arrival and departure schedules, in addition to activity service times. Table 8 lists the input variables that must be specified by the model user in subroutine UI.

There are several indicator and collector variables that are also initialized in subroutine UI. These variables are model dependent and remain constant for each terminal simulated. They are defined in Table 9.

TABLE 8

User Specified Input Variables

Variable	Definition
MXRAMP	The maximum number of flatcars permitted at the siding.
MXPARK	The maximum number of containers permitted in the regular parking lot.
IPRI1	IPRI1=0,1,2,3, or 4. IPRI1 sets the first priority for selecting cars to be switched from the inbound yard to the siding. 0 = Type car with largest number in yard. 1 = Type one cars. 2 = Type two cars. 3 = Type three cars. 4 = Type car waiting the longest in yard.
IPRI2	IPRI2=0,1,2,3, or 4 as defined in IPRI1. IPRI2 sets the second priority for selecting cars to be switched from the inbound yard to the siding if there are no IPRI1 cars available. If there are no IPRI1 or IPRI2 cars available, priority is automatically given to the type of car waiting the longest in the yard.
NGND	NGND=0 or 1. NGND sets the container consignee notification rule in effect at a terminal. 0 = Notification when container enters the yard. 1 = Notification after the container is offloaded.
ISAT	ISAT=0 or 1. ISAT establishes the Saturday work rule in effect at a terminal. 0 = no work. 1 = normal work day.
ISUN	ISUN=0 or 1 as defined for ISAT above. ISUN establishes the Sunday work rule in effect at a terminal.
IPARK1	IPARK1=0,1,2,...,M. IPARK1 initializes the number of type one inbound containers in the parking lot.
IPARK2	IPARK2=0,1,2,...,M. IPARK2 initializes the number of type two inbound containers in the parking lot.

TABLE 8 (continued)

Variable	Definition
IPARK3	IPARK3=0,1,2,...,M. IPARK3 initializes the number of type three inbound containers in the parking lot.
ISTART	ISTART=0,1,2,...,M. ISTART sets the total number of outbound containers to be placed into the outbound parking lot on initialization.
ITYP1E	ITYP1E=0,1,2,...,M. ITYP1E initializes the number of East outbound type one containers in the parking lot.
ITYP1W	ITYP1W=0,1,2,...,M. ITYP1W initializes the number of West outbound type one containers in the parking lot.
ITYP1N	ITYP1N=0,1,2,...,M. ITYP1N initializes the number of North outbound type one containers in the parking lot.
ITYP1S	ITYP1S=0,1,2,...,M. ITYP1S initializes the number of South outbound type one containers in the parking lot.
ITYP2E	ITYP2E=0,1,2,...,M. ITYP2E initializes the number of East outbound type two containers in the parking lot.
ITYP2W	ITYP2W=0,1,2,...,M. ITYP2W initializes the number of West outbound type two containers in the parking lot.
ITYP2N	ITYP2N=0,1,2,...,M. ITYP2N initializes the number of North outbound type two containers in the parking lot.
ITYP2S	ITYP2S=0,1,2,...,M. ITYP2S initializes the number of South outbound type two containers in the parking lot.
ITYP3E	ITYP3E=0,1,2,...,M. ITYP3E initializes the number of East outbound type three containers in the parking lot.
ITYP3W	ITYP3W=0,1,2,...,M. ITYP3W initializes the number of West outbound type three containers in the parking lot.
ITYP3N	ITYP3N=0,1,2,...,M. ITYP3N initializes the number of North outbound type three containers in the parking lot.
ITYP3S	ITYP3S=0,1,2,...,M. ITYP3S initializes the number of South outbound type three containers in the parking lot.

TABLE 8 (continued)

Variable	Definition
IEAST	IEAST=0,1,2,...,M. IEAST initializes the number of East bound flatcars in the outbound yard.
IWEST	IWEST=0,1,2,...,M. IWEST initializes the number of West bound flatcars in the outbound yard.
INORTH	INORTH=0,1,2,...,M. INORTH initializes the number of North bound flatcars in the outbound yard.
ISOUTH	ISOUTH=0,1,2,...,M. ISOUTH initializes the number of South bound flatcars in the outbound yard.
ATIME(1,1)	ATIME(1,1)=0.0. ATIME(1,1) is set equal to zero to enable the initialization of the inbound yard at simulation time TNOW=0.0.
ITP1(1,1)	ITP1(1,1)=0 or KLMN, where KLMN is a coded eight digit integer that specifies the configuration of type one cars on a train. ITP1(1,1) initializes the inbound yard for type one flatcars. ITP1(1,1) is automatically set to 0 unless KLMN is specified. K=00,01,02,...,99. K is the total number of the specified type flatcars on a train. L=00,01,02,...,99. L is the number of flatcars on a train that are loaded with two containers of the specified type. M=00,01,02,...,99. M is the number of flatcars on a train that are loaded with one container of the specified type. N=00,01,02,...,99. N is the number of empty flatcars on a train of the specified type. Example: K=10, L=05, M= 03, N=02 KLMN=10050302 Total Cars on Train 10 Cars with Two Containers 5 Cars with One Container 3 Empty Cars 2 Total Containers 13

TABLE 8 (continued)

Variable	Definition
ITP2(1,1)	ITP2(1,1)=0 or KLMN. ITP2(1,1) initializes the number and configuration of type two flatcars in the inbound yard. KLMN is coded as described above. KLMN is automatically set to 0 unless KLMN is specified.
ITP3(1,1)	ITP3(1,1)=0 or KLMN. ITP3(1,1) initializes the number and configuration of type three flatcars in the inbound yard. KLMN is coded as described above. KLMN is automatically set to 0 unless KLMN is specified.
ATIME(I,J)	ATIME(I,J)=0.0,1.0,...25.0, I=1,2,...,M, J=1,2,...,10. ATIME(I,J) specifies a scheduled train arrival time on day I and train J using the twenty-four hour clock. ATIME(I,J) is automatically set at 25.0 if no train arrival time is specified.
ITP1(I,J)	ITP1(I,J)=0 or KLMN, I=1,2,...,M, J=1,2,...,10. KLMN is defined above. ITP1(I,J) is the type one flatcar configuration of the train arriving at ATIME(I,J). ITP1(I,J) is automatically set to zero unless otherwise specified.
ITP2(I,J)	ITP2(I,J)=0 or KLMN, I=1,2,...,M, J=1,2,...,10. KLMN is defined above. ITP2(I,J) is the type two flatcar configuration of the train arriving at ATIME(I,J). ITP2(I,J) is automatically set to zero unless otherwise specified.
ITP3(I,J)	ITP3(I,J)=0 or KLMN, I=1,2,...,M. J=1,2,...,10. KLMN is defined above. ITP3(I,J) is the type three flatcar configuration of the train arriving at ATIME(I,J). ITP3(I,J) is automatically set to zero unless otherwise specified.
DTIME(I,J)	DTIME(I,J)=0.0,1.0,...25.0, I=1,2,...,M, J=1,2,...,10. DTIME(I,J) specifies a scheduled train departure time on day I and train J using the twenty-four hour clock. DTIME(I,J) is automatically set to 25.0 unless otherwise specified.
IDEPT(I,J)	IDEPT(I,J)=0 or KL, where KL is a four digit integer code that specifies the direction and maximum loading of the train departing at time DTIME(I,J). IDEPT(I,J) is automatically set to zero unless KL is specified.

TABLE 8 (continued)

Variable	Definition
IDEPT(I,J) (continued)	K codes the direction and L codes the flatcar capacity of a departing train. K=1,2,3, or 4 1=East 2=West 3=North 4=South L=000,001,...,999, the number of flatcars permitted on train Example: K=2, L=025 KL=2025 Direction of Departure West Maximum Number of Cars 25

TABLE 9

Constant Input Variables

Variable	Initialized Value	Definition
IDAY	1	IDAY=1,2,...,M. An indicator variable representing the current day during a simulation.
IHOUR	0	IHOUR=0,1,...,24. An indicator variable representing the current hour of IDAY.
NSWIN	0	NSWIN=0,1,...,M. An indicator variable recording the current flatcar inbound switching activity during IDAY.
NOFF	0	NOFF=0,1,...,M. An indicator variable recording the current container offload activity during IDAY.
NON	0	NON=0,1,...,M. An indicator variable recording the current container onload activity during IDAY.
NOUT	0	NOUT=0,1,...,M. An indicator variable recording the current flatcar outbound switching activity.
ITCARS	0	ITCARS=0,1,...,M. A collector variable recording the total number of flatcars entering system.
ITONE	0	ITONE=0,1,...,M. A collector variable recording the total number of flatcars entering the system with only one container.
ITTWO	0	ITTOW=0,1,...,M. A collector variable recording the total number of flatcars entering the system with two containers.
ITEMT	0	ITEMT=0,1,...,M. A collector variable recording the total number of empty flatcars entering the system.

TABLE 9 (continued)

Variable	Initialized Value	Definition
ITONER	0	ITONER=0,1,...,M. A collector variable recording the total number of reversed flatcars with one container entering the system.
ITTWOR	0	ITTWOR=0,1,...,M. A collector variable recording the total number of reversed flatcars with two containers entering the system.
ITEMTR	0	ITEMTR=0,1,...,M. A collector variable recording the total number of reversed empty flatcars entering the system.

The assignment of specific values to the variables listed in Table 8 is rather straight forward and develops naturally as information is gained about the system. The train arrival and departure parameters can, of course, represent historical experience, or a hypothetical future projection. Likewise, siding and parking lot capacities, switching priorities, weekend work rules, and container consignee notification rules can be varied to reflect actual, or anticipated, changes without modification to the Q-GERT program. However, any system characteristic that is time persistent, with the exception of the train arrival and departure schedules, requires specification through Q-GERT parameter set data statements and possible modification to the model's logic. These time persistent characteristics and related variables are discussed below.

Time Persistent Characteristic Input Requirements

The time persistent characteristics of a terminal are provided for in three different ways in the model. First, the time of day rules to enable switching and siding activities are incorporated into the user function logic. Secondly, activities that represent the movement of tractor/trailer combinations within the terminal's parking area are specified by Q-GERT input data statements for activities connecting one node to another. These statements usually require corresponding parameter specification statements. Finally, activities that require the movement of flatcars, or the offloading and onloading of containers, are specified in the user function logic. The activities also include any time delays that must be invoked because of time of day work restrictions. These user function specifications also require corresponding Q-GERT parameter specification data input statements.

Each of the time dependent characteristics of a terminal must be identified during data collection and appropriately provided for in the model. With the exception of the time of day working rules, the parameters of the activity service times, arrival rates, or time delays should be estimated and their inherent probability distributions identified. The probability distributions are then stipulated for each type of activity in the user function, or on the activity input data statements, and the parameters of the distributions are provided for in the parameter set statements. Time of day working rules are provided for in the user function logic. Table 10 illustrates the time dependent characteristics that are provided for in the model. The table also indicates where the corresponding variables, statements, and parameter specifications are located in the computer program that has been described in previous chapters.

It should be noted that the expected probability distributions, and their corresponding parameter specifications, associated with the FORTRAN variables and Q-GERT activities listed in Table 10 are related to the terminal that serves as the example for this research only. One would expect that any, or all, of the distributions could change from terminal to terminal. Most assuredly, the parameter specifications will change due to the different physical characteristics of each terminal. Therefore, it is highly recommended that whenever the data is

TABLE 10

Intermodal Terminal

Time Dependent Characteristics

Characteristics	Model Variable	User Function Statement Number	Q-GERT Input Data Activity Statement Node-Node	Expected Probability Distribution	Parameter Set Data Input Number	Parameter Specification for Expected Probability Distribution
Time of day Inbound/Outbound Switching Rule		156 185		Deterministic		A logical IF Statement Testing IHOUR
Time of day Offload/Onload Work Rule		149		Deterministic		A logical IF Statement Testing IHOUR
Train Arrival Distribution Based on Scheduled Arrival Time	ACTUAL	104		Normal	1	μ, a, b, σ

TABLE 10 (continued)

Characteristics	Model Variable	User Function Statement Number	Q-GERT Input Data Activity Statement Node-Node	Expected Probability Distribution	Parameter Set Data Input Number	Parameter Specification For Expected Probability Distribution
The Number of days a road departing container is "on the street" before returning to the parking lot	IRTCON	950		Exponential	2	μ,a,b
Inbound flatcar Switching Time	TIMESW	204 310		Normal	3	μ,a,b, σ
Outbound flatcar Switching Time	TIMEOU	156 600		Normal	3	μ,a,b,σ
Container Offload Start Time	TIMEOF	151		Poission (75%)	4	λ,a,b
				Uniform (25%)	5	a,b

TABLE 10 (continued)

Characteristics	Model Variable	User Function Statement Number	Q-GERT Input Data Activity Statement Node-Node	Expected Probability Distribution	Parameter Set Data Input Number	Parameter Specification For Expected Probability Distribution
Container Onload Start Time	TIMEON	153		Probabilistic		
Consignee Container Pick-up Time of Day	PICKUP	411		Uniform	6	a,b
Consignee Container Return Time of Day	DELV	965		Uniform	6	a,b
Container Offload Service Time	UNTIME	400 410		Exponential	7	μ,a,b

TABLE 10 (continued)

Characteristics	Model Variable	User Function Statement Number	Q-GERT Input Data Activity Statement Node-Node	Expected Probability Distribution	Parameter Set Data Input Number	Parameter Specification For Expected Probability Distribution
Consignee Container Pick-up Time of Day for Initial Container in Lot	SPICUP	178		Uniform	8	a,b
Container Onload Service Time	OLTIME	511 515		Exponential	9	μ,a,b
Service Time to Turn a Reversed Car			36-37	Normal	10	μ,a,b, σ
Service Time to Spot Over-The-Road Container Arrival in Lot			41-42	Normal	11	μ,a,b, σ

TABLE 10 (continued)

Characteristics	Model Variable	User Function Statement Number	Q-GERT Input Data Activity Statement Node-Node	Expected Probability Distribution	Parameter Set Data Input Number	Parameter Specification For Expected Probability Distribution
Waiting Time for Tractor Pick-up at Parking Lot			60-58	Normal	12	μ,a,b, σ
Service Time for Tractor/ Trailer Departure from Parking Lot			53-61 55-62 57-63	Normal	13	μ,a,b, σ
Consignee Notification Time of Day for Container Arrival	XNOTFI	411		Uniform	14	a,b
Time Between Over-The-Road Arrivals			40-40	Exponential	15	μ,a,b

Legend: μ =mean, a=minimum observation, b=maximum observation, σ=standard deviation, $\lambda=\mu-a$

available, and time permits, appropriate goodness-of-fit test be applied to fit a distribution to its respective set of data. Reliance upon the professional judgement of experts knowledgeable of a terminal's operations should provide an acceptable substitute for those distributions where the data is not available, its collection is impractical, or the cost of obtaining the data is prohibitive.

Phase II - Model Modification

The modification of the model to simulate the operations of a particular intermodal terminal for any period of time can be accomplished by following a four step procedure. The first step is required only if the number of days to be simulated exceeds ninety, and there are more than ten train arrivals and departures per day. Steps two, three, and four insure that the variables, probability distributions, activities, and parameter specifications identified in Phase I are properly implemented.

Step One

The FORTRAN subroutines UI, UO, and user function UF(IFN) have several arrays in common. These arrays in the model developed for this research are dimensioned such that a total of ninety days of terminal operations can be simulated with a provision for ten train arrivals and ten train departures per day. If a simulation is to be run which exceeds these limitations, then the arrays must be redimensioned. Table 11 is included to indicate the affected arrays. It should be noted that all of the common blocks are included in each of the subroutines.

TABLE 11

Arrays in Common

Labeled Common	Affected Array
UCOM1	ATIME(I,J) ITP1(I,J) ITP2(I,J) ITP3(I,J) DTIME(I,K) IDEPT(I,K) IPARKT(I) IOVP(I) MOVE(I)
UCOM2	TSWIN(I,M) ISWINT(I,M) TOFF(I,M) TON(I,M) LCON(I,M) NSWOUT(I,M) TSOUT(I,M) LCOFF(I,M)

I=Maximum days that can be simulated
J=Maximum train arrivals per day
K=Maximum train departures per day
M=Maximum in-switch, out-switch, offload, or onload operations per day.

Step Two

The values for the input variables listed in Table 8 above should be entered into subroutine UI once it has been determined that the arrays are properly dimensioned for the simulation. It should be noted that particular care is required when entering the train arrival and departure schedule. Each scheduled arrival time, ATIME(I,J), must have at least one corresponding ITP1(I,J), ITP2(I,J), or ITP3(I,J) defined and properly coded. Likewise, each scheduled train departure time, DTIME(I,K) must have an accompanying IDEPT(I,K) variable defined and coded.

Step Three

Step three is probably the most difficult to implement in the model modification process. Each FORTRAN variable that can assume a value randomly drawn from a Q-GERT provided probability distribution must be located in user function UF(IFN) and the sampling distribution changed where required. Additionally, the model's FORTRAN logic could require some modification if the time of day switching and the siding work rules are changed. Therefore, Table 10 should be referred to in order to identify the approximate FORTRAN statement numbers where specific variables and logic statements can be located in the user function.

It should be noted that Table 10 does not contain an inclusive list of all the probability distributions that are included in the Q-GERT simulation. For a formal discussion of the options available in the language, the book "Modeling and Analysis Using Q-GERT Networks" by Pritsker [10] is recommended.

Step Four

The final step in the model modification process involves changing the Q-GERT data input statements where required. The first input statement to change defines the simulation time per run, and the number of runs of the network for a complete simulation. These changes are accomplished by specifying the time to end one run of the network in field ten, and the number of network runs in field eleven in the GEN (General) data input statements of the Q-GERT model. It should be remembered that one unit of simulation time represents one hour, of one day, in the model. Therefore, if it is desired to simulate thirty days of terminal operations, field ten of the GEN statement should contain the value of 720.0 (30 days times 24 hours per day). Multiple runs of the network are also recommended. At the end of each run, averages are calculated for each of the output variables. These averages are estimates of the mean values for the variables. By calculating the standard deviation of these averages over multiple runs, estimates of the variance for these averages are obtained.

The seven Q-GERT data input statements that specify probability distributions for the time to turn a reversed car, the time between random over-the-road container arrivals, and the time required to move containers within the parking lot should also be modified where necessary. The activities for these data statements are identified in Table 10, and they are located in the Q-GERT data input section of the program following the GEN statement.

The PAR (parameter specifications) Q-GERT input data statements require modification once each of the probability distribution types have been identified or estimated. There is a PAR statement for each distribution employed in the user function, or in the network model. Generally, the requirement is that the user must specify the expected mean value for each distribution, including the estimated range and standard deviation of the distribution. The PAR statements are located near the end of the input data statements which describe the network, following the GEN statement. An example of the manner in which these statements are employed in the model is included in the illustration below.

Phase III - An Example of Implementation

The Q-GERT intermodal railway terminal model is designed such that transactions do not proceed through the network unless they are specifically enabled to do so. It is the status of the system, at any point in time that establishes the enabling conditions for transaction flow, and determines the number of transactions that can move from one node to another. Consequently, the usual concept of transactions flow in Q-GERT modeling is not applicable in this model. Normally, transactions enter the system at a node, wait for processing on activities, proceed through following nodes and eventually pass through the system in a continuous movement. In this model, transactions enter the system and wait until they are allowed to be processed. The movement is not continuous. Therefore, in the description of activities for the system discussed below, it should be understood that transaction movement is directed by the conditions present in the network and not on the status of servers following Q-nodes.

Scenario

The example of implementation of the model is based on the operating experience of the Roanoke, Virginia, intermodal terminal of the Norfolk and Western Railway from January 26, 1981, through March 25, 1981. During this period, five hundred nine (509) flatcars of various container loading configurations entered the terminal for processing. Seven hundred ninety-three (793) containers were offloaded and eight hundred seven (807) containers were onloaded at the facility. The onloaded containers were shipped by flatcars from the terminal to various locations within the Norfolk and Western System.

The inbound and outbound yards for the terminal, though of a finite capacity, are not considered to be limiting factors for the simulation. However, the siding has a maximum capacity of eleven (11) cars. Likewise, the container parking lot's maximum capacity is seventy-five (75) standard over-the-road type containers. The facility, on occasion, processes Type 2 and Type 3 containers, but these are exceptions that require special handling. For the purposes of this illustration, it is assumed that all of the cars and containers in the system are Type 1, which is, in fact, the type of cars and containers in the system for the period to be simulated.

The switching rules in effect at the terminal for the level of activity during the fifty-nine day period essentially provides for two in-switches and two out-switches for the siding per day. Flatcars are switched onto the siding such that they are available for offloading at eight AM each working day. When the offloading is complete, onloading commences. Upon completion of the activity, the cars are moved to the outbound yard and additional cars are switched to the siding. Usually, the offloading and onloading service times permit a switch from the siding at approximately noon each day, with additional loaded cars being available for offloading at the siding in the early afternoon. When these cars have been offloaded and onloaded, they are available to be switched to the outbound yard during the evening, and loaded cars moved to the siding prior to eight AM the following day. Priority for selecting cars at the inbound yard for switching is given to the type of cars with largest number in the yard.

Working rules for the siding and parking lot operation allow for containers to be offloaded and onloaded between the hours of eight AM and six PM daily. However, this operation is only permitted Monday through Friday of each week. Container pick up and delivery can be accomplished during these hours, and consignee notification of a container's arrival is not made until a container is on the ground. Immediately prior to eight AM on January 26, there were eighteen inbound, and twenty-two outbound containers in the parking lot. Additionally, there were eighteen flatcars with a total of eighteen containers in the inbound yard. Eight of the cars in the yard were empty. There were also ten cars in the outbound yard waiting for departure.

A final observation about the operations at the Roanoke facility is that random over-the-road container arrivals are rare. This is not uncommon in areas that are primarily serviced by a single, or a predominate, railroad. The situation is such that once a customer begins to utilize the railroad's intermodal services in these areas, a cycle develops. For example, containers are delivered to the railroad for shipment and eventually return to the terminal for the customer to pick up. The containers can be loaded or empty when they arrive at the terminal from either source. Assume that an empty container is at the terminal waiting for customer pickup. After it is picked up by the customer, it is loaded, then returned to the terminal for rail shipment. The container is shipped out of the terminal, and will eventually return to the terminal by rail empty, and the cycle continues. Therefore, there are always containers "on the street" which will return to the terminal system in due course. This cycle is provided for during a simulation by scheduling a container's return to the terminal as it departs over-the-road. There were seventy-nine (79) containers on the street just prior to January 26. They are provided for in the simulation.

The Initial Values for the Input Variables

Subroutines UI, UO, and user function UF(IFN) are dimensioned through their common statements to accommodate ninety days of operations with up to ten train arrivals and departures per day. Additionally, ten switching and siding operations have been provided for. Historical records indicate that the maximum number of train arrivals with intermodal traffic for the fifty-nine day period to be simulated was five per day. The maximum number of train departures per day was nine. There were no more than two switching and siding operations per day. Therefore, sufficient computer CPU region is available to simulate the terminal for this period. The initial values for the input variables are listed in Table 12.

TABLE 12

Roanoke Terminal Input Variables

Variable	Initial Value*
MXRAMP	11
MXPARK	75
IPRI1	1
IPRI2	0
NGND	1
ISAT	0
ISUN	0
IPRK1	18
ISTART	22
ITYP1E	1
ITYP1W	19
ITYP1N	2
IWEST	8
INORTH	2
ATIME(1,1)	0.0
ITP1(1,1)	18080208
ATIME(1,2) through ATIME(59,10)	**
ITP1(1,2) through ITP1(59,10)	**
DTIME(1,1) through DTIME(59,10)	***
IDEPT(1,1) through IDEPT(59,10)	***

*All variables in Table 8 not listed in Table 12 are initialized at zero.

**There are a total of one hundred seventy-nine (179) individual train arrivals during the fifty-nine day simulation period. The coding for each ATIME(I,J) and ITP1(I,J) variable is listed under Subroutine UI in APPENDIX C. ATIME(1,1) and ITP1(1,1) are reversed to initialize the inbound yard.

***There are six departing trains that operate on a six day schedule and an additional three trains that operate seven days a week. The coding for each departure is listed in Subroutine UI in APPENDIX C. IWEST and IEAST are used to initialize the outbound yard.

Activity Descriptions

There are fourteen activities that are accounted for during a simulation of the Roanoke terminal facility operation. In addition, the scheduled train arrival times are adjusted to allow for deviations in the arrival time of individual trains. This is actually an event occurrence time, but is included as an activity because it represents the activity ending time that would be ordinarily associated with arrival rates and activities in a simulation model. Wherever possible, the distributions associated with each activity have been fitted to data available from the historical records, or obtained through observation, of the Roanoke facility using Chi-square goodness-of-fit test procedures. In the instances where data was not available, and physical observation impracticable, reliance has been placed in the professional judgement of managers who are familiar with the terminal. The activity descriptions are presented in Table 13. The activities that have been tested for appropriateness of their assigned distributions are identified by an asterisk (*) following the distribution type.

Referring to Table 13, historical records indicate that train arrivals are recorded at the nearest hour of arrival, while experience indicates that arrivals do vary around this time. Hence, it is assumed that all trains arrive on the hour reported, plus, or minus, a one-half hour deviation which is approximated by a normal distribution. The exponential distribution associated with container street time is more difficult to defend, however, since no records exist to

TABLE 13

Activity Descriptions

Parameter Set No.	Activity Description	Time in Hours			Standard Deviation	Distribution
		MIN	MAX	Mean		
1	Train Arrival Time	-0.5	0.5	0.0	0.167	Normal
2	Container Street Time	48.0	240.0	96.0		Exponential
3	Inbound and Outbound Flatcar Switching Time	0.1275	0.1725	0.1485	0.004	Normal*
4	Container Offload Start Time (75% of the Observations)	0.0	4.0	0.22		Poisson*
5	Container Offload Start Time (25% of the Observations)	4.0	7.0			Uniform*
6	Returned Container Delivery Time	0.0	9.0			Uniform*
7	Time to Offload a Container	0.05	0.334	0.1025		Exponential*

TABLE 13 (continued)

Parameter Set No.	Activity Description	Time in Hours MIN	Time in Hours MAX	Mean	Standard Deviation	Distribution
8	Consignee Container Pickup Time	8.0	17.0			Uniform*
9	Time to Onload a Container	0.067	0.2667	0.105		Exponential*
10	Time to Turn Reversed Car					
11	Tractor Disconnect/ Park Time	0.05	0.4	0.2	0.06	Normal
12	Tractor Wait Time	0.1	1.0	0.5	0.15	Normal
13	Container Hookup Time	0.1	1.0	0.2	0.15	Normal
14	Consignee Notification Time	10.	4.0			Uniform*

*Fitted distribution using Chi-square goodness-of-fit test.

support the assumption. The intermodal managers estimate that, on the average, containers are on the street from between two and four days before they are returned. On occasion the containers are delayed as much as ten days before they are returned. Therefore, the distribution selected seems reasonable under the circumstances and experience indicates that the model is not overly sensitive to this assumption. Likewise, no records exist for tractor wait, disconnect/park, and hookup times. The reversed car provision is not used in this simulation because the physical layout of the siding at the Roanoke terminal does not require cars to be turned.

Simulation Output and Model Validation

The Q-GERT Intermodal Terminal Model is intended to provide managers with information about their system that can be incorporated into their decision process. In order for the model to provide a meaningful contribution to this process, the manager should be able to evaluate the degree of confidence that can be placed in the model. One way to establish a level of confidence is to use the model to simulate recent operations and then compare the simulated results with the actual results experienced by the terminal under study. The discussion that follows illustrates one way the comparison can be made, and the validity of the model estimated for the system variables.

Generally, intermodal managers are interested in operating costs. These costs, however, are dependent upon other variables associated with the system. For example, an intermodal terminal is charged a per diem fee for the time a flatcar is in the system. This charge begins when the car enters the inbound yard, and terminates when the car departs the outbound yard. Similar charges are made for containers. There are two costs associated with the flatcars in the system. The first is related to the average time a car spends in the system. The second is related to the average number of cars in the system per day. Again, like costs are associated with containers. However, there are two categories of containers; inbound and outbound. Each category of containers follows its own path through the system. Therefore, the model is designed to first estimate the average values for the system's variables, then the cost estimations are computed based on these values. Comparisons can then be made based on actual operating experience and cost incurred.

The system variables provided for in the model are listed in Figure 12. Figure 12 is the actual user written output for the model after fifty-nine days of simulation time consisting of one hundred independent simulation runs. The system's input variables and parameter specifications were as previously described in this section. The average value for each variable is collected at the end of each run, then averaged over all runs. Their variance is then estimated and the standard deviation obtained. The standard deviation of the averages is then estimated. For example, after one hundred simulation runs the estimated average number of flatcars in the system per day is 38.94. The standard deviation of the averages used to obtain this value is 0.04 flatcars. Therefore, one could expect the average number of flatcars in the system to be very close to the estimated average, given that there is a high degree of confidence in the model.

```
                         SYSTEM VARIABLES

                                                            SD
                                      AVE         SD      OF AVE
UNITS IN THE SYSTEM
    (PER DAY)

FLATCARS                             38.94       0.36      0.04
INBOUND CONTAINERS                   71.21       0.35      0.03
OUTBOUND CONTAINERS                  52.41       2.12      0.21


TIME IN THE SYSTEM
    (PER UNIT)

FLATCARS                             82.33       0.43      0.04
INBOUND CONTAINERS                  133.79       0.49      0.05
OUTBOUND CONTAINERS                  42.57       2.12      0.21


TOTAL UNITS IN SYSTEM
          (PER RUN)

FLATCARS                            509.00       1.80      0.18
INBOUND CONTAINERS                  890.09       3.60      0.36
OUTBOUND CONTAINERS                 848.57       3.19      0.32


SIDING OPERATIONS
   (CONTAINERS)

OFFLOADS PER WORK DAY                18.44       0.08      0.01
ONLOADS PER WORK DAY                 18.77       0.05      0.00

OFFLOAD TIME/UNIT (HOURS)             0.11       0.00      0.00
ONLOAD TIME/UNIT (HOURS)              0.11       0.00      0.00

TOTAL OFFLOADS PER RUN              793.00       0.70      0.07
TOTAL ONLOADS PER RUN               807.49       5.05      0.51


SWITCHING TIMES
    (HOURS)

INBOUND TIME PER CAR                  0.15       0.00      0.00
OUTBOUND TIME PER CAR                 0.15       0.00      0.00

TOTAL TIME PER WORK DAY               3.28       0.01      0.00
TOTAL TIME PER RUN                  141.23       0.69      0.07


CONTAINER PARKING
    (PER DAY)

REGULAR LOT                          49.11       2.28      0.23
OVERFLOW LOT (WHEN USED)              5.29       2.26      0.24
TOTAL PARKING                        49.37       2.42      0.24
DAYS OVERPARKING OCCURRED             3.14       1.85      0.19


INBOUND YARD

CARS IN YARD                         42.37       1.12      0.11
MAXIMUM IN YARD                      57.88       0.46      0.05
```

FIGURE 12

THE RESULTS OF 59 DAYS SIMULATION AFTER 100 RUNS

The validity of the model can be established by comparing the output results of the simulation to historical results based on actual operations for the terminal. This process is accomplished by estimating the validity of individual components of the system. It is assumed that if each component of the model is valid, then by implication, the model as a whole is also valid. In order to make some of the comparisons, however, it is necessary to rely on the Q-GERT generated output of a single simulation run, in addition to the user written model output for one hundred runs. Figure 13 illustrates the Q-GERT output for a single run during the simulation.

The validation of train arrival section of the model is a straight forward process. Historical records show that 509 cars entered the system during the fifty-nine days simulated by the model. The total average number of flatcars in the system during the simulation, as reported in Figure 12, is 509, with a standard error of 0.18. Consequently, it can be assumed that the train arrival section of the model adequately represents the system.

Concluding that the train departure section of the model represents the system is somewhat more difficult to demonstrate. However, if it is recognized that every car that enters the system must eventually leave the system, then intuitively, the number of car departures from the system added to the number of cars remaining in the system at the end of a simulation run should equal the number of cars that entered the system. Analysis of the Q-GERT output for a single run in Figure 13 indicates that 447 cars departed the system through nodes 28, 30, 32, and 39. There are 62 cars remaining in the system at the end

```
              GERT SIMULATION PROJECT INTMOD YARD  BY HAMMESFAHR
                            DATE  2/ 18/ 1981

                   **FINAL RESULTS FOR FIRST SIMULATION**

                      TOTAL ELAPSED TIME =   1415.9900

           **NUMBER IN Q-NODE**                            ** WAITING TIME **
                                                                IN QUEUE

NODE    LABEL        AVE.      MIN.   MAX.    CURRENT       AVERAGE
                                              NUMBER

13      YARD1L      17.0443     0.     39.      17          51.2411
14      YARD2L       0.0        0.      0.       0           0.0
15      YARD 3L      0.0        0.      0.       0           0.0
16      YARD 1E     15.2061     0.     30.      21         439.4226
17      YARD 2E      0.0        0.      0.       0           0.0
18      YARD 3E      0.0        0.      0.       0           0.0
37      REV CARS     0.0        0.      0.       0           0.0
21      UNLOAD       6.6307     0.     10.      10          19.5604
23      EMPTY        0.5997     0.     10.       0           1.8105
25      ONLOAD       2.0854     0.     10.       0           6.2963
27      EAST         0.0413     0.      2.       0           1.6256
29      WEST         1.2852     0.     10.       6           5.1406
31      NORTH        0.1322     0.      3.       2           2.9254
33      SOUTH        0.0        0.      0.       0           0.0
70      OVERLOAD     0.5898     0.     29.       0           3.2001
43      TP1 OUT     28.0188     0.     58.      20          25.4976
44      TP2 OUT      0.0        0.      0.       0           0.0
45      TP3 OUT      0.0        0.      0.       0           0.0
49      TP1 IN      11.6377     0.     35.      18          19.8781
50      TP2 IN       0.0        0.      0.       0           0.0
51      TP3 IN       0.0        0.      0.       0           0.0
58      TR ARR       0.0        0.      0.       0           0.0
53      ONE OUT      0.0659     0.      4.       0           0.1151
55      TWO OUT      0.0        0.      0.       0           0.0
57      THREE OU     0.0        0.      0.       0           0.0
```

FIGURE 13

SELECTED Q-GERT OUTPUT FOR ONE RUN

```
NODE     TRANSACTION
           PASSAGES

  1         1416
  2           58
  3           36
  4          354
  5           64
 10          509
 11          471
 12           38
 13          454
 14            0
 15            0
 16           28
 17            0
 18            0
 19          480
 20          480
 21          470
 22          469
 23          469
 24          469
 25          469
 26          469
 27           36
 28           35
 29          348
 30          347
 31           62
 32           61
 33            0
 35           15
 37            0
 39            4
 40            1
 41          751
 42          848
 43         1536
 44            0
 45            0
 49          811
 50            0
 51            0
 52          811
 53          811
 55            0
 57            0
 58          811
 60          812
 61          810
 70          261
 80         3001
```

```
       PRINTOUT OF ONGOING ACTIVITIES
                AT    1415.99

ACTIVITY END       END         ACTIVITY
    TIME           NODE         NUMBER

   1416.00           1             0
   1416.01          61            20
   1416.34          58             0
  10000.00          40             0
  11291.93          19             4
  11403.40          46            11
  11410.15          30            16
  11410.15          32            17
  11410.15          28            15
  11411.95          22             8
  11411.95          19             1
```

FIGURE 13 (continued)

of this simulation run. These are represented by the current number of transactions in Q-nodes 13, 16, 21, 29, and 31, in addition to one transaction in service on each of the following activities. There is also one car inservice on activity 15 behind Q-node 27. Therefore, 509 cars are accounted for during the simulation. By implication, it can be assumed that the train departure section of the model is representative of the system.

To validate the model for the number of inbound containers in the system, it is first demonstrated that the number of containers offloaded at the siding during a simulation are representative of the number reported in the system's historical records. The records indicate that 793 containers were offloaded from January 26 through March 25. The estimated average number of container offloads per run reported in Figure 12 after one hundred simulations is 793.0, with a standard error of the mean of 0.07 containers. A 95% confidence interval for the number of container offloads per run is 793.0 ± 0.1372. Since this interval contains the value 793, the model is producing valid estimates of the number of containers that are offloaded per run.

There were 890 inbound containers in the system during the fifty-nine day period according to the terminal's historical records. These include 854 containers that arrived on inbound trains during the period, 18 on cars in the inbound yard, and 18 in the inbound parking lot just prior to January 26. The simulation's output estimates the average total number of containers in the system during the period to be 890.09. The standard error of this estimate is 0.36. A 95% confidence interval for the estimate is 890.09 ± 0.7056. The interval contains the value of 890. Therefore, valid estimates of the total number of containers in the system are being provided by the simulation.

A total of 807 containers were loaded onto flatcars at the terminal during the period under study. The simulation estimates that an average of 807.49 containers were onloaded with the standard error of 0.51. A 95% confidence interval for the estimate is 807.49 ± 0.9996. The 807 cars actually loaded during the period is within the interval. The simulation also estimates that there were an average of 848.57 total outbound containers in the system with a standard error of 0.32. Historical records indicate that in addition to the 807 containers that were onloaded, there were 22 containers in the outbound parking lot, and 20 containers loaded on 10 cars in the outbound yard just prior to January 26. Therefore, the actual number of outbound containers in the system during the period was 849. A 95% confidence interval for the estimated number of containers in the system is 848.57 ± 0.6272. The 849 outbound containers actually in the system is within this interval. Therefore, the simulation is providing valid estimates for both the number of outbound containers loaded onto flatcars and the total number of outbound containers in the system for the fifty-nine day period.

The estimated average number of container onloads and offloads per work day is readily confirmed through the information provided above. The onload and offload times per container represent their specified mean values as provided for in their respective parameter

Q-GERT input data statements for the model. This is also true of the reported estimated average switching times for the flatcars.

There are no records available to validate the estimated values for the remaining system variables listed in Figure 12. These include the estimated average number of flatcars and containers in the system per day, the average time units are in the system, total switching times, the average number of containers in the parking lot, and the number of flatcars in the inbound lot. Validation of these variables depends upon the experience and professional judgement of intermodal managers familiar with the system. Managers were asked to compare the simulation output values for these variables with their knowledge about these variables based on the level of daily terminal activity during the period at the Roanoke facility. In each instance, it was confirmed that the simulation's output was representative of the actual system.

The managers expressed a high degree of confidence in the model's ability to approximate the Roanoke terminal for the fifty-nine day period and expressed equal confidence in utilizing the model as an analytical tool. This appraisal, coupled with the validation of the model's performance where recorded information is available for the system's variables, indicates that the model is valid for the Roanoke Intermodal Terminal.

Phase IV - Model Application

The variable costs incurred by an intermodal terminal are primarily dependent on the time units are in the system, and the switch ing time required to move flatcars to and from the siding. There are also variable costs associated with the loading and unloading of containers, but at the Roanoke terminal these operations are handled on a contract basis and are considered fixed cost for the term of the contract. In this section, some of these costs are analyzed. This is followed by some suggested uses of the model to predict the changes to anticipate in the system should certain actions be taken to reduce cost.

Current Operations

Figure 14 illustrates some of the typical cost incurred by the Roanoke terminal during the fifty-nine day operating period. These are estimated costs based on the estimated values of the corresponding system's variables reported as simulation output in Figure 12. Analysis of these costs indicates that the switching cost incurred during the period is significant. The estimated daily cost of switching is more than twice the estimated cost of the inbound, or outbound, containers in the system per day, and nearly double the estimated cost of the average number of flatcars in the system. A second observation is that the individual cost for a flatcar, or inbound container, in the system is over three times the cost of an outbound container. The cost per flatcar, however, represents the cost of loaded flatcars in the system only. Further analysis of the system is possible by referring to the standard Q-GERT output for one hundred simulations reported in Figure 15.

The arrival rate of flatcars in the system is approximately 8.62 cars per day and there are an estimated 1.75 containers per flatcar. Additionally, there are about 10.5 flatcars processed at the siding per

```
                          THE RESULTS OF  59 DAYS SIMULATION
                                   AFTER 100 RUNS

                                   ESTIMATED COST

                                                                              SD
                                                  AVE            SD         OF AVE
FLATCARS

COST PER CAR                             $       34.58          0.18          0.02
COST PER DAY                             $      392.48          3.65          0.37

INBOUND CONTAINERS

COST PER CONTAINER                       $       29.03          0.11          0.01
COST PER DAY                             $      370.27          1.82          0.18

OUTBOUND CONTAINERS

COST PER CONTAINER                       $        9.24          0.46          0.05
COST PER DAY                             $      272.54         11.01          1.10

SWITCHING AND CLASSIFICATION

INBOUND PER CAR                          $       33.66          0.11          0.01
OUTBOUND PER CAR                         $       33.59          0.07          0.01
COST PER WORK DAY                        $      742.94          2.56          0.26
COST PER RUN                             $    31946.50        155.86         15.59
```

FIGURE 14

ESTIMATE OF SELECTED COST AFTER 100 SIMULATION RUNS

```
                              GERT SIMULATION PROJECT INTMOD YARD   BY HAMMESFAHR
                                              DATE  2/ 18/ 1981


                      **FINAL RESULTS FOR  100 SIMULATIONS**


                  **AVERAGE NUMBER IN Q-NODE**                                          **AVERAGE WAITING TIME**          **NUMBER IN Q-NODE**
NODE   LABEL        AVE.     STD.DEV.  SD OF AVE     MIN.      MAX.        AVE.     STD.DEV.  SD OF AVE              MAX.

13     YARD1L     17.0241    0.0492    0.0049      17.0008   17.0539      51.1801    0.1706    0.0171               39.0000
14     YARD2L      0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
15     YARD 3L     0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
16     YARD 1E    14.8264    1.0930    0.1093      13.4199   17.0872     427.7700   23.1158    2.3116               32.0000
17     YARD 2E     0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
18     YARD 3E     0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
37     REV CARS    0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
21     UNLOAD      6.6350    0.0344    0.0034       6.5821    6.6636      19.5524    0.0595    0.0059               10.0000
23     EMPTY       0.6216    0.0158    0.0016       0.5885    0.6566       1.8747    0.0467    0.0047               10.0000
25     ONLOAD      2.0668    0.0200    0.0020       2.0044    2.1173       6.2330    0.0557    0.0056               10.0000
27     EAST        0.0403    0.0124    0.0012       0.0083    0.0785       1.6324    0.3846    0.0385                4.0000
29     WEST        1.2820    0.0338    0.0034       1.2188    1.3844       5.1386    0.0298    0.0030               10.0000
31     NORTH       0.1292    0.0202    0.0020       0.0826    0.1736       2.8068    0.2278    0.0223                6.0000
33     SOUTH       0.0       0.0       0.0          0.0       0.0          0.0       0.9045    0.0905                0.0
70     OVERLOAD    0.6683    0.2811    0.0281       0.1904    1.6302       3.0961    0.7024    0.0702               38.0000
43     TP1 OUT    29.3957    2.0179    0.2018      25.3662   35.5131      25.9883    2.0510    0.2051               68.0000
44     TP2 OUT     0.0       0.0       0.0          0.0       0.0          0.0       0.9374    0.0937                0.0
45     TP3 OUT     0.0       0.0       0.0          0.0       0.0          0.0       0.9847    0.0985                0.0
49     TP1 IN     11.5622    0.2207    0.0221      10.9959   12.1833      19.7491    0.3769    0.0377               37.0000
50     TP2 IN      0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
51     TP3 IN      0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
58     TR APR      0.0001    0.0005    0.0001       0.0       0.0051       0.0001    0.0009    0.0001                3.0000
53     ONE OUT     0.0618    0.0062    0.0006       0.0465    0.0753       0.1075    0.0108    0.0011                9.0000
55     TWO OUT     0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
57     THREE OU    0.0       0.0       0.0          0.0       0.0          0.0       1.0050    0.1005                0.0
```

<u>SELECTED Q-GERT OUTPUT AFTER 100 SIMULATION RUNS</u>

FIGURE 15

working day, where an estimated 18.44 containers are offloaded and 18.77 containers are onloaded. Analysis of the Q-GERT output in Figure 15 indicates that flatcars spend an average of 51.18 hours in the inbound yard (Q-node 13) and an additional 19.55 hours at the siding waiting to be offloaded (Q-node 21). This represents a total of 70.73 hours, of the estimated total 82.33 hours in the system, that the cars are waiting in the system just to be offloaded so that they may be reloaded and shipped out of the system. This delay also directly contributes to the comparatively long time inbound containers spend in the system in relation to the outbound container. Inbound containers are in the parking lot an average of 19.75 hours (Q-node 49). Outbound containers are in the lot an estimated 25.99 hours (Q-node 43). Yet, the total estimated time an inbound container is in the system is 82.33 hours compared to the 42.57 hours for an outbound container.

It would seem apparent that if flatcar inbound processing time could be reduced, then the associated operating cost for flatcars and inbound containers would also be reduced. However, the inbound processing time for flatcars in the system is dependent both on the physical limitations of the terminal and the working rules in effect at the siding. A change in the physical layout of the facility to accommodate additional flatcars at the siding would require capital expenditures. Additionally, any savings in operating cost realized by the reduction in time the cars are in the system could easily be offset by the expected increase in the significant switching cost. The container parking lot would also be affected. A change in siding working rules to allow a two, or three, shift working day, or a six or seven day work week would also reduce the time flatcars are in the system, but again additional switching cost would be incurred, and an additional burden would be placed on the container parking lot.

Currently, the container parking lot is overloaded an estimated five percent of the time for the reported level of activity in the system. An increase in container offloads would require additional space, and for the Roanoke facility, automatically create further requirements for space for returning outbound containers. Therefore, for any increase in siding activity, it could be anticipated that the parking lot overloading condition would deteriorate. This would result in increased cost that would further reduce the benefits accrued by reducing the time flatcars and inbound containers are in the system.

One further observation about the current intermodal operations at the Roanoke facility is that given the system's current siding and parking lot capacities, and the working rules that are in effect for the system, managers can anticipate that the facilities' operating cost will increase at rates beyond those currently being incurred with only moderate increases in traffic activity. This forecast is predicated on the current flatcar service rate at the siding per work day (10.5), and the arrival rate of inbound cars in the system per day (8.6). Should traffic increase to a rate of 10.5 car arrivals per day, or beyond, changes in current switching and working rules will be required to increase the service rate at the siding. This will increase operating cost and could reduce profits if the increase in revenue provided by the additional traffic is not sufficient to offset the additional cost.

A Suggested System Evaluation

Validation of the intermodal terminal simulation model indicates that it does reasonably represent the current operations of the Norfolk and Western facility at Roanoke, Virginia. An analysis of the simulation output for the current operations of the terminal suggest that certain cost reductions might be realized by modifying the working and switching rules at the facility, but that other cost would be expected to increase. These increased costs could well offset any cost reduction at disproportionate rates. Further, the rules will require modification for even moderate increases in traffic beyond the levels currently experienced at the facility. This action will almost certainly increase operating cost. The effect of changes in the working and switching rules can be estimated by incorporating the changes into the model, simulating the system under the proposed conditions, then comparing the output results to those that have been reported above. A comparative analysis of the output should reveal where improvements could, or could not, be achieved. Additional model changes could also be indicated for subsequent simulations. Operating strategies could also be developed for given levels of traffic activity at the facility.

The analysis of the simulation results for the current operations of the Roanoke terminal also reveal that there could be an opportunity to realize substantial savings in operating cost if switching times could be reduced. The opportunity for additional savings might also be available if the capacities of the siding and parking lot were increased. One practical way to reduce switching cost is to reduce the time required to switch a car to and from the siding. Unfortunately, this usually requires moving the siding nearer to the yards. However, if a new facility layout is being considered, then it would be appropriate to estimate the expected switching times per car and incorporate them into a simulation for the proposed facility. Switching cost estimates could then be obtained, and trade-off analysis initiated. Changes in parking lot and siding locations, in addition to variations in their capacities' could also be incorporated in the model. The resulting cost estimates could then be incorporated into the analysis.

The suggested changes in the system that would result in physical modification of the facility would require capital expenditures, however. The estimated savings expected from such investments should be offset against the internal rate of return for the cost of these investments over the expected useful life of the assets generating the savings. The capital investment decision would require a greater in-depth analysis of the overall operation of the entire yard facility than can be provided for in the intermodal terminal model. However, the expected values of the cost estimates generated by a simulation of the incorporation of a proposed physical change in the intermodal operations of the facility could provide the basis for comparison of the relative merits of all alternatives under consideration. Therefore, the model could provide intermodal managers with an analysis tool that might be valuable in presenting their proposals to those in upper management levels responsible for capital investment decisions.

Summary

This chapter details a four phase process suggested for the implementation of the intermodal railway terminal model. In the first phase, emphasis is placed on the observation of the system to be modeled, and the gathering of the data necessary for model implementation. The second phase outlines the procedures required to modify the model presented in this paper to conform to the terminal to be simulated.

Recommended procedures for the validation of the model for a particular terminal are illustrated in the third phase of model implementation. This is accomplished through the use of an example of model validation for an existing terminal. The terminal's operations were simulated for a fifty-nine day period, and the simulation's results reported for the facility for the same operating period.

Phase four includes model applications and some suggested ways in which the model could aid intermodal managers in planning for different levels of traffic flow within the system. Operating cost estimates are also presented in this section and methods for incorporating the model's output into the capital investment decision process are discussed.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The purpose of this research is to provide intermodal managers in the rail industry with a computer simulation model for a railway intermodal terminal system to analyze the sensitivity of a terminal to changes in intermodal operations. The Q-GERT simulation model that has been developed incorporates the daily arrival/departure train schedule, yard switching rules for the terminal, siding operations and working rules, container parking lot activities, and over-the-road traffic patterns. Multiple types of flatcars and containers have also been provided for in the model, including car switching priority selection rules that provide a realistic representation of complex terminal systems. The physical layout of an individual terminal including the type of equipment employed, is provided for through model input specifications. These are represented by switching time, container onloading and offloading times, parking lot processing times, and the maximum capacities of the siding and the parking lot. The capability of the model to provide information about an intermodal terminal is discussed below.

Conclusions

The intermodal terminal simulation model developed through this research has been designed to provide managers with information about their system. This information includes the estimation of the system's flatcar and container over-all capacities, the time required to process a flatcar or container through the system, the average number of units in the system, operating efficiency, and the expected cost of operations. This information should assist managers in the evaluation of the performance of a terminal. In addition, by modifying the input data for the model, a manager can estimate the effects of changes in the system on the performance of the terminal. Therefore, simulation experiments can be conducted to test the effectiveness of modifications to the existing working environment, such as the number of switches permitted per day, weekend working rules, parking lot and siding working hours, or container consignee notification rules. Or, experiments can be accomplished to estimate the impact of physical changes in the system.

A supplemental application of the model could result from the network orientation of the Q-GERT simulation language in which it is written. The flow of flatcars and containers is readily identifiable in the model through the graphical representation of the network by individuals with no simulation experience. Therefore, the model provides a visual reference of a terminal that communicates the technical aspects of operations in a manner that is straight forward. This feature could prove valuable to intermodal managers when communicating with higher levels of management. For example, a graphical illustration of the model could be used to demonstrate to upper management a congestion problem that is encountered at a facility. Alternatives for a solution of the problem, with cost estimates based on simulation results, could then be presented.

The Q-GERT Intermodal Terminal Model can provide managers with an accurate replication of a terminal's operations given that proper attention is given to the specification of the input data and parameter values required for the model. The model also provides a great deal of flexibility. The design of the model is such that many changes can be incorporated for a given terminal. In addition, with minimum model modification, several different terminals can be simulated. Convenience is another feature of the model once the Q-GERT Analysis Program is available on the user's computer. The simulation program representing a terminal can be run as often as required, and the results of each simulation readily compared.

Recommendations

The primary applications of the intermodal terminal simulation model developed during this research are viewed to be in the areas of planning and analysis. A secondary application could be in communications.

As a planning tool, the model is capable of providing the testing vehicle for experiments involving new concepts of facility design or operational criteria. These could include the physical layout of a terminal, the type of equipment employed, the allocation of equipment, switching schedules, container loading and unloading procedures, or parking lot utilization.

The model has the ability to provide information for the analysis of the limitations of an existing terminal. This analysis could provide information on the operating capacity of the system, in terms of both flatcars and containers, over time, and under varying circumstances. Estimates can be obtained for the time units are in the system, operating costs, and equipment utilization.

The graphical network orientation of the model could be used as a communications tool to apprise decision makers of new concepts, or current problems. Recommended solutions could also be illustrated. Future applications in communications could include incorporating selected output of a simulation into existing management information systems. This information might include the expected impact on the system of an impending train arrival, provide more efficient container consignee notification procedures, or include the specific location of individual cars, or containers, within the system.

BIBLIOGRAPHY

BOOKS

1. Chen, Gordon K. C., and Kaczka, Eugene E., Operations and Systems Analysis, A Simulation Approach, Allyn and Bacon, Inc., Boston, Mass., 1974.

2. Evans, George W., II, Wallace, Graham F., and Sutherland, Georgia L., Simulation Using Digital Computers, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1967.

3. Emshoff, James R., and Sisson, Roger L., Design and Use of Computer Simulation Models, The Macmillan Company, New York, NY, 1970.

4. Gordon, Geoffrey, System Simulation, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.

5. Maisel, Herbert, and Gnugnoli, Guiliano, Simulation of Discrete Stochastic Systems, Science Research Associates, Inc., Chicago, IL, 1972.

6. Mandell, Steven L., Computers and Data Processing, Concepts and Applications with Basic, West Publishing Company, New York, 1979.

7. Meier, Robert C., Newell, William T., and Pazer, Harold L., Simulation in Business and Economics, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.

8. Mize, Joe H., and Cox, J. Grady, Essentials of Simulation, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1968.

9. Naylor, Thomas H., Bolintfy, Joseph L., Burdick, Donald S., and Chu, Kong, Computer Simulation Techniques, John Wiley and Sons, Inc., New York, NY, 1968.

10. Pritsker, A. A. B., Modeling and Analysis Using Q-GERT Networks, 2nd Ed., John Wiley and Sons, Inc., New York, NY, 1979.

11. Pritsker, A. A. B., and Pegden, Claud D., Introduction to Simulation and Slam, John Wiley and Sons, New York, NY, 1979.

12. Pritsker, A. A. B., The GASP IV Simulation Language, John Wiley and Sons, Inc., New York, NY, 1974.

13. Pritsker, A. A. B., and Kiviot, Philip J., Simulation With GASP II, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.

14. Shannon, R. E., System Simulation, The Art and Science, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.

REPORTS-PUBLISHED

15. Ainsworth, Donald P., and Isacowitz, Donald A., An Evaluation of Alternative Railroad Terminal Container Handling Systems, Robert Reebie and Associates, Inc., Greenwich, Conn., March, 1971.

16. Best, N. M., Systems Engineering for Intermodal Freight Systems. Phase I. Exploratory Planning. Volume II, Task Results, Report No. FRA/ORD-78/24. II, A. T. Kearney and Company, Inc., Washington, DC, August, 1978.

17. Blaze, J. R., "Rail and Rail-Truck Terminals: An Opportunity in Urban Transportation Planning", Transportation Research Record, N511, Transportation Research Board, Washington, DC, 1974.

18. Bushness, R., Low, L., and Pearsall, E., Integrated Computer Model for Analyzing Modal Shifts Due to Changes in Cost Conditions, Report No. CONF-790455-4, School of Business Administration, Wayne State University, Detroit, MI, April, 1979.

19. Combined Truck/Rail Transportation Service: Action Needed to Enhance Effectiveness, Report No. B-139052, General Accounting Office; Community and Economic Development Division, General Accounting Office Distribution Section, Washington, DC, December, 1977.

20. Danielsson, L., and Wachtmeister, F., European Piggyback-Transport. When? Where? How? Report No. 5118/1979, Lund University of Technology, Lund, Sweden, February, 1979.

21. DeBoer, D. J., "Intermodal Realities," Transportation Research Board, No. 656, Transportation Research Board Publications Office, Washington, DC, 1977.

22. Follett, Earl, and Shrier, Elliot, The Impact of Containerization on the U. S. Economy, Volume I, Matson Research Corporation, San Francisco, Calif., September, 1970.

23. Future Outlook for Combined Transport in the Federal Republic of Germany: Market Analyses for Container and for Piggyback Transport, Ministry of Transport, Bonn, West Germany, 1971.

24. Hammesfahr, R. D., and Hawk, T. R., A Comparative Economic Analysis of a Lighter-Than-Air Logistics System, SLSR 39-73B, School of Systems and Logistics, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, August, 1973.

25. Hill, D., Leilich, R., Elliff, A., Morice, W., and Edsforth, J., Systems Engineering for Intermodal Freight Systems. Phase I Exploratory Planning. Volume V. Task Results: Appendices, Report No. FRA/ORD-78/24.V, Peat, Marwick, Mitchell and Company, Washington, DC, July, 1978.

26. Hill, D., Leilich, R., Elliff, A., Morice, W., and Edsforth, J., Systems Engineering for Intermodal Freight Systems. Phase I Exploratory Planning, Volume IV, Task Results, Report No. FRA/ORD-78/24.IV, Peat, Marwick, Mitchell and Company, Washington, DC, July, 1978.

27. Jones, H. W., "Intermodal Transport and Containerization," Transportation Research Record, No. 656, Transportation Research Board Publications Office, Washington, DC, 1977.

28. Kachadourian, George, TOFC (Trailer on Flatcar) Lading Response Analysis for Several Track Profile and Hunting Conditions, MITRE Corporation, McLean, VA, April, 1980.

29. Law, Averill, M., "Validation of Simulation Models, I: An Overview and Survey of Real-World Practice," Technical Report No. 78-14 for the Office of Naval Research, University of Wisconsin, Madison, Wisc., August, 1979.

30. Law, Averill M., "Validation of Simulation Models, II: Comparison of Real-World and Simulation Output Data," Technical Report No. 78-15 for the Office of Naval Research, Department of Industrial Engineering, University of Wisconsin, Madison, Wisc., Fall, 1979.

31. Leilich, R. H., Economics of Improved TOFC/COFC Systems, Peat, Marwick, Mitchell and Company, Washington, DC, 1979.

32. Mayer, H. M., "Changing Urban Structure and Its Implications for Terminals and Pickup and Delivery Problems in Metropolitan Areas," Highway Research Board Special Reports, No. 120, pp. 110-120, 1971.

33. Moore, L. J., "An Experimental Investigation of a Computerized Check-Processing System in a Large City Bank Using Digital Simulation," Dissertation, Arizona State University, September, 1970.

34. Nyquist, Haver, Kluk, Markham, Lkoss, DeClaire, Hood, Britt, and Lipman, Systems Engineering for Intermodal Freight Systems. Phase I - Exploratory Planning -- Volume III -- Task Results: Appendices, Report No. FRA/ORD-78/24. III, Washington, DC, August, 1978.

35. Peternick, J., Fredrickson, V., Pflugrad, A., Rynders, B., and Dillenback, D., Intermodal Management Information Systems (IMIS). Phase III, Task I, Baseline Specifications, Report No. FRA/ORD-79/21, PRC Systems Sciences Company, McLean, VA, September, 1979.

36. Peternick, J., Fredrickson, V., Pflugrad, A., Rynders, B., and Wiersema, R., State-of-the-Art Survey for Intermodal Management Information System--Phase II. Task I. Report No. FRA/OPPD-78/8, PRC Systems Sciences Company, McLean, VA, March, 1978.

37. Peternick, J., Fredrickson, V., Heisey, M., Pflugrad, A., Rynders, B., and Rudwick, B., Interim Report, Intermodal Management Information System--Phase II. Task II. Report No. FRA/OPPD-78/12, PRC Systems Sciences Company, McLean, VA, April, 1978.

38. Samuelson, R. D., and Roberts, P. O., COFC VS. TOFC: A Comparison of Technologies, Report No. CTS-77-10, Center for Transportation Studies, Massachusetts Institute of Technology, Cambridge, Mass., June, 1977.

39. Systems Engineering for Intermodal Freight System--Phase I, Exploratory Planning Volume I--Executive Summaries, Report No. FRA/ORD-78/24.I, A. T. Kearney and Company, Inc., Washington, DC, April, 1978.

40. Systems Engineering for Intermodal Freight Systems-Phase I, FRA Contract No. DOT-FE-749-4273, A. T. Kearney and Company, Inc., and Peat, Marwick, Mitchell and Company, Washington, DC, Start date August 1977. Termination date 1980. No final report.

41. Tripp, Robert S., A Simulation Analysis of the Economic Consequences of Establishing Multi-Modal Transportation Companies, Technical Report No. AFIT-TR-72-6, Air Force Institute of Technology, School of Engineering, Wright-Patterson AFB, Ohio, December, 1972.

PROCEEDINGS

42. Assarabowske, R. J., and Sussman, J. M., "Simulation of Dedicated Fleet Operations in the Railroad Industry," Proceedings of the 1976 Winter Simulation Conference, pp. 257-264.

43. Bagwell, B. M., and Brown, K. S., "An Application of Goal Programming To a Railroad Terminal Scheduling Problem," Proceedings of the 6th Annual South East AIDS Conference, Atlanta, 1976, pp. 186-187.

44. Hammesfahr, R. D., and Clayton, E. R., "The Simulation of Inventory Problems with the New Q-GERT Analysis Program," Proceedings of the 9th Annual South East AIDS Conference, Nashville, Tenn., February, 1979.

45. Hammesfahr, R. D., Rakes, T. R., and Clayton, E. R., "An Application of Q-GERT to the System Reliability Problem," Proceedings 14th Annual South East TIMS Conference, Atlanta, GA, October, 1978.

46. Holland, R. E., and Clayton, E. R., "Toledo Intermodal Facility Operational Considerations," Proceedings 6th Annual South East AIDS Conference, Atlanta, GA, 1976.

47. Minger, W. K., and Cetinich, J. N., "Association of American Railroad's Network Model," Proceedings of the 1969 Winter Simulation Conference, December 8-9, 1969, pp. 193-196.

48. Rakes, T. R., and Clayton, E. R., "Applications of GASP IV to a Railroad Ramp Feasibility Study," Eleventh Annual Simulation Symposium, March, 1978.

49. Shen, N. T., Kang, A. N. C., and Kang, C. K., "Computer Simulation of a Small Railroad System for Optimal Passenger Service with Minimal Capital Investment," Proceedings of the Eighth Annual Simulation Symposium, March, 1975, pp. 117-179.

ARTICLES IN JOURNALS AND MAGAZINES

50. Balderstone, R. J., "Straddle v. Sidelift -- Room for All in Terminal Operations," Cargo Systems International, Vol. 5, No. 12, Ship Research Institute of Norway, Technical University of Norway, Trondheim, Norway, 1978.

51. Beier, F. J., and Frick, S. W., "The Limits of Piggyback: Light at the End of the Tunnel," Transportation Journal, Winter, 1978, pp. 12-18.

52. "Can Piggyback Widen Its Markets?," *Railway Age*, April 24, 1978, pp. 27-28.

53. "Fuel Efficiency: TOFC vs. Trucks," *Railway Age*, January 28, 1979, pp. 14-16.

54. Grygiel, J. A., "Problems at the Interface. A Rail Viewpoint," *ICHCA Journal*, Engineering Societies Library, New York, NY, September, 1973.

55. "GTW Consistently Wrings Profits from a 'Too Short' Corridor," *Railway Age*, April 28, 1980, pp. 34-35.

56. Hoefer, R., "The Most Modern Container Storage Installations and New Railway Installations in the Hamburg Port Railway Station Area," *Verkehr und Technik*, Vol. 29, No. 6, Bielefeld, West Germany, June, 1976.

57. "How FEC Handles Its Growing Piggyback Volume," *Railway Age*, March 26, 1979, pp. 54.

58. "Improving Intermodal Terminals: A Prime Target of FRA Study," *Railway Age*, October 30, 1978, pp. 34.

59. Kobayashi, M., and Ohnuma, T., "Plan of New Container Handling at Intermediate Station," *Railway Technical Research Institute*, Vol. 15, No. 4, Tokyo, Japan, December, 1974.

60. Kuhns, R. E., and Mulinazzi, T. E., "Locational Decision Methodology for Intermodal Rail-Truck Terminals," Paper No. P&P-35, *American Society of Mechanical Engineers*, New York, NY, 1976.

61. Lallemand, A., "Road Vehiclles Transported by Rail," *Container Actualities*, No. 76, Paris, France, December, 1977.

62. Lehmann, E., "Development of a New Wagon for the Horizontal Transfer of Lorries and Semi-Trailers," *Internationales Verkehrswesen*, Vol. 31, No. 2, Darmstadt, West Germany, March, 1979.

63. Malone, F., "Intermodal - How CN Moves Ahead by Reining In," *Railway Age*, April 28, 1980, pp. 32-33.

64. Malone F., "New Peaks for Piggybacks," *Railway Age*, October 30, 1978, pp. 22-25.

65. "Market-Piggyback Loadings," *Railway Age*, June 9, 1980, pp. 51.

66. McKenney, James L., "Critique of: 'Verification of Computer Simulation Models'," *Management Science*, Vol. 14, No. 2, October, 1967.

67. Montgomery, A. J., "The Role of Intermodal Service in Improving Railroad Productivity," Transportation Journal, Summer, 1977, pp. 46-53.

68. Morash, E. A., Hille, S. J., and Bruning, E. R., "Marketing Rail Piggyback Services," Transportation Journal, Winter, 1978, pp. 40-50.

69. Naylor, Thomas H., and Finger, J. M., "Verification of Computer Simulation Models," Management Science, Vol. 14, No. 2, October, 1967.

70. Nellis, J. F., and Weart, W. W., "Will Intermodal 'Make it Big'?," Railway Age, April 24, 1978, pp. 30-31.

71. Newbourne, M. J., "Piggyback Needs the 45-Foot Trailer," Traffic World, Vol. 177, No. 8, Traffic Services Corporation, February, 1979.

72. "100 Pigs in a Row: Santa Fe's New Super Piggyback," Forbes, November 12, 1979, pp. 61.

73. "Piggyback in Western Europe," Railway Gazette International, Vol. 129, No. 9, IPC Transportation Press, Ltd., London, England, September, 1971.

74. Roberts, R., "The Future is Intermodal," Modern Railroads, November, 1979.

75. "Sante Fe Will Test 'Ten-Pack' Trains for Fuel Savings," Container News, February, 1978, pp. 49.

76. Schrank, William E., and Holt, Charles C., "Critique of: 'Verification of Computer Simulation Models'," Management Science, Vol. 14, No. 2, October, 1967.

77. "Short and Significant," Railway Age, September 11, 1978, pp. 4.

78. Short, R. B., "N & W's Anchor Program Works: Will 'Common Box' Be Next?," Container News, February 1979, pp. 49.

79. Short, R. B., "Intermodal Dinosaurs," Intermodal, November, 1977.

80. Short, R. B., "Railroads Need an Efficient Container Utilization System," American Import/Export Bulletin, July, 1977, pp. 449-450.

81. Stieler, G., "The Piggy-Back System 'Unit Pocket Wagons and Verticle Transshipment'," Glasers Annalen ZEV, No. 12, International Union of Railways, Paris, France, 1972.

82. Stone, N. M., "Terminal Delays are Still a Major Problem," Container News, Vol. 8, No. 5, Communications Channels, Inc., New York, NY, May, 1973.

83. Strobel, M., "Combined Transport Trends," Die Bundesbahn, Vol. 55, No. 7, Darmstadt, West Germany, July, 1979.

84. Taylor, Bernard W., III, Moore, Laurence J., and Hammesfahr, R. D., "Global Analysis of a Multi-Product, Multi-Line Production System Using Q-GERT Modeling and Simulation," AIIE Transactions, Vol. 12, No. 2, June, 1980.

85. Van der Sluys, W., Spence, J. H., and Marshall, M. G., "Performance of TOFC-COFC Arrangements in Yard-Type Impact Tests," paper presented to the American Society of Mechanical Engineers, New York, NY, 1971.

86. VanHorn, Richard L., "Validation of Simulation Results," Management Science, Vol. 17, No. 5, January, 1971.

87. Welty, G., "Piggyback's Potential for Perishables," Railway Age, January 8, 1979, pp. 30-32.

88. Welty, G., "Rolling Toward New Record-Highs," Railway Age, May 28, 1979, pp. 26-30.

89. Welty, G., "Technology is Booming," Railway Age, April 24, 1978, pp. 22-24.

90. Welty, G., "TOFC/COFC Terminals: Key to Service and Growth," Railway Age, May, 1969.

91. Whitehurst, Clinton H., Jr., "The Multimodal Transportation Company: An Idea Whose Time Has Come?" The Clemson University Review of Industrial Management and Textile Science, Vol. 19, No. 1, Spring, 1980.

92. Wright, J. F., "New Roles for Intermodalism," Progressive Railroading, Vol. 22, No. 11, Chicago, IL, November, 1979.

UNPUBLISHED MATERIALS

93. Cole, P., et al, "Strategic Intermodal Marketing Campaign for Norfolk and Western Railway," An Unpublished Report for Dr. S. Cosmas, VPI&SU, February, 1980.

94. Cracker, W. F., and Wong, P. J., "New Technology in the Design, Measurement and Control of Railroad Yards," Paper Presented to the ASME Conference, December 12, 1979.

95. Williamson, W. V., and Wong, P. J., "SP/SRI Dynamic Track Assignment Program: An Overview," the 1978 AAR Data Systems Division Annual Meeting, October 9-11, 1978.

96. Wong, P. J., Conrad, B., and Johnson, J. M., "Tactical Planning for Coordinating Railroad Operations," Paper Presented at the Fifty-Ninth Annual Meeting of the Transportation Research Board, January 21-25, 1980.

97. Wong, P. J., "Yard Design Methods: State of the Art," Paper Presented at a Classification Yard Technology Workshop sponsored by the Office of Research and Development, FRA, October 30, 1971.

PERSONAL INTERVIEWS AND TELEPHONE CONVERSATIONS

98. Blanchfield, J., Systems Engineer, Federal Railroad Authority, Washington, DC, March 31, 1980.

99. Brooks, W., Intermodal Operations, Federal Railroad Authority, Washington, DC, March 31, 1980.

100. Holland, R. E., Manager, Intermodal Planning and Control, Norfolk and Western Railway Company, Roanoke, VA, December 17, 1979, February 1, 1980, and March 14, 1980.

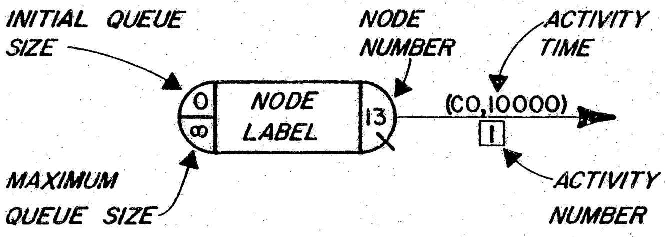
101. Short, R. B., Director, Intermodal Sales and Service, Norfolk and Western Railway, Roanoke, VA, December 17, 1979.

APPENDIX A

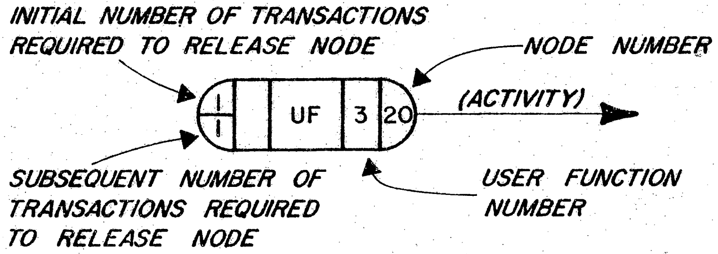
Q-Gert Notation and Symbols for the Intermodal Rail Terminal Model

Q-GERT NOTATION AND SYMBOLS FOR THE INTERMODAL RAIL TERMINAL MODEL

QUEUE NODE



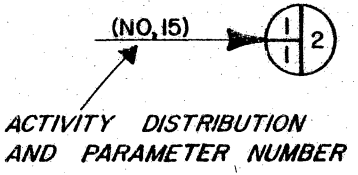
NETWORK NODE WITH USER FUNCTION



STATISTICS / SINK NODE

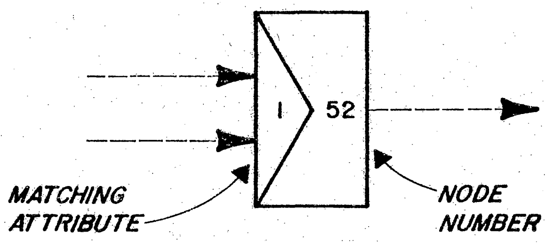
STATISTICS NODES PROVIDE PROGRAM MAINTAINED STATISTICS

SINK NODES CAN BE USED TO SPECIFY PROGRAM STOPPING CONDITIONS



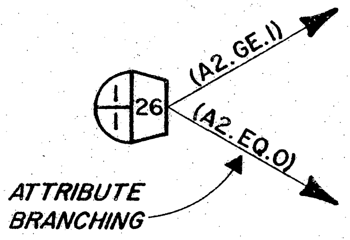
MATCHING NODE

MATCHES TRANSACTIONS RESIDING IN SPECIFIED Q-NODES THAT HAVE EQUAL VALUES FOR A SPECIFIED ATTRIBUTE



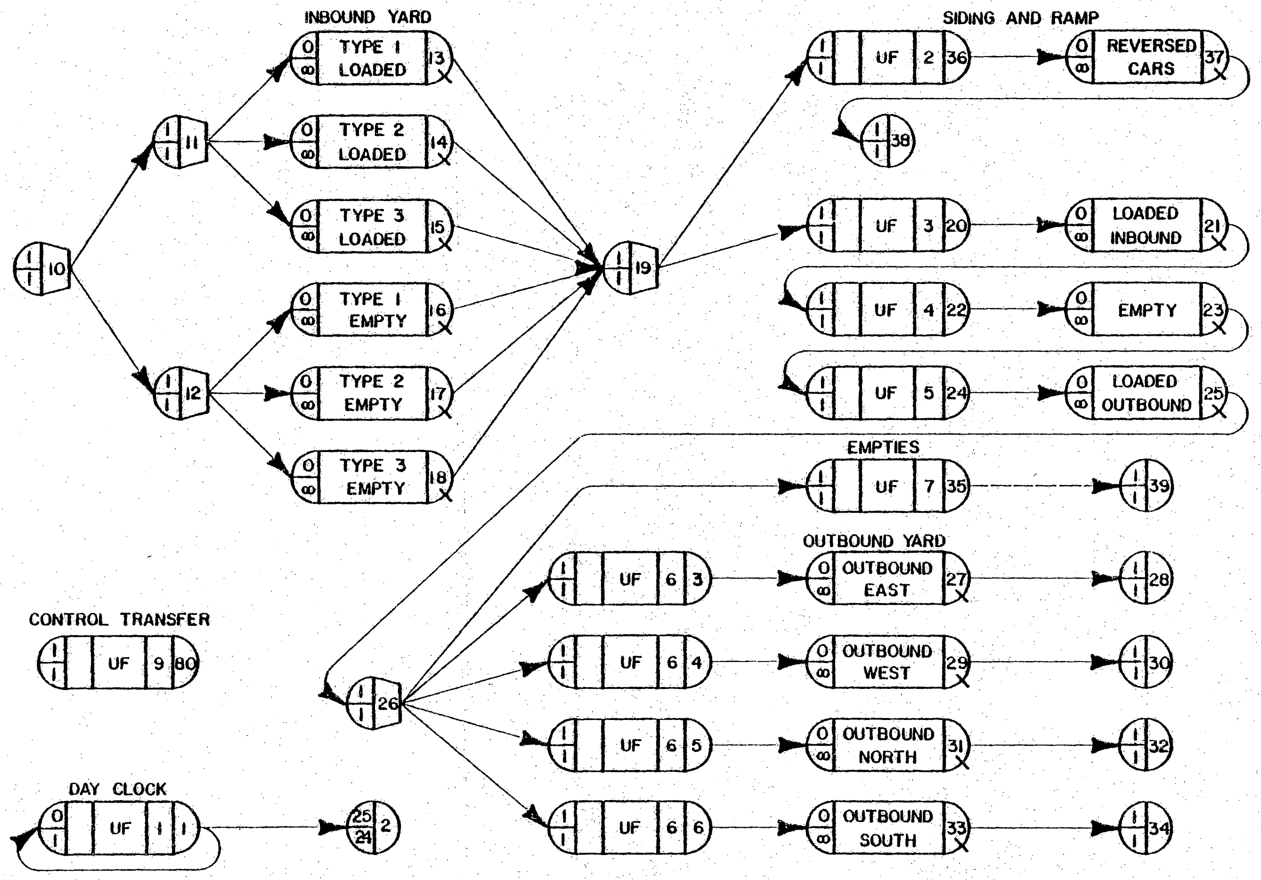
CONDITIONAL BRANCHING NODE

TRANSACTIONS ARE ROUTED TO FOLLOWING NODES BASED ON SPECIFIED ATTRIBUTE VALUES FOR ACTIVITIES

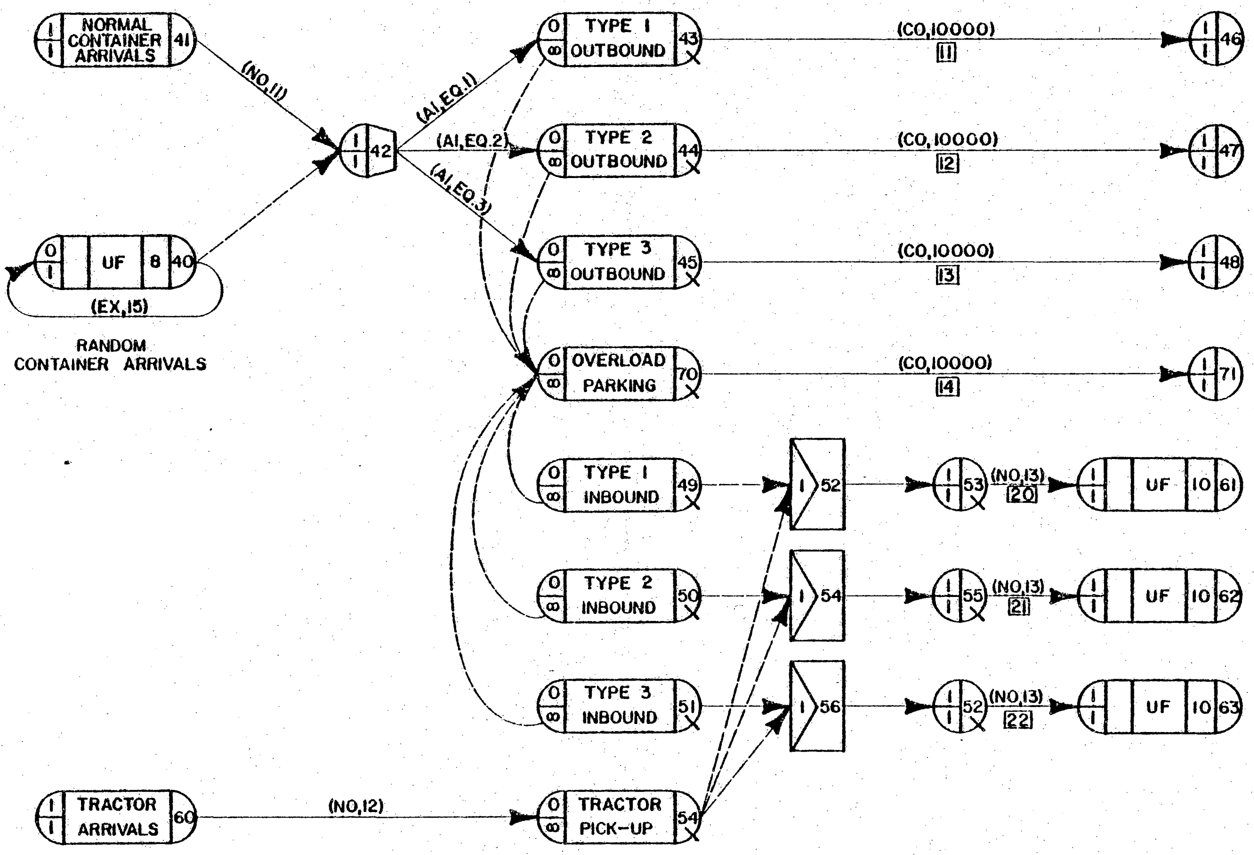


APPENDIX B

The Q-GERT Model for the Railway Intermodal Terminal



Q-GERT INTERMODAL TERMINAL RAIL OPERATIONS MODEL



Q-GERT INTERMODAL TERMINAL PARKING LOT OPERATIONS MODEL

APPENDIX C

Computer Program Listing for Subroutine UI for the Intermodal Rail Terminal Model

```
      SUBROUTINE UI
      COMMON/QVAR/NDE,NFTBU(100),NREL(100),NRELP(100),NREL2(100).
     1NRUN,NRUNS,NTC(100).PARAM(100,4),TBEG.TNOW
      COMMON/UCOM1/IHOUR,IDAY,ATIME(90,10).ITP1(90,10),ITP2(90,10),
     1ITP3(90,10),DTIME(90,10),IDEPT(90,10).ITCARS,ITONE,ITTWO,
     2ITEMT.ITONER,ITEMTR,MXRAMP,MXPARK,ISTART.ITYP1E,ITYP1W,ITYP1N.
     3ITTWOR,ITYP1S,ITYP2E.ITYP2W,ITYP2N,ITYP2S.ITYP3E,ITYP3W,ITYP3N,
     4ITYP3S,IEAST,IWEST.INORTH,ISOUTH,IPRI1,IPRI2.INIT.IPARK1,IPARK2,
     5IPARK3,IPARKT(90),IOVP(90),MOVE(90)
      COMMON/UCOM2/TSWIN(90.10),ISWINT(90.10),TOFF(90,10),TON(90.10)
     1,LCON(90,10).NSWOUT(90,10),TSOUT(90,10),LCOFF(90,10).MCAP,
     2NSWIN,NOFF,NON,NOUT.ISAT.ISUN,IWORK.NGND,INPRO
      COMMON/USTAT/UOBV(25,5),XYZZZ(906)


C
C     READ IN PRIORITIES AND INITIALIZE SYSTEM.
C
   40 IPRI1=1
      IPRI2=0
      INIT=0
      INPRO=0
      NGND=1
      IDAY=1
      IHOUR=0
      MXRAMP=11
      MXPARK=75
      MCAP=MXPARK
      NSWIN=0
      NOFF=0
      NON=0
      NOUT=0
      ITCARS=0
      ITONE=0
      ITTWO=0
      ITEMT=0
      ITONER=0
      ITTWOR=0
      ITEMTR=0
C
C     SET WORKING RULES FOR WEEKENDS.
C
C     ISAT=1, WORK ON SATURDAYS.
C     ISAT=0, NO WORK ON SATURDAYS.
C     ISUN=1, WORK ON SUNDAY
C     ISUN=0, NO WORK ON SUNDAY.
C
      ISAT=0
      ISUN=0
      IWORK=1
C
C     INITIALIZE PARKING LOT
C
C     INBOUND LOT
C
```

```
      ITYP1W=19
      ITYP1N=2
      ITYP1S=0
      ITYP2W=0
      ITYP2E=0
      ITYP2N=0
      ITYP2S=0
      ITYP3W=0
      ITYP3E=0
      ITYP3N=0
      ITYP3S=0
C
C     INITIALIZE OUTBOUND YARD
C
      IEAST=0
      IWEST=8
      INORTH=2
      ISOUTH=0
C
C      INITIALIZE INBOUND YARD
C
      DO 50 I=1,90
      DO 45 J=1,10
      ATIME(I,J)=25.0
      ITP1(I,J)=0
      ITP2(I,J)=0
      ITP3(I,J)=0
      DTIME(I,J)=25.0
      IDEPT(I,J)=0
   45 CONTINUE
   50 CONTINUE
      ATIME(1,1)=0.0
      ITP1(1,1)=18080208
C
C     READ IN SCHEDULE.
C
C
C
C     TRAIN ARRIVALS
C
      ATIME(1,2)=10.0
      ITP1(1,2)=03010100
      ATIME(1,3)=17.0
      ITP1(1,3)=01010000
      ATIME(2,1)=8.0
      ITP1(2,1)=05030200
      ATIME(3,1)=0.0
      ITP1(3,1)=04030100
      ATIME(3,2)=13.0
      ITP1(3,2)=01010000
      ATIME(3,3)=20.0
      ITP1(3,3)=03020000
      ATIME(4,1)=3.0
      ITP1(4,1)=05050000
      ATIME(4,2)=6.0
      ITP1(4,2)=03030000
      ATIME(4,3)=23.0
      ITP1(4,3)=01000100
      ATIME(5,1)=3.0
      ITP1(5,1)=01000100
      ATIME(5,2)=5.0
      ITP1(5,2)=03000300
```

```
ATIME(5,3)=8.0
ITP1(5,3)=07070000
ATIME(5,4)=15.0
ITP1(5,4)=01010000
ATIME(5,5)=22.0
ITP1(5,5)=07070000
ATIME(6,1)=1.0
ITP1(6,1)=01010000
ATIME(6,2)=6.0
ITP1(6,2)=01010000
ATIME(6,3)=19.0
ITP1(6,3)=03030000
ATIME(7,1)=9.0
ITP1(7,1)=01010000
ATIME(7,2)=14.0
ITP1(7,2)=07070000
ATIME(7,3)=21.0
ITP1(7,3)=02020000
ATIME(8,1)=7.0
ITP1(8,1)=02020000
ATIME(8,2)=18.0
ITP1(8,2)=06060000
ATIME(9,1)=0.0
ITP1(9,1)=03010200
ATIME(9,2)=15.0
ITP1(9,2)=02020000
ATIME(10,1)=8.0
ITP1(10,1)=04000103
ATIME(10,2)=15.0
ITP1(10,2)=04040000
ATIME(10,3)=17.0
ITP1(10,3)=04040000
ATIME(11,1)=4.0
ITP1(11,1)=03010200
ATIME(11,2)=6.0
ITP1(11,2)=02020000
ATIME(11,3)=18.0
ITP1(11,3)=02020000
ATIME(12,1)=0.0
ITP1(12,1)=01010000
ATIME(12,2)=8.0
ITP1(12,2)=02020000
ATIME(12,3)=14.0
ITP1(12,3)=03030000
ATIME(12,4)=16.0
ITP1(12,4)=09080100
ATIME(12,5)=23.0
ITP1(12,5)=01010000
ATIME(13,1)=4.0
ITP1(13,1)=01010000
ATIME(13,2)=9.0
ITP1(13,2)=02010100
ATIME(13,3)=11.0
ITP1(13,3)=01010000
ATIME(13,4)=19.0
ITP1(13,4)=03030000
ATIME(14,1)=4.0
ITP1(14,1)=03030000
ATIME(14,2)=7.0
ITP1(14,2)=02010100
ATIME(14,3)=18.0
```

```
ITP1(14,3)=08080000
ATIME(14,4)=23.0
ITP1(14,4)=02010100
ATIME(15,1)=6.0
ITP1(15,1)=03030100
ATIME(15,2)=7.0
ITP1(15,2)=01000100
ATIME(16,1)=7.0
ITP1(16,1)=05030101
ATIME(16,2)=15.0
ITP1(16,2)=01010000
ATIME(17,1)=3.0
ITP1(17,1)=01010000
ATIME(17,2)=4.0
ITP1(17,2)=01010000
ATIME(17,3)=6.0
ITP1(17,3)=02000200
ATIME(17,4)=15.0
ITP1(17,4)=01010000
ATIME(18,1)=4.0
ITP1(18,1)=01000100
ATIME(18,2)=5.0
ITP1(18,2)=01000001
ATIME(18,3)=6.0
ITP1(18,3)=01000100
ATIME(19,1)=3.0
ITP1(19,1)=04040000
ATIME(19,2)=8.0
ITP1(19,2)=07050200
ATIME(19,3)=11.0
ITP1(19,3)=02010100
ATIME(20,1)=0.0
ITP1(20,1)=01000100
ATIME(20,2)=3.0
ITP1(20,2)=03030000
ATIME(20,3)=6.0
ITP1(20,3)=06040200
ATIME(20,4)=10.0
ITP1(20,4)=03030000
ATIME(20,5)=19.0
ITP1(20,5)=07000601
ATIME(21,1)=5.0
ITP1(21,1)=04040000
ATIME(21,2)=17.0
ITP1(21,2)=05010004
ATIME(21,3)=22.0
ITP1(21,3)=03020100
ATIME(22,1)=7.0
ITP1(22,1)=02020000
ATIME(23,1)=10.0
ITP1(23,1)=06050100
ATIME(23,2)=16.0
ITP1(23,2)=03000003
ATIME(24,1)=4.0
ITP1(24,1)=10100000
ATIME(24,2)=5.0
ITP1(24,2)=01000100
ATIME(24,3)=7.0
ITP1(24,3)=01010000
ATIME(24,4)=18.0
ITP1(24,4)=01010000
```

```
ATIME(24,5)=19.0
ITP1(24,5)=01000100
ATIME(25,1)=4.0
ITP1(25,1)=02020000
ATIME(25,2)=7.0
ITP1(25,2)=04010300
ATIME(26,1)=6.0
ITP1(26,1)=01010000
ATIME(26,2)=18.0
ITP1(26,2)=09090000
ATIME(27,1)=6.0
ITP1(27,1)=03020100
ATIME(27,2)=18.0
ITP1(27,2)=01000100
ATIME(27,3)=21.0
ITP1(27.3)=01000100
ATIME(28,1)=7 .0
ITP1(28,1)=02020000
ATIME(28,2)=11.0
ITP1(28,2)=01000001
ATIME(28,3)=19.0
ITP1(28,3)=08070100
ATIME(29,1)=8.0
ITP1(29,1)=01000100
ATIME(29,2)=9.0
ITP1(29,2)=01010000
ATIME(29,3)=19.0
ITP1(29,3)=01010000
ATIME(29,4)=22.0
ITP1(29,4)=01000100
ATIME(30,1)=2.0
ITP1(30,1)=01010000
ATIME(30,2)=7.0
ITP1(30,2)=02010100
ATIME(31,1)=5.0
ITP1(31,1)=01000100
ATIME(31,2)=10.0
ITP1(31,2)=01010000
ATIME(31,3)=18.0
ITP1(31,3)=06060000
ATIME(32,1)=6.0
ITP1(32,1)=01000100
ATIME(32,2)=19.0
ITP1(32,2)=11000407
ATIME(32,3)=22.0
ITP1(32,3)=06060000
ATIME(33,1)=3.0
ITP1(33,1)=02020000
ATIME(33,2)=7.0
ITP1(33,2)=02020000
ATIME(33,3)=18.0
ITP1(33,3)=01000100
ATIME(33,4)=19.0
ITP1(33,4)=03030000
ATIME(33,5)=22.0
ITP1(33,5)=02010100
ATIME(34,1)=6.0
ITP1(34,1)=02010100
ATIME(34,2)=11.0
ITP1(34,2)=01010000
ATIME(34,3)=12.0
```

```
ITP1(34,3)=02010100
ATIME(35,1)=18.0
ITP1(35,1)=01010000
ATIME(35,2)=21.0
ITP1(35,2)=01010000
ATIME(36,1)=11.0
ITP1(36,1)=02010100
ATIME(36,2)=16.0
ITP1(36,2)=01010000
ATIME(36,3)=19.0
ITP1(36,3)=05030200
ATIME(37,1)=2.0
ITP1(37,1)=03030000
ATIME(37,2)=4.0
ITP1(37,2)=01000001
ATIME(37,3)=7.0
ITP1(37,3)=01000100
ATIME(38,1)=2.0
ITP1(38,1)=04040000
ATIME(38,2)=16.0
ITP1(38,2)=02020000
ATIME(39,1)=2.0
ITP1(39,1)=02000002
ATIME(39,2)=3.0
ITP1(39,2)=01000100
ATIME(39,3)=4.0
ITP1(39,3)=06060000
ATIME(39,4)=8.0
ITP1(39,4)=01000001
ATIME(39,5)=18.0
ITP1(39,5)=03000003
ATIME(40,1)=6.0
ITP1(40,1)=02020000
ATIME(40,2)=8.0
ITP1(40,2)=01000001
ATIME(40,3)=16.0
ITP1(40,3)=03030000
ATIME(40,4)=23.0
ITP1(40,4)=01010000
ATIME(42,1)=1.0
ITP1(42,1)=01010000
ATIME(42,2)=16.0
ITP1(42,2)=11110000
ATIME(43,1)=6.0
ITP1(43,1)=01000100
ATIME(43,2)=8.0
ITP1(43,2)=01000100
ATIME(43,3)=12.0
ITP1(43,3)=01000100
ATIME(43,4)=14.0
ITP1(43,4)=01010000
ATIME(44,1)=3.0
ITP1(44,1)=05050000
ATIME(44,2)=6.0
ITP1(44,2)=05040100
ATIME(44,3)=7.0
ITP1(44,3)=02020000
ATIME(44,4)=14.0
ITP1(44,4)=01000100
ATIME(45,1)=2.0
ITP1(45,1)=03030000
```

```
ATIME(45,2)=8.0
ITP1(45,2)=08080000
ATIME(45,3)=17.0
ITP1(45,3)=01010000
ATIME(46,1)=5.0
ITP1(46,1)=05050000
ATIME(46,2)=6.0
ITP1(46,2)=03030000
ATIME(46,3)=15.0
ITP1(46,3)=01010000
ATIME(46,4)=17.0
ITP1(46,4)=04040000
ATIME(47,1)=0.0
ITP1(47,1)=01010000
ATIME(47,2)=7.0
ITP1(47,2)=01010000
ATIME(47,3)=15.0
ITP1(47,3)=03030000
ATIME(48,1)=1.0
ITP1(48,1)=02020000
ATIME(48,2)=2.0
ITP1(48,2)=02020000
ATIME(48,3)=4.0
ITP1(48,3)=02020000
ATIME(48,4)=6.0
ITP1(48,4)=05050000
ATIME(48,5)=20.0
ITP1(48,5)=01010000
ATIME(49,1)=15.0
ITP1(49,1)=06060000
ATIME(50,1)=8.0
ITP1(50,1)=01010000
ATIME(50,2)=12.0
ITP1(50,2)=03030000
ATIME(51,1)=6.0
ITP1(51,1)=03030000
ATIME(51,2)=7.0
ITP1(51,2)=01010000
ATIME(51,3)=11.0
ITP1(51,3)=01010000
ATIME(53,1)=2.0
ITP1(53,1)=01010000
ATIME(53,2)=4.0
ITP1(53,2)=01010000
ATIME(53,3)=11.0
ITP1(53,3)=02020000
ATIME(53,4)=13.0
ITP1(53,4)=01010000
ATIME(53,5)=21.0
ITP1(53,5)=01010000
ATIME(54,1)=8.0
ITP1(54,1)=05050000
ATIME(54,2)=17.0
ITP1(54,2)=04040000
ATIME(55,1)=3.0
ITP1(55,1)=01010000
ATIME(55,2)=8.0
ITP1(55,2)=06060000
ATIME(55,3)=16.0
ITP1(55,3)=02020000
ATIME(56,1)=7.0
```

```
      ITP1(56,1)=01010000
      ATIME(56,2)=10.0
      ITP1(56,2)=04040000
      ATIME(56,3)=13.0
      ITP1(56,3)=04040000
      ATIME(56,4)=15.0
      ITP1(56,4)=05050000
      ATIME(57,1)=9.0
      ITP1(57,1)=03030000
      ATIME(57,2)=13.0
      ITP1(57,2)=01010000
      ATIME(57,3)=18.0
      ITP1(57,3)=02020000
      ATIME(58,1)=5.0
      ITP1(58,1)=01010000
      ATIME(58,2)=7.0
      ITP1(58,2)=05050000
      ATIME(58,3)=14.0
      ITP1(58,3)=03030000
      ATIME(58,4)=20.0
      ITP1(58,4)=03030000
      ATIME(58,5)=23.0
      ITP1(58,5)=01000100
      ATIME(59,1)=4.0
      ITP1(59,1)=02020000
      ATIME(59,2)=7.0
      ITP1(59,2)=06050100
      ATIME(59,3)=16.0
      ITP1(59,3)=01010000
      ATIME(59,4)=19.0
      ITP1(59,4)=01000001
C
C     TRAIN DEPARTURES
C
      DTIME(1,1)=9.0
      IDEPT(1,1)=2025
      DTIME(1,2)=10.0
      IDEPT(1,2)=3025
      DTIME(1,3)=10.0
      IDEPT(1,3)=2025
      DTIME(1,4)=15.0
      IDEPT(1,4)=3025
      DTIME(1,5)=16.0
      IDEPT(1,5)=3025
      DTIME(1,6)=22.0
      IDEPT(1,6)=2025
      DTIME(1,7)=19.0
      IDEPT(1,7)=2025
      DTIME(1,8)=20.0
      IDEPT(1,8)=2025
      DTIME(1,9)=5.0
      IDEPT(1,9)=1025
      DO 52 I=2,63
      DO 51 K=1,9
      DTIME(I,K)=DTIME(1,K)
      IDEPT(I,K)=IDEPT(1,K)
   51 CONTINUE
   52 CONTINUE
      DO 54 I=7,63,7
      DO 53 K=1,6
      DTIME(I,K)=25.0
      IDEPT(I,K)=0
   53 CONTINUE
   54 CONTINUE
      RETURN
      END
```

APPENDIX D

Computer Program Listing for Function UF(IFN) for the Intermodal Rail Terminal Model

```
      FUNCTION UF(IFN)
      COMMON/QVAR/NDE,NFTBU(100),NREL(100),NRELP(100),NREL2(100),
     1NRUN,NRUNS,NTC(100),PARAM(100,4),TBEG,TNOW
      COMMON/UCOM1/IHOUR,IDAY,ATIME(90.10),ITP1(90.10),ITP2(90,10),
     1ITP3(90,10),DTIME(90,10),IDEPT(90.10),ITCARS,ITONE.ITTWO,
     2ITEMT.ITONER.ITEMTR.MXRAMP,MXPARK.ISTART,ITYP1E,ITYP1W,ITYP1N,
     3ITTWOR,ITYP1S,ITYP2E.ITYP2W.ITYP2N.ITYP2S,ITYP3E,ITYP3W,ITYP3N.
     4ITYP3S.IEAST.IWEST.INORTH,ISOUTH,IPRI1.IPRI2.INIT,IPARK1,IPARK2,
     5IPARK3,IPARKT(90),IOVP(90),MOVE(90)
      COMMON/UCOM2/TSWIN(90,10),ISWINT(90,10),TOFF(90,10),TON(90,10)
     1,LCON(90,10).NSWOUT(90,10),TSOUT(90,10),LCOFF(90,10),MCAP,
     2NSWIN,NOFF.NON,NOUT,ISAT,ISUN,IWORK.NGND,INPRO
      COMMON/USTAT/UOBV(25,5),XYZZZ(906)
      DIMENSION TYPE(3).ICARS(3),ICAR(3),ITWO(3),IONE(3),IEMT(3),
     1ITWOR(3).IONER(3),IEMTR(3).IS(50),IP(50).ISL(50),T(50).NUM(50)
      REAL EX
      REAL BE
      REAL ATT(7)
      REAL UN
      REAL NO
      UF=0.0
      IF(INPRO.EQ.0) GO TO 102
      GO TO(60,290,300,400,500,600,700,800,900,950),IFN
   60 IHOUR=IHOUR+1
      IF(IHOUR.LT.24) GO TO 140
      ISS=ISTUS(70,14)
      IF(ISS.EQ.0) GO TO 71
      IOVP(IDAY)=XNINQ(70)
      IOVP(IDAY)=IOVP(IDAY)+1
      IF(MXPARK.LE.0) GO TO 72
      IF(IOVP(IDAY)-MXPARK) 61,62,62
   61 MOVE(IDAY)=IOVP(IDAY)
      GO TO 63
   62 MOVE(IDAY)=MXPARK
   63 IOVP(IDAY)=IOVP(IDAY)-MOVE(IDAY)
      JMOVEV=MOVE(IDAY)
      DO 70 I=1.JMOVEV
      ICF=63
      CALL STAGO(14,80,0.0,0,ATT)
      RETURN
   64 MXPARK=MXPARK-1
      IF(ATT(2).EQ.200.0) GO TO 65
      ATT(2)=300.0
      CALL PTIN(42,0.0,TNOW,ATT)
      GO TO 66
   65 JTYPE=ATT(1)+48.0
      CALL PTIN(JTYPE,0.0,TNOW,ATT)
   66 ISS=ISTUS(70,14)
      IF(ISS.EQ.0) GO TO 95
   70 CONTINUE
      GO TO 95
   71 IOVP(IDAY)=0
   72 MOVE(IDAY)=0
   95 IHOUR=0
      IPARKT(IDAY)=MCAP-MXPARK
      NSWIN=NSWIN+1
      NOFF=NOFF+1
      NON=NON+1
```

```
      NOFF=NOFF+1
      NON=NON+1
      NOUT=NOUT+1
      TSWIN(IDAY,NSWIN)=0.0
      ISWINT(IDAY,NSWIN)=0
      TOFF(IDAY,NOFF)=0.0
      LCOFF(IDAY,NOFF)=0
      TON(IDAY,NON)=0.0
      LCON(IDAY,NON)=0
      TSOUT(IDAY,NOUT)=0.0
      NSWOUT(IDAY,NOUT)=0
      IDAY=IDAY+1
      NON=0
      NOUT=0
      NOFF=0
      NSWIN=0
      IWDAY=IDAY+5
      DO 96 I=6,IWDAY,7
      IF(I.EQ.IDAY) GO TO 98
   96 CONTINUE
      DO 97 I=7,IWDAY,7
      IF(I.EQ.IDAY) GO TO 99
   97 CONTINUE
      GO TO 101
   98 IF(ISAT.EQ.1) GO TO 101
      GO TO 100
   99 IF(ISUN.EQ.1) GO TO 101
  100 IWORK=0
      GO TO 102
  101 IWORK=1
C
C     ARE THERE ANY TRAINS SCHEDULED TO ARRIVE THIS HOUR?
C
  102 N=10
      INPRO=1
      DO 117 M=1,N
      TRANA=ATIME(IDAY,M)
      IF(TRANA.LT.25.0) GO TO 104
      GO TO 117
  104 ACTUAL=NO(1)
      ARR=TRANA+ACTUAL
      IF(ARR.LT.0.0) ARR=0.0
      ICAR(1)=ITP1(IDAY,M)
      ICAR(2)=ITP2(IDAY,M)
      ICAR(3)=ITP3(IDAY,M)
      K=3
  105 DO 116 I=1,K
      IF(ICAR(I).EQ.0) GO TO 116
      TYPE(I)=I
      ICARS(I)=ICAR(I)/1000000
      IMULT=ICARS(I)*1000000
      IPASS=ICAR(I)-IMULT
      ITWO(I)=IPASS/10000
      IMULT=ITWO(I)*10000
      IPAST=IPASS-IMULT
      IONE(I)=IPAST/100
      IMULT=IONE(I)*100
      IEMT(I)=IPAST-IMULT
C
C     DETERMINE TYPE, NUMBER, LOAD, AND DIRECTION OF CARS
C
```

```
C
C     CALCULATE REVERSED CARS.
C
      ITWOR(I)=0
      IONER(I)=0
      IEMTR(I)=0
      ITWO(I)=ITWO(I)-ITWOR(I)
      IONE(I)=IONE(I)-IONER(I)
      IEMT(I)=IEMT(I)-IEMTR(I)
C
C     RECORD TOTAL NUMBER AND DIRECTION OF CARS.
C
      ITCARS=ITCARS+ICARS(I)
      ITONE=ITONE+IONE(I)
      ITTWO=ITTWO+ITWO(I)
      ITEMT=ITEMT+IEMT(I)
      ITONER=ITONER+IONER(I)

      ITTWOR=ITTWOR+ITWOR(I)
      ITEMTR=ITEMTR+IEMTR(I)
C
C     PLACE CARS INTO INBOUND YARD.
C
      NUMCAR=ICARS(I)
  106 DO 115 J=1,NUMCAR
      ATT(1)=TYPE(I)
      ATT(4)=0.0
      ATT(5)=0.0
      ATT(6)=0.0
      ATT(7)=0.0
      IF(IONE(I).EQ.0) GO TO 107
      ATT(2)=1.
      ATT(3)=1.
      IONE(I)=IONE(I)-1
      GO TO 112
  107 IF(ITWO(I).EQ.0) GO TO 108
      ATT(2)=2.
      ATT(3)=1.
      ITWO(I)=ITWO(I)-1
      GO TO 112
  108 IF(IEMT(I).EQ.0) GO TO 109
      ATT(2)=0.
      ATT(3)=1.
      IEMT(I)=IEMT(I)-1
      GO TO 112
  109 IF(IONER(I).EQ.0) GO TO 110
      ATT(2)=1.
      ATT(3)=0.
      IONER(I)=IONER(I)-1
      GO TO 112
  110 IF(ITWOR(I).EQ.0) GO TO 111
      ATT(2)=2.
      ATT(3)=0.
      ITWOR(I)=ITWOR(I)-1
      GO TO 112
  111 IF(IEMTR(I).EQ.0) GO TO 115
      ATT(2)=0.
      ATT(3)=0.
      IEMTR(I)=IEMTR(I)-1
  112 CALL PTIN(10,ARR,TNOW,ATT)
C
```

```
C      SCHEDULE CONTAINER PICK-UP
C
       IF(NGND.EQ.1) GO TO 115
       IPICUP=2
       DO 114 L=1,IPICUP
       IF(ATT(2).EQ.0.0) GO TO 115
       PICUP=UN(6)
       CALL PTIN(60,PICUP,TNOW,ATT)
C
C      WILL THERE BE A CONTAINER DELIVERY WHEN PICK-UP IS MADE?
C
       PN=DRAND(L)
       IF(PN.GT.0.5) GO TO 113
       DN=DRAND(L)
       ATT(4)=4.
       IF(DN.GT.0.25) ATT(4)=3.
       IF(DN.GT.0.50) ATT(4)=2.
       IF(DN.GT.0.75) ATT(4)=1.
       ATT(5)=2.
       IF(DN.LT.0.10) ATT(5)=1.
       ATT(6)=112.0
       CALL PTIN(80,PICUP,TNOW,ATT)
  113  IF(ATT(2).EQ.1.0) GO TO 115
  114  CONTINUE
  115  CONTINUE
  116  CONTINUE
  117  CONTINUE
C
C      ARE THERE TRAINS SCHEDULED TO DEPART THIS HOUR?
C
  120  IF(INIT.EQ.0) GO TO 160
       J=10
       IMT=0
       DO 135 I=1,J
       DEPART=DTIME(IDAY,I)
       IF(DEPART.LT.25.0) GO TO 125
       GO TO 135
  125  IDEPAR=IDEPT(IDAY,I)
C
C      A DEPARTURE, DETERMINE DIRECTION.
C
       IF(IDEPAR.LT.4000) GO TO 126
C
C      A SOUTH BOUND DEPARTURE
C
       IDEPAR=IDEPAR-4000
       IQ=33
       ISER=18
       IOUT=34
       GO TO 129
C
  126  IF(IDEPAR.LT.3000) GO TO 127
C
C      A NORTH BOUND DEPARTURE.
C
       IDEPAR=IDEPAR-3000
       IQ=31
       ISER=17
       IOUT=32
       GO TO 129
C
```

```
  127 IF(IDEPAR.LT.2000) GO TO 128
C
C     A WEST BOUND DEPARTURE
C
      IDEPAR=IDEPAR-2000
      IQ=29
      ISER=16
      IOUT=30
      GO TO 129
C
C     AN EAST BOUND DEPARTURE.
  128 IDEPAR=IDEPAR-1000
      IQ=27
      ISER=15
      IOUT=28
C
C     DETERMINE MAXIMUM NUMBER OF CARS AVAILABLE FOR TRAIN.
C
  129 ISS=ISTUS(IQ,ISER)
      IF(ISS.EQ.0) GO TO 134
      INQ=XNINQ(IQ)
      INQ=INQ+1
      IGO=IDEPAR-INQ
      IF(IGO.GE.0) GO TO 130
      IGO=IDEPAR
      GO TO 131
  130 IGO=INQ
C
C     SHIP CARS OUT-BOUND.
C
  131 DO 132 K=1,IGO
      CALL STAGO(ISER,IOUT,DEPART,0,ATT)
  132 CONTINUE
  134 IF(IMT.EQ.136) GO TO 137
      IF(ISS.EQ.0) GO TO 135
      ATT(1)=IGO
      ATT(2)=ISER
      ATT(3)=IOUT
      ATT(4)=IDEPAR
      ATT(5)=IQ
      ATT(6)=132.0
      ATT(7)=DEPART
      CALL PTIN(80,DEPART,TNOW,ATT)
      ATT(6)=0.0
      ATT(7)=0.0
  135 CONTINUE
      GO TO 140
  136 IQ=ATT(5)
      ISER=ATT(2)
      IDEPAR=ATT(4)-ATT(1)
      IOUT=ATT(3)
      DEPART=0.0
      IMT=136
      GO TO 129
  137 IMT=0
      ATT(6)=0.0
      ATT(7)=0.0
      RETURN
C
C     IS THERE A SWITICHING SCHEDULED THIS HOUR?
C
```

```
  140 IF(IWORK.EQ.0) RETURN
      IF(INIT.EQ.1) GO TO 185
      GO TO 160
  145 RN=DRAND(1)
      IF(RN.GT.0.2) RETURN
      GO TO 186
C
C     THE RAMP IS OCCUPIED
C
  149 IF((IHOUR.GT.8.).AND.(IHOUR.LT.18)) GO TO 151
      GO TO 156
C
C     THE TIME IS BETWEEN 6PM AND 8AM.
C     OVERTIME/SECOND SHIFT?
C
  150 ON=DRAND(1)
      IF(ON.GT.0.0) RETURN
      OTIME=OTIME+4.0
C
C     THE TIME IS BETWEEN 8AM AND 6PM.
C     ARE THERE CONTAINERS TO BE OFF LOADED?
C
  151 IF(LOADOF.EQ.1) RETURN
      ISS=ISTUS(21,8)
      IF(ISS.EQ.0) GO TO 153
      LOADOF=1
      TIMEOF=UN(5)
      POI=DRAND(1)
      IF(POI.GT.0.75) GO TO 152
      EPOI=DRAND(2)
      IPOI=PO(4)
      TIMEOF=IPOI+EPOI
  152 NOFF=NOFF+1
      TOFF(IDAY,NOFF)=0.0
      LCOFF(IDAY,NOFF)=0
      CALL STAGO(8,22,TIMEOF,0,ATT)
      RETURN
C
C     NO CONTAINERS TO OFF LOAD.  CAN CONTAINERS BE LOADED?
C
  153 IF(LOADON.EQ.1) RETURN
      IF(LOADOF.EQ.1) RETURN
C
C     THERE IS AN ON-LOAD INPROGRESS.
C
      ISS=ISTUS(23,9)
      IF(ISS.EQ.0) RETURN
      LOADON=1
      STONLD=15-IHOUR
      TLDM=DRAND(1)
      TLD=DRAND(2)
      TIMEON=0.0
      IF(STONLD) 155,154,154
  154 TIMEON=STONLD+TLDM
      IF(TLD.LE.0.92) GO TO 155
      TIMEON=TIMEON+1.0
      IF(TLD.LE.0.97) GO TO 155
      TIMEON=TIMEON+1.0
  155 NON=NON+1
      TON(IDAY,NON)=0.0
      LCON(IDAY,NON)=0
```

```
      CALL STAGO(9,24,TIMEON,0,ATT)
      RETURN
C
C     NO CONTAINERS TO ON-LOAD.  CAN CARS BE SWITCHED FROM RAMP?
C
  156 IF(LOADOU.EQ.1) RETURN
      IF(LOADON.EQ.1) RETURN
      IF((IHOUR.GT.8).AND.(IHOUR.LT.18)) RETURN
C     A SWITCH TO OUTBOUND YARD IS IN PROGRESS.
C
      ISS=ISTUS(25,10)
      IF(ISS.EQ.0) GO TO 150
      LOADOU=1
      TIMEOU=NO(3)
      NOUT=NOUT+1
      TSOUT(IDAY,NOUT)=TIMEOU
      NSWOUT(IDAY,NOUT)=1
      CALL STAGO(10,26,TIMEOU,0,ATT)
      RETURN
C
C     SET PARKING LOT AT START TIME.
C
  160 DO 175 L=1,ISTART
      IF(ITYP1E.EQ.0) GO TO 161
      ATT(1)=1.
      ATT(4)=1.
      ITYP1E=ITYP1E-1
      GO TO 172
  161 IF(ITYP1W.EQ.0) GO TO 162
      ATT(1)=1.
      ATT(4)=2.
      ITYP1W=ITYP1W-1
      GO TO 172
  162 IF(ITYP1N.EQ.0) GO TO 163
      ATT(1)=1.
      ATT(4)=3.
      ITYP1N=ITYP1N-1
      GO TO 172
  163 IF(ITYP1S.EQ.0) GO TO 164
      ATT(1)=1.
      ATT(4)=4.
      ITYP1S=ITYP1S-1
      GO TO 172
  164 IF(ITYP2E.EQ.0) GO TO 165
      ATT(1)=2.
      ATT(4)=1.
      ITYP2E=ITYP2E-1
      GO TO 172
  165 IF(ITYP2W.EQ.0) GO TO 166
      ATT(1)=2.
      ATT(4)=2.
      ITYP2W=ITYP2W-1
      GO TO 172
  166 IF(ITYP2N.EQ.0) GO TO 167
      ATT(1)=2.
      ATT(4)=3.
      ITYP2N=ITYP2N-1
      GO TO 172
  167 IF(ITYP2S.EQ.0) GO TO 168
      ATT(1)=2.
```

```
      ITYP2S=ITYP2S-1
      GO TO 172
  168 IF(ITYP3E.EQ.0) GO TO 169
      ATT(1)=3.
      ATT(4)=1.
      ITYP3E=ITYP3E-1
      GO TO 172
  169 IF(ITYP3W.EQ.0) GO TO 170
      ATT(1)=3.
      ATT(4)=2.
      ITYP3W=ITYP3W-1
      GO TO 172
  170 IF(ITYP3N.EQ.0) GO TO 171
      ATT(1)=3.
      ATT(4)=3.
      ITYP3N=ITYP3N-1
      GO TO 172
  171 IF(ITYP3S.EQ.0) GO TO 172
      ATT(1)=3.
      ATT(4)=4.
      ITYP3S=ITYP3S-1
  172 OC=DRAND(1)
      IF(OC.GT.0.05) GO TO 173
      ATT(5)=1.
      GO TO 174
  173 ATT(5)=2.
  174 CALL PTIN(42,0.0,TNOW,ATT)
      MXPARK=MXPARK-1
  175 CONTINUE
      IPARK5=IPARK1+IPARK2+IPARK3
      DO 179 I=1,IPARK5
      IF(IPARK1.EQ.0) GO TO 176
      ATT(1)=1.0
      ATT(2)=176.0
      CALL PTIN(49,0.0,TNOW,ATT)
      IPARK1=IPARK1-1
      GO TO 178
  176 IF(IPARK2.EQ.0) GO TO 177
      ATT(1)=2.0
      ATT(2)=176.0
      CALL PTIN(50,0.0,TNOW,ATT)
      IPARK2=IPARK2-1
      GO TO 178
  177 IF(IPARK3.EQ.0)  GO TO 179
      ATT(1)=3.0
      ATT(2)=176.0
      CALL PTIN(50,0.0,TNOW,ATT)
      IPARK3=IPARK3-1
  178 SPICUP=UN(8)
      CALL PTIN(60,SPICUP,TNOW,ATT)
      MXPARK=MXPARK-1
  179 CONTINUE
C
C     SET OUT-BOUND YARD AT START TIME.
C
      IF(ISOUTH.EQ.0) GO TO 184
      DO 180 M=1,IEAST
      CALL PTIN(27,0.0,TNOW,ATT)
  180 CONTINUE
      DO 181 M=1,IWEST
      CALL PTIN(29,0.0,TNOW,ATT)
```

```
  181 CONTINUE
      DO 182 M=1,INORTH
      CALL PTIN(31,0.0,TNOW,ATT)
  182 CONTINUE
      DO 183 M=1,ISOUTH
      CALL PTIN(33,0.0,TNOW,ATT)
  183 CONTINUE
  184 LOADOU=0
      LOADOF=0
      LOADON=0
      ISWITC=0
      ISWIN=0
      ISWOUT=0
      ATT(1)=160.0
      INIT=1
      ICF=160
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
  185 IF(ISWIN.EQ.1) GO TO 149
      IF(ISWOUT.EQ.1) GO TO 149
      IF((IHOUR.GT.8.).AND.(IHOUR.LT.18)) GO TO 145
C
C     THE RAMP IS EMPTY AND THE TIME IS BETWEEN 6PM AND 8AM.
C     SCHEDULE A SWITCH TO THE RAMP.
C
  186 DO 209 I=1,3
      NOLOAD=0
      K=12+I
      IS(I)=ISTUS(K,I)
      IF(IS(I).EQ.0) GO TO 209
C
C     THERE ARE LOADED CARS IN THE YARD.
C     SCHEDULE A SWITCH TO THE RAMP.
C
      IF(IPRI1.EQ.0) GO TO 188
      IF(IPRI1.EQ.4) GO TO 194
      IQ=IPRI1
      KQ= 12+IQ
      IST=ISTUS(KQ,IQ)
      IF(IST.EQ.0) GO TO 187
      GO TO 204
C
C     THERE ARE NO PRIORITY 1 CONTAINERS.  CHECK PRIOTITY 2.
C
  187 IF(IPRI2.EQ.0) GO TO 188
      IF(IPRI2.EQ.4) GO TO 194
      IQ=IPRI2
      KQ=12+IQ
      IST=ISTUS(KQ.IQ)
      IF(IST.EQ.1) GO TO 204
C
C     THERE ARE NO PRIORITY 2 CONTAINERS.
C     SWITCH CONTAINERS THAT ARE IN YARD.
C
      IQ=I
      KQ=K
      GO TO 204
C
C     GIVE PRIORITY TO TYPE OF CAR WITH THE LARGEST NUMBER IN YARD.
C
  188 MAX13=XNINQ(13)
```

```
      MAX14=XNINQ(14)
      MAX15=XNINQ(15)
      IF(MAX13-MAX14) 189,191,191
  189 IF(MAX14-MAX15) 192,190,190
  190 IQ=2
      KQ=14
      GO TO 204
  191 IF(MAX13-MAX15) 192,193,193
  192 IQ=3
      KQ=15
      GO TO 204
  193 IQ=1
      KQ=13
      GO TO 204
C
C     GIVE PRIORITY TO THE CAR WAITING THE LONGEST IN THE YARD.
C     IF TIME IS EQUAL, SET PRIORITY FOR THE LARGEST NUMBER IN QUEUE.
C
  194 TTCS1=REMST(1)
      TTCS2=REMST(2)
      TTCS3=REMST(3)
      IF(TTCS1.EQ.0.0) GO TO 197
      IF(TTCS2.EQ.0.0) GO TO 195
      IF(TTCS1-TTCS2) 195,200,197
  195 IF(TTCS3.EQ.0.0) GO TO 196
      IF(TTCS1-TTCS3) 196,203,199
  196 IQ=1
      KQ=13
      GO TO 204
  197 IF(TTCS2.EQ.0.0) GO TO 199
      IF(TTCS3.EQ.0.0) GO TO 198
      IF(TTCS2-TTCS3) 198,201,199
  198 IQ=2
      KQ=14
      GO TO 204
  199 IQ=3
      KQ=15
      GO TO 204
  200 IF(TTCS3.EQ.0.0) GO TO 188
      IF(TTCS3-TTCS2) 199,188,202
  201 MAX13=0
      MAX14=XNINQ(14)
      MAX15=XNINQ(15)
      GO TO 189
  202 MAX13=XNINQ(13)
      MAX14=XNINQ(14)
      IF(MAX13-MAX14) 190,193,193
  203 MAX13=XNINQ(13)
      MAX14=0
      MAX15=XNINQ(15)
      GO TO 191
C
C     MOVE CARS FROM YARD TO RAMP.
C
  204 TIMESW=NO(3)
      NSWIN=NSWIN+1
      TSWIN(IDAY,NSWIN)=TIMESW
      ISWIN=1
      ISWOUT=1
      YARDIN=XNINQ(13)+XNINQ(14)+XNINQ(15)+XNINQ(16)+XNINQ(17)+XNINQ(1)
      II=12
```

```
      DO 208 IA=1,6
      II=II+1
      ISY=ISTUS(II,IA)
      IF(ISY.EQ.0) GO TO 208
      YARDIN=YARDIN+1
  208 CONTINUE
      CALL COL(YARDIN,3)
      CALL STAGO(IQ,19,TIMESW,0,ATT)
      RETURN
  209 CONTINUE
      J=6
      DO 210 I=4,J
      K=12+I
      IS(I)=ISTUS(K,I)
      IF(IS(I).EQ.0) GO TO 210
      L=K+27
      MI=7+I
      IP(I)=ISTUS(L,MI)
      IF(IP(I).EQ.0) GO TO 210
      NOLOAD=1
      KQ=K
      IQ=I
      GO TO 204
  210 CONTINUE
      IF(ISWIN.EQ.0) RETURN
      GO TO 340
C
C      CAR IS REVERSED.  TURN CAR AND GET NEXT CAR FROM YARD.
C
  290 GO TO 301
C
C     PLACE CAR ON RAMP SIDING AND GET NEXT CAR.
C
  300 ISWITC=ISWITC+1
      IF(ISWITC.EQ.MXRAMP) GO TO 340
  301 ISR=ISTUS(37,7)
      IF(ISR.EQ.0) GO TO 310
      CALL STAGO(7,20,0.0,0,ATT)
      RETURN
  310 IF(NOLOAD.EQ.1) GO TO 186
      ISSER=ISTUS(KQ,IQ)
      IF(ISSER.EQ.0) GO TO 186
      TIMESW=NO(3)
      TSWIN(IDAY,NSWIN)=TSWIN(IDAY,NSWIN)+TIMESW
      CALL STAGO(IQ,19,0.0,0,ATT)
      RETURN
C
C     NO ADDITIONAL CARS CAN BE SWITCHED.
C
  340 ATT(6)=340.0
      ISWINT(IDAY,NSWIN)=ISWITC
      CALL PTIN(80,TSWIN(IDAY,NSWIN),TNOW,ATT)
      RETURN
  350 ISWIN=0
      ISWOUT=1
      NOLOAD=0
      ISWITC=0
      LOADOF=0
      LOADOU=0
      LOADON=0
      GO TO 185
```

```
C
C     OFF LOAD CARS.
C
  400 CALL GETAT(ATT)
      INQ=ATT(1)+48.
      IF(ATT(2).EQ.0.0) GO TO 416
      J=ATT(2)
      DO 405 I=1,J
      IF(MXPARK.EQ.0) GO TO 410
      MXPARK=MXPARK-1
      UNTIME=EX(7)
      TOFF(IDAY,NOFF)=TOFF(IDAY,NOFF)+UNTIME
      LCOFF(IDAY,NOFF)=LCOFF(IDAY,NOFF)+1
      CALL PTIN(INQ,UNTIME,TNOW,ATT)
      IF(NGND.EQ.1) GO TO 411
  405 CONTINUE
      GO TO 416
  410 UNTIME=EX(7)
      TOFF(IDAY,NOFF)=TOFF(IDAY,NOFF)+UNTIME
      LCOFF(IDAY,NOFF)=LCOFF(IDAY,NOFF)+1
      ATT(2)=200.0
      CALL PTIN(70,UNTIME,TNOW,ATT)
  411 XNOTFI=UN(14)
      PICKUP=UN(6)
      PHOUR=IHOUR
      UPICUP=PHOUR+XNOTFI+PICKUP
      IF(UPICUP.GT.17.0) GO TO 412
      CALL PTIN(60,UPICUP,TNOW,ATT)
      GO TO 405
  412 QHOUR=PHOUR-8.0
      IF(QHOUR) 413,413,414
  413 HOUR8=-QHOUR
      GO TO 415
  414 HOUR8=24.0-QHOUR
  415 PICKUP=PICKUP+HOUR8
      CALL PTIN(60,PICKUP,TNOW,ATT)
      GO TO 405
  416 ISS=ISTUS(21,8)
      IF(ISS.EQ.0) GO TO 420
      CALL STAGO(8,22,0.0,0,ATT)
      RETURN
  420 ATT(6)=420.0
      CALL PTIN(80,TOFF(IDAY,NOFF),TNOW,ATT)
      RETURN
  430 LOADOF=0
      GO TO 185
C
C     ON LOAD CARS
C
  500 ICF=500
      CALL GETAT(ATT)
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
  501 ITYPE=ATT(1)+10.
      IQP=ITYPE+32
      ISS=ISTUS(IQP,ITYPE)
      IF(ISS.EQ.1) GO TO 505
      ATT(4)=5.0
      CALL PTIN(25,0.0,TNOW,ATT)
      ISS=ISTUS(23,9)
      IF(ISS.EQ.0) GO TO 520
```

```
      CALL STAGO(9,24,0.0,0,ATT)
      RETURN
  505 ICF=505
      CALL STAGO(ITYPE,80,0.0,0,ATT)
      RETURN
  506 IF(ATT(5).EQ.1.0) GO TO 515
      ISS=ISTUS(IQP,ITYPE)
      IF(ISS.EQ.0) GO TO 515
      A=ATT(4)
      B=ATT(5)
      ICF=506
      CALL STAGO(ITYPE,80,0.0,0,ATT)
      RETURN
  507 MAX=XNINQ(IQP)
      IF(MAX.EQ.0) GO TO 521
      DO 510 I=1,MAX
      IF(A.EQ.ATT(4)) GO TO 508
      GO TO 509
  508 IF(B.EQ.ATT(5)) GO TO 511
  509 ICF=509
      CALL PTIN(IQP,0.0,TIMEM,ATT)
      CALL STAGO(ITYPE,80,0.0,0,ATT)
      RETURN
  510 CONTINUE
      ATT(4)=A
      ATT(5)=1.
      GO TO 515
  511 MXPARK=MXPARK+2
      ICF=511
      OLTIME=EX(9)
      OLTIME=OLTIME+EX(9)
      LCON(IDAY,NON)=LCON(IDAY,NON)+2
      TON(IDAY,NON)=TON(IDAY,NON)+OLTIME
      CALL PTIN(25,OLTIME,TNOW,ATT)
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
  515 MXPARK=MXPARK+1
      ICF=515
      OLTIME=EX(9)
      LCON(IDAY,NON)=LCON(IDAY,NON)+1
      TON(IDAY,NON)=TON(IDAY,NON)+OLTIME
      CALL PTIN(25,OLTIME,TNOW,ATT)
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
  516 ISS=ISTUS(23,9)
      IF(ISS.EQ.0) GO TO 517
      CALL STAGO(9,24,0.0,0,ATT)
      RETURN
  517 ATT(6)=517.0
      CALL PTIN(80,TON(IDAY,NON),TNOW,ATT)
      RETURN
  520 LOADON=0
      GO TO 185
  521 IF(A.EQ.ATT(4)) GO TO 523
  522 CALL PTIN(IQP,0.0,TIMEM,ATT)
      ATT(4)=A
      ATT(5)=1.0
      GO TO 515
  523 IF(B.EQ.ATT(5)) GO TO 511
      GO TO 522
C     SWITCH CARS TO OUT-BOUND YARD.
```

```
C
  600 ISS=ISTUS(25,10)
      IF(ISS.EQ.0) GO TO 604
      NSWOUT(IDAY,NOUT)=NSWOUT(IDAY,NOUT)+1
      TIMEOU=NO(3)
      TSOUT(IDAY,NOUT)=TSOUT(IDAY,NOUT)+TIMEOU
      CALL STAGO(10,26,0.0,0,ATT)
      RETURN
  604 ATT(6)=604.0
      CALL PTIN(80,TSOUT(IDAY,NOUT),TNOW,ATT)
      RETURN
  605 LOADOU=0
      ISWOUT=0
      GO TO 185
C
C     EMPTY FROM RAMP.
C
  700 CALL GETAT(ATT)
      ATT(2)=0.0
      ATT(3)=1.0
      ICF=700
      E=ATT(1)
      IE=E+15
      NUM(IE)=XNINQ(IE)
      IF(NUM(IE).GT.5) GO TO 701
      CALL PTIN(IE,0.0,TNOW,ATT)
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
  701 ICF=701
      CALL PTIN(39,0.0,TNOW,ATT)
      CALL PTIN(80,0.0,TNOW,ATT)
      RETURN
C
C     RANDOM OVER-THE-ROAD CONTAINER ARRIVALS
C
C     DETERMINE TYPE OF CONTAINER
C
  800 TCON=1.0
      TC=DRAND(1)
      IF(TC.LT.0.8) GO TO 805
      TCON=2.0
      IF(TC.LT.0.95) GO TO 805
      TCON=3.0
  805 ATT(1)=TCON
C
C     DETERMINE DESTINATION OF CONTAINER
      DCON=1.0
      DC=DRAND(2)
      IF(DC.GT.0.75) GO TO 810
      DCON=2.0
      IF(DC.GT.0.50) GO TO 810
      DCON=3.0
      IF(DC.GT.0.25) GO TO 810
      DCON=4.0
  810 ATT(4)=DCON
C
C     DETERMINE IF CONTAINER REQUIRES SINGLE CAR.
C
      PERCAR=2.0
      PC=DRAND(3)
      IF(PC.LT.0.90) GO TO 815
```

```
      PERCAR=1.0
  815 ATT(5)=PERCAR
      ATT(2)=0.0
      ATT(3)=0.0
C
C     DETERMINE IF PARKING LOT IS AT CAPACITY.
C
      IF(MXPARK.EQ.0) GO TO 820
      MXPARK=MXPARK-1
      CALL PTIN(42,0.0,TNOW,ATT)
      RETURN
  820 CALL PTIN(70,0.0,TNOW,ATT)
      RETURN
C
C     CURRENT ATTRIBUTE FUNCTION
C
  900 TIMEM=TMARK(IDUM)
      CALL GETAT(ATT)
      IF(ATT(6).EQ.112.0) GO TO 925
      IF(ATT(6).EQ.132.0) GO TO 136
      IF(ATT(6).EQ.340.0) GO TO 350
      IF(ATT(6).EQ.420.0) GO TO 430
      IF(ATT(6).EQ.517.0) GO TO 520
      IF(ATT(6).EQ.604.0) GO TO 605
      IF(ATT(6).EQ.965.0) GO TO 920
      IF(ICF.EQ.63) GO TO 64
      IF(ICF.EQ.160) GO TO 120
      IF(ICF.EQ.500) GO TO 501
      IF(ICF.EQ.505) GO TO 506
      IF(ICF.EQ.506) GO TO 507
      IF(ICF.EQ.509) GO TO 510
      IF(ICF.EQ.511) GO TO 516
      IF(ICF.EQ.515) GO TO 516
      IF(ICF.EQ.700) GO TO 600
      IF(ICF.EQ.701) GO TO 600
  920 DN=DRAND(1)
      RN=DRAND(2)
      ATT(2)=920.0
      ATT(4)=2.0
      IF(DN.GT.0.79) ATT(4)=3.0
      IF(DN.GT.0.93) ATT(4)=1.0
      ATT(5)=2.0
      IF(RN.LT.0.10) ATT(5)=1.0
  925 ATT(6)=0.0
      IF(MXPARK.EQ.0) GO TO 930
      MXPARK=MXPARK-1
      CALL PTIN(41,0.0,TIMEM,ATT)
      RETURN
  930 CALL PTIN(70,0.0,TIMEM,ATT)
      RETURN
  950 MXPARK=MXPARK+1
      IF(NGND.EQ.0) RETURN
      CALL GETAT(ATT)
      IRTCON=EX(2)
      RTCON=IRTCON*24
      NHOUR=IHOUR-8
      IF(NHOUR) 955,955,960
  955 HOURD=-NHOUR
      GO TO 965
  960 HOURD=24-NHOUR
  965 DELV=UN(6)
      DELVT=DELV+RTCON+HOURD
      ATT(6)=965.0
      CALL PTIN(80,DELVT,TNOW,ATT)
      RETURN
      END
```

APPENDIX E

Computer Program Listing for Subroutine UO for the Intermodal Rail Terminal Model

```
      SUBROUTINE UO
      COMMON/QVAR/NDE,NFTBU(100),NREL(100).NRELP(100),NREL2(100),
     1NRUN,NRUNS,NTC(100),PARAM(100,4).TBEG,TNOW
      COMMON/UCOM1/IHOUR,IDAY,ATIME(90,10).ITP1(90,10),ITP2(90,10),
     1ITP3(90,10),DTIME(90,10),IDEPT(90,10),ITCARS.ITONE,ITTWO.
     2ITEMT.ITONER,ITEMTR,MXRAMP,MXPARK,ISTART,ITYP1E,ITYP1W.ITYP1N,
     3ITTWOR,ITYP1S.ITYP2E,ITYP2W.ITYP2N.ITYP2S,ITYP3E,ITYP3W,ITYP3N.
     4ITYP3S,IEAST.IWEST,INORTH,ISOUTH,IPRI1,IPRI2,INIT,IPARK1,IPARK2,
     5IPARK3,IPARKT(90),IOVP(90),MOVE(90)
      COMMON/UCOM2/TSWIN(90,10),ISWINT(90,10),TOFF(90,10),TON(90,10)
     1,LCON(90,10).NSWOUT(90.10),TSOUT(90,10),LCOFF(90,10),MCAP,
     2NSWIN,NOFF.NON,NOUT,ISAT,ISUN,IWORK.NGND,INPRO,
      COMMON/USTAT/UOBV(25,5),XYZZZ(906)
      DIMENSION OAVG(25),OSD(25),OSDA(25)
      DATA CPFH.CPFD.CPCH,CPCD,CPSH/.42,10.08,.217,5.2,226.2/
      IF(NRUN.GT.1) GO TO 1005
      YDMAX=0.0
      SYDMAX=0.0
 1005 ORUN=NRUN
      ODAY=IDAY
      XOP=0.0
      DOP=0.0
      DO 1010 I=1,IDAY
      IF(IOVP(I).LE.0) GO TO 1010
      YZ=IOVP(I)
      XOP=XOP+YZ
      DOP=DOP+1.0
 1010 CONTINUE
      IF(DOP.EQ.0.0) GO TO 1015
      AXOP=XOP/DOP
      CALL COL(AXOP,1)
      CALL COL(DOP,2)
 1015 XMV=0.0
      XMD=0.0
      DO 1020 I=1,IDAY
      IF(MOVE(I).LE.0) GO TO 1020
      ZY=MOVE(I)
      XMV=XMV+ZY
      XMD=XMD+1.0
 1020 CONTINUE
      IF(XMD.EQ.0.0) GO TO 1025
      AXMV=XMV/XMD
 1025 PKT=0.0
      DO 1030 I=1,IDAY
      ZZ=IPARKT(I)
      PKT=PKT+ZZ
 1030 CONTINUE
      ZDAY=IDAY
      APKT=PKT/ZDAY
      TPK=PKT+XOP
      ATPKT=TPK/ZDAY
      CALL COL(APKT,7)
      CALL COL(ATPKT,8)
      AV1=AVEWT(13)
      AV2=AVEWT(14)
      AV3=AVEWT(15)
      DAV=3.0
      IF(AV1.EQ.0.0) DAV=DAV-1.0
```

```
      IF(AV2.EQ.0.0) DAV=DAV-1.0
      IF(AV3.EQ.0.0) DAV=DAV-1.0
      IF(DAV.EQ.0.0) GO TO 1031
      WTINB=AV1+AV2+AV3
      AVGWT=WTINB/DAV
      GO TO 1032
1031  AVGWT=0.0
1032  AV4=AVEWT(16)
      AV5=AVEWT(17)
      AV6=AVEWT(18)
      RAV=3.0
      IF(AV4.EQ.0.0) RAV=RAV-1.0
      IF(AV5.EQ.0.0) RAV=RAV-1.0
      IF(AV6.EQ.0.0) RAV=RAV-1.0
      IF(RAV.EQ.0.0) GO TO 1033
      REVT=AV4+AV5+AV6
      AVGRT=REVT/RAV
      AVGRT=0.0
      GO TO 1034
1033  AVGRT=0.0
1034  AV7=AVEWT(21)
      AV8=AVEWT(23)
      AV9=AVEWT(25)
      AVGRMP=AV7+AV8+AV9
      TAVGWT=AVGWT+AVGRMP+AVGRT
      AV10=AVEWT(27)
      AV11=AVEWT(29)
      AV12=AVEWT(31)
      AV13=AVEWT(33)
      OAV=4.0
      IF(AV10.EQ.0.0) OAV=OAV-1.0
      IF(AV11.EQ.0.0)OAV=OAV-1.0
      IF(AV12.EQ.0.0) OAV=OAV-1.0
      IF(AV13.EQ.0.0) OAV=OAV-1.0
      IF(OAV.EQ.0.0) GO TO 1035
      WTOTB=AV10+AV11+AV12+AV13
      AVGOT=WTOTB/OAV
      GO TO 1036
1035  AVGOT=0.0
1036  TAVGWT=TAVGWT+AVGOT
      XSWIN=0.0
      XISW=0.0
      XTOFF=0.0
      XCOFF=0.0
      XTON=0.0
      XCON=0.0
      XSOUT=0.0
      XNSOUT=0.0
      DO 1050 I=1,IDAY
      DO 1045 K=1,10
      IF(TSWIN(I,K).LE.0.0) GO TO 1040
      XSWIN=XSWIN+TSWIN(I,K)
      XY1=ISWINT(I,K)
      XISW=XISW+XY1
1040  IF(TOFF(I,K).LE.0.0) GO TO 1041
      XTOFF=XTOFF+TOFF(I,K)
      XY2=LCOFF(I,K)
      XCOFF=XCOFF+XY2
```

```
 1041 IF(TON(I,K).LE.0.0) GO TO 1042
      XTON=XTON+TON(I,K)
      XY3=LCON(I,K)
      XCON=XCON+XY3
 1042 IF(TSOUT(I,K).LE.0.0) GO TO 1045
      XSOUT=XSOUT+TSOUT(I,K)
      XY4=NSWOUT(I,K)
      XNSOUT=XNSOUT+XY4
 1045 CONTINUE
 1050 CONTINUE
      ASWIN=XSWIN/NTC(20)
      AOFF=XTOFF/XCOFF
      ATON=XTON/XCON
      ASOUT=XSOUT/XNSOUT
      TAVGWT=TAVGWT+ASWIN+ASOUT
      TOTSW=XSWIN+XSOUT
      CALL COL(TOTSW,11)
      CALL COL(ASWIN,12)
      CALL COL(ASOUT,13)
      CALL COL(TAVGWT,14)
      CALL COL(AOFF,15)
      CALL COL(ATON,16)
      CALL COL(XCOFF,17)
      CALL COL(XCON,18)
      CONIN=XCOFF/NTC(20)
      CWTIN=CONIN*AVGWT
      CSWIN=CONIN*ASWIN
      CRMPIN=CONIN*AVEWT(21)
      AV14=AVEWT(49)
      AV15=AVEWT(50)
      AV16=AVEWT(51)
      CIN=3.0
      IF(AV14.EQ.0.0) CIN=CIN-1.0
      IF(AV15.EQ.0.0) CIN=CIN-1.0
      IF(AV16.EQ.0.0) CIC=CIN-1.0
      IF(CIN.EQ.0.0) GO TO 1054
      CINWT=AV14+AV15+AV16
      CINAV=CINWT/CIN
      CINAWT=CWTIN+CSWIN+AOFF+CINAV+CRMPIN
      GO TO 1055
 1054 CINAWT=0.0
 1055 CALL COL(CINAWT,21)
      CONOUT=XCON/XNSOUT
      CWTOUT=CONOUT*AVGOT
      CSWOUT=CONOUT*ASOUT
      CWRMP=CONOUT*AVEWT(25)
      AV17=AVEWT(43)
      AV18=AVEWT(44)
      AV19=AVEWT(45)
      COUT=3.0
      IF(AV17.EQ.0.0) COUT=COUT-1.0
      IF(AV18.EQ.0.0) COUT=COUT-1.0
      IF(AV19.EQ.0.0) COUT=COUT-1.0
      IF(COUT.EQ.0.0) GO TO 1056
      COUTWT=(AV17+AV18+AV19)/COUT
      COUTAV=COUTWT+ATON+CSWOUT+CWTOUT+CWRMP
      GO TO 1057
 1056 COUTAV=0.0
```

```
1057 CALL COL(COUTAV,22)
     CKIN=(XNINQ(13)+XNINQ(14)+XNINQ(15))*CONIN
     CKARS=NTC(10)
     CKOUT=NTC(42)
     CKIN1=XNINQ(49)+XNINQ(50)+XNINQ(51)
     CKIN2=NTC(49)+NTC(50)+NTC(51)
     CKIN3=CKIN+CKIN1+CKIN2
     CALL COL(CKARS,23)
     CALL COL(CKOUT,24)
     CALL COL(CKIN3,25)
     YDMAX=YDMAX+UOBV(3,5)
     SYDMAX=SYDMAX+(UOBV(3,5)**2)
     OAF1=(TINIQ(13)+TINIQ(14)+TINIQ(15))/TNOW
     OAF11=(TISS(13,1)+TISS(14,2)+TISS(15,3))/TNOW
     OAF2=(TINIQ(16)+TINIQ(17)+TINIQ(18))/TNOW
     OAF22=(TISS(16,4)+TISS(17,5)+TISS(18,6))/TNOW
     OAF3=(TINIQ(21)+TINIQ(23)+TINIQ(25)+TINIQ(37))/TNOW
     OAF33=(TISS(21,8)+TISS(23,9)+TISS(25,10)+TISS(37,7))/TNOW
     OAF4=(TINIQ(27)+TINIQ(29)+TINIQ(31)+TINIQ(33))/TNOW
     OAF44=(TISS(27,15)+TISS(29,16)+TISS(31,17)+TISS(33,18))/TNOW
     OAF5=UOBV(23,1)/ORUN
     OAF55=OAF5/ODAY
     OAF6=OAF1+OAF11+OAF3+OAF33+OAF4+OAF44+OAF55
     CALL COL(OAF6,4)
     OACI1=(OAF1+OAF11+((TINIQ(21)+TISS(21,8))/TNOW))*CONIN
     OACI2=(TINIQ(49)+TINIQ(50)+TINIQ(51))/TNOW
     OACI3=((TINIQ(70)+TISS(70,14))/TNOW)*0.5
     OACI4=UOBV(25,1)/ORUN
     OACI44=OACI4/ODAY
     OACI5= OACI1+OACI2+OACI3+OACI44
     CALL COL(OACI5,9)
     OACO1=(OAF4+OAF44+((TINIQ(25)+TISS(25,10))/TNOW))*CONOUT
     OACO2=(TINIQ(43)+TINIQ(44)+TINIQ(45))/TNOW
     OACO22=(TISS(43,11)+TISS(44,12)+TISS(45,13))/TNOW
     OACO3=UOBV(24,1)/ORUN
     OACO33=OACO3/ODAY
     OACO4=OACO1+OACO2+OACO22+OACO33
     CALL COL(OACO4,10)
     OINYD=OAF1+OAF11+OAF2+OAF22+OAF55
     CALL COL(OINYD,6)
     WDAY=2.0
     IF(ISAT.EQ.1) WDAY=WDAY-1.0
     IF(ISUN.EQ.1) WDAY=WDAY-1.0
     IF(WDAY.EQ.0.0) GO TO 1058
     OWORK=ODAY/7.0
     LDAY=OWORK*WDAY
     PDAY=LDAY
     WKDAY=ODAY-PDAY
     GO TO 1059
1058 WKDAY=ODAY
1059 OCOFPD=(UOBV(17,1)/ORUN)/WKDAY
     CALL COL(OCOFPD,5)
     OCONPD=(UOBV(18,1)/ORUN)/WKDAY
     CALL COL(OCONPD,19)
     OASWPD=(UOBV(11,1)/ORUN)/WKDAY
     CALL COL(OASWPD,20)
     IF(NRUN.EQ.NRUNS) GO TO 1060
     IF(NRUN.GT.1) GO TO 1160
```

```
1060 ARUN=ORUN-1.0
     DO 1065 I=1,25
     OAVG(I)=UOBV(I,1)/ORUN
     IF(NRUN.EQ.1) GO TO 1061
     OSD(I)=((UOBV(I,2)-((UOBV(I,1)**2)/ORUN))/ARUN)**0.5
     OSDA(I)=OSD(I)/(ORUN**0.5)
     GO TO 1065
1061 OSD(I)=0.0
     OSDA(I)=0.0
1065 CONTINUE
     OAVG(1)=UOBV(1,1)/UOBV(1,3)
     OAVG(2)=UOBV(2,1)/UOBV(2,3)
     OAVG(3)=YDMAX/ORUN
     IF(NRUN.EQ.1) GO TO 1066
     OSD(1)=((UOBV(1,2)-((UOBV(1,1)**2)/UOBV(1,3)))/(UOBV(1,3)-1))**0.5
     OSD(2)=((UOBV(2,2)-((UOBV(2,1)**2)/UOBV(2,3)))/(UOBV(2,3)-1))**0.5
     OSD(3)=((SYDMAX-((YDMAX**2)/ORUN))/ARUN)**0.5
     OSDA(1)=OSD(1)/(UOBV(1,3)**0.5)
     OSDA(2)=OSD(2)/(UOBV(2,3)**0.5)
     OSDA(3)=OSD(3)/(ORUN**0.5)
1066 WRITE(6,1100) IDAY,NRUN
     COAV14=OAVG(14)*CPFH
     COSD14=OSD(14)*CPFH
     CSDA14=OSDA(14)*CPFH
     COAV4=OAVG(4)*CPFD
     COSD4=OSD(4)*CPFD
     CSDA4=OSDA(4)*CPFD
     WRITE(6,1110) COAV14  ,COSD14,CSDA14,COAV4,COSD4,CSDA4
     COAV21=OAVG(21)*CPCH
     COSD21=OSD(21)*CPCH
     CSDA21=OSDA(21)*CPCH
     COAV9=OAVG(9)*CPCD
     COSD9=OSD(9)*CPCD
```

```
 CSDA9=OSDA(9)*CPCD
 WRITE(6,1111)COAV21,COSD21,CSDA21,COAV9,COSD9,CSDA9
 COAV22=OAVG(22)*CPCH
 COSD22=OSD(22)*CPCH
 CSDA22=OSDA(22)*CPCH
 COAV10=OAVG(10)*CPCD
 COSD10=OSD(10)*CPCD
 CSDA10=OSDA(10)*CPCD
 WRITE(6,1112) COAV22,COSD22,CSDA22,COAV10,COSD10,CSDA10
 COAV11=OAVG(11)*CPSH
 COSD11=OSD(11)*CPSH
 CSDA11=OSDA(11)*CPSH
 COAV20=OAVG(20)*CPSH
 COSD20=OSD(20)*CPSH
 CSDA20=OSDA(20)*CPSH
 COAV12=OAVG(12)*CPSH
 COSD12=OSD(12)*CPSH
 CSDA12=OSDA(12)*CPSH
 COAV13=OAVG(13)*CPSH
 COSD13=OSD(13)*CPSH
 CSDA13=OSDA(13)*CPSH
 WRITE(6,1113) COAV12,COSD12,CSDA12,COAV13,COSD13,CSDA13,
1COAV20,COSD20,CSDA20,COAV11,COSD11,CSDA11
 WRITE(6,1120) OAVG(4),OSD(4),OSDA(4),OAVG(9),OSD(9),OSDA(9),
1OAVG(10),OSD(10),OSDA(10)
```

```
      WRITE(6,1121) OAVG(14),OSD(14),OSDA(14),OAVG(21),OSD(21),OSDA(21),
     1OAVG(22),OSD(22),OSDA(22)
      WRITE(6,1122) OAVG(23),OSD(23),OSDA(23),OAVG(25),OSD(25),OSDA(25),
     1OAVG(24),OSD(24),OSDA(24)
      WRITE(6,1123) OAVG(5),OSD(5),OSDA(5),OAVG(19),OSD(19),OSDA(19),
     1OAVG(15),OSD(15),OSDA(15),OAVG(16),OSD(16),OSDA(16),
     2OAVG(17),OSD(17),OSDA(17),OAVG(18),OSD(18),OSDA(18)
      WRITE(6,1124) OAVG(12),OSD(12),OSDA(12),OAVG(13),OSD(13),OSDA(13),
     1OAVG(20),OSD(20),OSDA(20),OAVG(11),OSD(11),OSDA(11)
      WRITE(6,1125) OAVG(7),OSD(7),OSDA(7),OAVG(1),OSD(1),OSDA(1),
     1OAVG(8),OSD(8),OSDA(8),OAVG(2),OSD(2),OSDA(2)
      WRITE(6,1126) OAVG(6),OSD(6),OSDA(6),OAVG(3),OSD(3),OSDA(3)
 1100 FORMAT(/////////////////////////////////T28,'THE RESULTS OF ',I3,' DAY
     1S SIMULATION',/T37,'AFTER ',I3,' RUNS')
 1110 FORMAT(//T37,'ESTIMATED COST',//T70,'SD',
     1/T50,'AVE',T61,'SD',T68,'OF AVE',
     2//T15,'FLATCARS',
     3//T15,'COST PER CAR',T44,'$',         T45,F9.2,T55,F9.2,T66,F7.2,
     4/T15,'COST PER DAY', T44,'$',         T45,F9.2,T55,F9.2,T66,F7.2)
 1111 FORMAT(//T15,'INBOUND CONTAINERS',
     1//T15,'COST PER CONTAINER',      T44,'$',T45,F9.2,T55,F9.2,T66,F7.2,
     2/T15,'COST PER DAY',             T44,'$',T45,F9.2,T55,F9.2,T66,F7.2)
 1112 FORMAT(//T15,'OUTBOUND CONTAINERS',
     1//T15,'COST PER CONTAINER',   T44,'$',T45,F9.2,T55,F9.2,T66,F7.2,
     2/T15,'COST PER DAY',          T44,'$',T45,F9.2,T55,F9.2,T66,F7.2)
 1113 FORMAT(//T15,'SWITCHING AND CLASSIFICATION',
     1//T15,'INBOUND PER CAR',      T44,'$',T45,F9.2,T55,F9.2,T66,F7.2,
     2/T15,'OUTBOUND PER CAR',      T44,'$',T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'COST PER WORK DAY',     T44,'$',T45,F9.2,T55,F9.2,T66,F7.2,
     4/T15,'COST PER RUN',          T44,'$',T45,F9.2,T55,F9.2,T66,F7.2)
 1120 FORMAT(///T36,'SYSTEM VARIABLES',//T70,'SD',
     1/T50,'AVE',T61,'SD',T68,'OF AVE',
     2//T15,'UNITS IN THE SYSTEM',
```

```
     3/T20,'(PER DAY)',
     4//T15,'FLATCARS',            T45,F9.2,T55,F9.2,T66,F7.2,
     5/T15,'INBOUND CONTAINERS',   T45,F9.2,T55,F9.2,T66,F7.2,
     6/T15,'OUTBOUND CONTAINERS',  T45,F9.2,T55,F9.2,T66,F7.2)
1121 FORMAT(//T15,'TIME IN THE SYSTEM',
     1/T19,'(PER UNIT)',
     2//T15,'FLATCARS',            T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'INBOUND CONTAINERS',   T45,F9.2,T55,F9.2,T66,F7.2,
     4/T15,'OUTBOUND CONTAINERS',  T45,F9.2,T55,F9.2,T66,F7.2)
1122 FORMAT(//T15,'TOTAL UNITS IN SYSTEM'.
     1/T23,'(PER RUN)',
     2//T15,'FLATCARS',            T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'INBOUND CONTAINERS',   T45,F9.2,T55,F9.2,T66,F7.2,
     4/T15,'OUTBOUND CONTAINERS',  T45,F9.2,T55,F9.2,T66,F7.2)
1123 FORMAT(//T15,'SIDING OPERATIONS',
     1/T18,'(CONTAINERS)'.
     2//T15,'OFFLOADS PER WORK DAY',   T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'ONLOADS PER WORK DAY',     T45,F9.2,T55,F9.2,T66,F7.2,
     4//T15,'OFFLOAD TIME/UNIT (HOURS)',  T45,F9.2,T55,F9.2,T66,F7.2,
     5/T15,'ONLOAD TIME/UNIT (HOURS)',    T45,F9.2,T55,F9.2,T66,F7.2,
     6//T15,'TOTAL OFFLOADS PER RUN',     T45,F9.2,T55,F9.2,T66,F7.2,
     7/T15,'TOTAL ONLOADS PER RUN',       T45,F9.2,T55,F9.2,T66,F7.2)
1124 FORMAT(//T15,'SWITCHING TIMES',
     1/T19,'(HOURS)',
     2//T15,'INBOUND TIME PER CAR',    T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'OUTBOUND TIME PER CAR',    T45,F9.2,T55,F9.2,T66,F7.2,
     4//T15,'TOTAL TIME PER WORK DAY', T45,F9.2,T55,F9.2,T66,F7.2,
     5/T15,'TOTAL TIME PER RUN',       T45,F9.2,T55,F9.2,T66,F7.2)
1125 FORMAT(//T15,'CONTAINER PARKING',
     1/T19,'(PER DAY)',
     2//T15,'REGULAR LOT',             T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'OVERFLOW LOT (WHEN USED)', T45,F9.2,T55,F9.2,T66,F7.2,
     4/T15,'TOTAL PARKING',            T45,F9.2,T55,F9.2,T66,F7.2,
     5/T15,'DAYS OVERPARKING OCCURRED',T45,F9.2,T55,F9.2,T66,F7.2)
1126 FORMAT(//T15,'INBOUND YARD',
     2//T15,'CARS IN YARD',            T45,F9.2,T55,F9.2,T66,F7.2,
     3/T15,'MAXIMUM IN YARD',          T45,F9.2,T55,F9.2,T66,F7.2)
1160 RETURN
     END
```

The vita has been removed from the scanned document

A SIMULATION MODEL FOR THE ANALYSIS OF RAILWAY INTERMODAL TERMINAL OPERATIONS

by

Roy D. Hammesfahr

(ABSTRACT)

Intermodal traffic has been steadily increasing on the nation's railroads since the mid 1950's. Intermodal flatcar activity is now second only to coal in terms of total car loadings throughout the industry. The intermodal segment of the nation's transportation system is expected to play an ever increasing role in the future. Intermodal managers faced with increasing demands on their systems, have expressed a need for methods to aid in the development of new management techniques, economic costing models, and management information systems. The computer simulation intermodal model that is presented in this paper is designed to aid managers with the analysis of their current terminal systems and to plan for future growth in intermodal activity.

The intermodal terminal model employs discrete, next event, simulation techniques. The Q-GERT simulation language, developed by A. Allen B. Pritsker, provides the vehicle necessary to approximate the required activities and associated flow of transactions through the terminal system. Three specific types of containers and flatcars are provided for, in addition to provisions for over-the-road container

pick up and delivery. Thus, the model is adaptable to complex terminal systems, including sea ports where highly specialized containers are commonly encountered with rail, truck and ship interfaces. It is possible to simulate terminal activities for any period of time required for a specific analysis. The model's simulation output can also be modified, with little difficulty, to provide estimates of specific variables of interest for a particular terminal. Provisions for the operating environment of a terminal are also included in the model. These include week-end work rules, switching rules, container consignee notification rules, types of handling equipment employed and the standard working hours for a terminal.

The primary applications of the model are viewed to be in the areas of planning and analysis for intermodal terminal current operations and future design concepts. The graphical network orientation of the model, however, could provide managers with a communications tool to apprise upper level decision makers of new concepts. Current problems, with recommended solutions, could also be visually illustrated.