DEVELOPMENT OF A MULTIMODAL PORT FREIGHT TRANSPORTATION MODEL FOR ESTIMATING CONTAINER THROUGHPUT

A Thesis
Presented to
The Academic Faculty

by

Franklin E Gbologah

In Partial Fulfillment
of the Requirements for the Degree
MS in Civil Engineering in the
School of CIVIL AND ENVIRONMENTAL ENGINEERING

Georgia Institute of Technology
August 2010

DEVELOPMENT OF A MULTIMODAL PORT FREIGHT TRANSPORTATION MODEL FOR ESTIMATING CONTAINER THROUGHPUT

Approved by:

Dr. Michael Rodgers, Advisor
School of Civil & Environmental Eng.
Georgia Institute of Technology

Dr. Michael Hunter
School of Civil & Environmental Eng.
Georgia Institute of Technology

Dr. Randall Guensler
School of Civil & Environmental Eng.
Georgia Institute of Technology

Date Approved: July 6, 2010

To Pearl and Joana

ACKNOWLEDGEMENTS

First, I wish to thank God Almighty, who is the source of my academic abilities. Next, I wish to express my heartfelt gratitude to my advisor, Dr. Michael Rodgers, and the entire faculty of Transportation Systems Engineering at Georgia Tech for their support during my time of study. In addition, I am grateful for the contributions of Mr. Lakshmi Peesapati, a doctoral student in the Transportation Systems Engineering program at Georgia Tech. Finally, I wish to thank Joana, my wife, for her understanding and support which made it possible to undertake my postgraduate study away from my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Computer based simulation models have often been used to study the multimodal freight transportation system. But these studies have not been able to dynamically couple the various modes into one model; therefore, they are limited in their ability to inform on dynamic system level interactions. This research thesis is motivated by the need to dynamically couple the multimodal freight transportation system to operate at multiple spatial and temporal scales. It is part of a larger research program to develop a systems modeling framework applicable to freight transportation. This larger research program attempts to dynamically couple railroad, seaport, and highway freight transportation models. The focus of this thesis is the development of the coupled railroad and seaport models. A separate volume (Wall 2010) on the development of the highway model has been completed.

The model railroad and seaport was developed using Arena® simulation software and it comprises of the Ports of Savannah, GA, Charleston, NC, Jacksonville, FL, their adjacent CSX rail terminal, and connecting CSX railroads in the southeastern U.S. However, only the simulation outputs for the Port of Savannah are discussed in this paper. It should be mentioned that the modeled port layout is only conceptual; therefore, any inferences drawn from the model's outputs do not represent actual port performance.

The model was run for 26 continuous simulation days, generating 141 containership calls, 147 highway truck deliveries of containers, 900 trains, and a throughput of 28,738 containers at the Port of Savannah, GA. An analysis of each train's trajectory from origin to destination shows that trains spend between 24 – 67 percent of

their travel time idle on the tracks waiting for permission to move. Train parking demand analysis on the adjacent shunting area at the multimodal terminal seems to indicate that there aren't enough containers coming from the port because the demand is due to only trains waiting to load. The simulation also shows that on average it takes containerships calling at the Port of Savannah about 3.2 days to find an available dock to berth and unload containers. The observed mean turnaround time for containerships was 4.5 days.

This experiment also shows that container residence time within the port and adjacent multimodal rail terminal varies widely. Residence times within the port range from about 0.2 hours to 9 hours with a mean of 1 hour. The average residence time inside the rail terminal is about 20 minutes but observations varied from as little as 2 minutes to a high of 2.5 hours. In addition, about 85 percent of container residence time in the port is spent idle.

This research thesis demonstrates that it is possible to dynamically couple the different sub-models of the multimodal freight transportation system. However, there are challenges that need to be addressed by future research. The principal challenge is the development of a more efficient train movement algorithm that can incorporate the actual Direct Traffic Control (DTC) and / or Automatic Block Signal (ABS) track segmentation. Such an algorithm would likely improve the capacity estimates of the railroad network. In addition, future research should seek to reduce the high computational cost imposed by a discrete process modeling methodology and the adoption of single container resolution level for terminal operations. A methodology combining both discrete and continuous process modeling as proposed in this study could lessen computational costs and lower computer system requirements at a cost of some of the feedback capabilities of the model This tradeoff must be carefully examined.

CHAPTER 1

INTRODUCTION

The advent of the shipping container or intermodal terminal unit (ITU) has transformed the world in ways that were inconceivable in April 1956 when the maiden shipment was made aboard an old retrofitted oil tanker, the Ideal X, from Newark, NJ, to Houston, TX. As pointed out by economist Marc Levinson in his book *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger (Levinson 2006),* Mexican vacuum cleaners and Brazilian shoes can now be found in stores in Kansas, families in far away Japan can afford to eat beef produced in Wyoming and manufacturers no longer have to tolerate the high cost of urban plants just to be close to their suppliers. Likewise, many venerable shipping lines that could not transition from break-bulk cargo to containerized shipping have failed and once sleepy harbors like Busan and Seattle are now centers of global commerce. Mammoth containerships plying the world's oceans now allow once poor countries from one part of the globe (e.g. China) to become vital suppliers to richer countries elsewhere.

These changes have all resulted from the substantial drop in shipping costs brought about by the emergence of the ITU (Erie 2004 and Levinson 2006). Prior to its emergence, some freight forwarding companies had tried different forms of containers: the household goods movement industry had used steel containers (8' x 8' x 14') called Port-O-Vans to distribute household goods by rail throughout the U. S. (Strom 1972), and by the mid-1930s some railroads were offering container service in the form of "piggy-back" operations where highway dry goods semitrailers were transported on railroad flatcars. In addition, before the start of the Korean War the United States Army was already using special containers (8.5’ x 6.3’ x 6.8’) called “transporters” to send supplies to officers in the field. However, the use of these earlier containers was limited to specific companies or agencies until Malcolm McLean of Pan American (later renamed Sea-Land) Transportation introduced the ITU, on the 26th of April 1956. The ITU was designed to be an alternative to break-bulk cargo (cargo that is loaded and shipped as palletized units) shipping. Unlike its predecessors the ITU was specifically designed to enable quick loading and unloading onto ships and also quick intermodal exchange. It was an 8’ x 8’ x 10’ steel box made from approximately 1 inch thick corrugated steel. The design also incorporated a lock mechanism at its top four corners that enabled the ITU to be lifted by cranes. It was revolutionary and it completely outperformed break-bulk cargo shipping. For example, McLean’s Sea-Land Services (now Maersk Sealand) demonstrated that 40,000 tons of containerized cargo required only 750 man-hours to load and unload whereas the same quantity of break-bulk cargo required 24,000 man-hours (Strom 1972). These advantages aside, the transition from break-bulk cargo to containerized freight including standardization of the size and shape of freight containers to allow for fast mechanized handling of cargo at seaports and inland terminals was not made without opposition. Containerization had to overcome several attempts by organized labor leaders to stop its ascent as well as many years of contention among various industry players before final design standards were agreed upon (Levinson 2004). Now, virtually all global freight that can be containerized travels in this manner.

From 1995 to 2008, as world containerized freight traffic almost tripled in volume, containerized freight traffic volume in the U.S. more than doubled, growing at a rate of 6 percent annually (USDOT BTS 2009). Currently the U.S. accounts for about10 percent of global containerized freight traffic (BTS 2009) which is equivalent to 38 million TEUs[1] in 2008. Container seaports in the U.S. serve as points of entry or exit for most of this traffic. Traditionally, inland transportation of freight has been done by road-only transportation using long-haul highway trucks. However, multimodal transportation has grown to become a competitive means of freight shipment over the past few decades. This system of freight transportation involves the use of more than one mode during a single shipment; usually a rail and truck combination of long-haul rail transport and short distance trucking at both ends of the freight shipment. Long-haul trains using Class 1 railroads are often preferred to long-haul trucks for various reasons including reduced highway accidents rates, reduced highway traffic congestion, traffic pollution, and lower rail costs of shipment. As shown in Figure 1, since 1987 long-haul rail has had the largest fraction of annual freight ton-miles in the U.S.

From 1980 to 2007 multimodal traffic on railroads increased from 3 million units to more than 12 million units (USDOT BTS 2009). The high volume of containerized freight traffic that moves through the seaports, highways, and railroads underscores the impact it can have on the natural environment, local communities, and both state and national economies in the U.S. This impact should be expected to grow in significance as containerized international trade is expected to double between 2001 and 2020 (USDOT BTS 2009).

[1] TEU (Twenty-foot Equivalent Unit) is based on volume of the original twenty-feet-long standard cargo container although today forty foot containers are more common and standards exist for containers up to fifty-three feet in length. A single forty foot container is considered to be two TEU

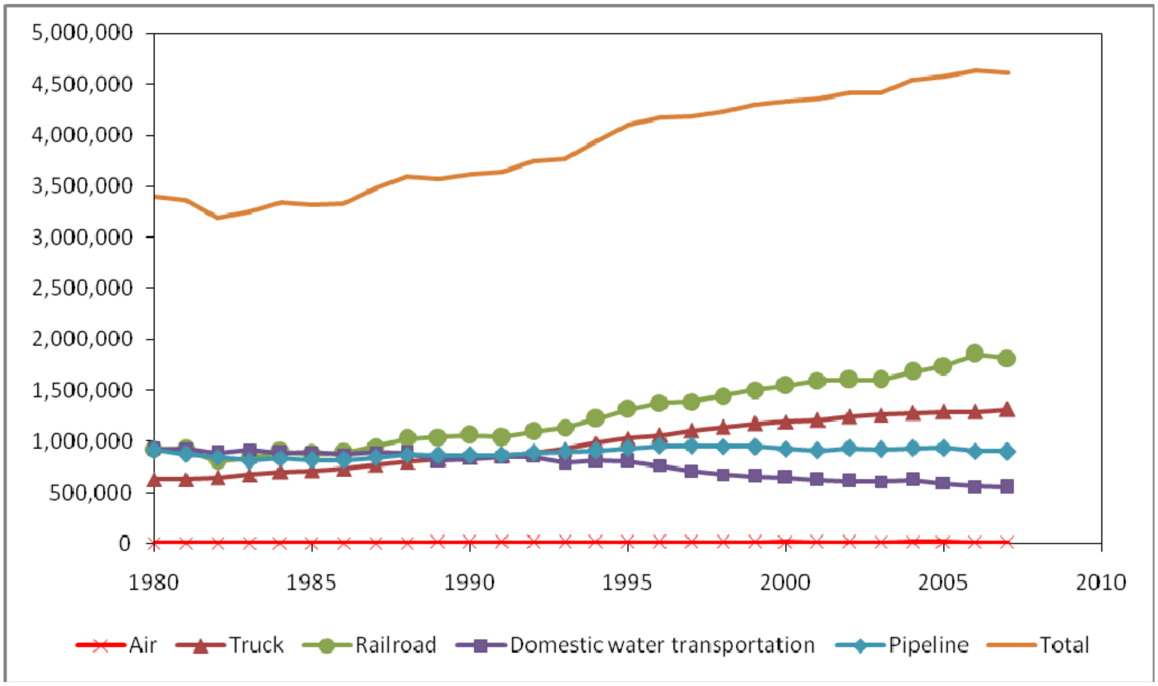

Figure 1 U.S. ton-miles of freight by transport mode. (USDOT BTS 2010)

Moreover, both container capacity and landside access of U.S. ports are lagging behind in terms of what is needed to efficiently handle this expected growth in container traffic. Dense urban development patterns around many ports are restricting infrastructure expansions for rail, road, and other facilities that serve these ports. In addition, the increasing capacity of containerships is shifting congestion from waterways to these rail and truck infrastructures (USDOT FHWA 2008). In a report to Congress on *The Performance of Ports and the Intermodal System* (USDOT MARAD 2005), it was estimated that the cost of landside access related congestion was as much as $200 billion, wasting 2.3 billion gallons of fuel and 3.7 billion man-hours annually. Further compounding these challenges is the fact that the multimodal freight rail industry is now experiencing demand that is greater than its capacity (USDOT BTS 2009).

Therefore, an efficient intermodal freight transportation system is essential to sustain the U.S. economic growth and dominance in international trade. There is a need for optimal cost strategies and decision support tools to efficiently handle container cargo, alleviate highway congestion around the seaports, improve landside access, and remove bottlenecks at facilities within intermodal terminals where truck and rail corridors connect to seaports. One such decision support tool that can be effectively used is a computer based simulation model. Many researchers have used computer based simulation to seek insights into many aspects of intermodal freight transportation. Simulation models have also become popular because they are able to overcome certain challenges that make it difficult to use analytical models (Dessouky and Leachman 1995) such as their inability to accurately capture blockage delays. However, these simulation models have not been developed at the system-level but rather they have focused mostly on isolated sub-systems to the neglect of the dynamic system-level interactions. The intermodal freight transportation system, especially in areas surrounding container seaports needs to be modeled as a complex dynamically coupled system of roads, connecting rails, and seaside activities all operating at multiple temporal scales and spatial references. The hypothesis is that such a model will be able to capture the impact of management policies on both the internal (terminal) and external (landside access and environment) operations of a container seaport terminal. Such a model may demonstrate complex and / or counter-intuitive behavior in addition to nonlinear responses which might not be observable with simulation models that are not dynamically coupled. Therefore, multimodal or intermodal system simulation models must dynamically couple port activities (where present), connecting rail network, and highway system over a spatial scale sufficient wide enough to capture effects of system interactions.

This research study is part of a larger research effort to develop a systems modeling framework for multimodal freight transportation. The larger research program attempts to dynamically couple the interactions between railroad, seaport, and highway truck operations at multimodal terminals. This research thesis contributes to the larger research program by developing dynamically coupled railroad, rail terminal, and seaport simulation models for container movement at three major multimodal container freight terminals in the southeastern U.S.; the Ports of Savannah, Charleston, and Jacksonville. The models of these three terminals cover portside activities, connecting CSX rail network, and adjacent multimodal rail terminal. The rail network developed is extensive and goes beyond the local network. It covers CSX railroads from Chicago, IL, through Erwin, TN, Spartanburg, SC, and Augusta, GA, CSX railroads from Washington, DC, through Harriet, NC, Florence, SC, and Charleston, SC, and CSX railroads from Miami, FL, through Jacksonville, FL. All these lines are connected to the Port of Savannah, GA. This study uses ARENA® to simulate the portside, rail network, and rail terminal operations. It should be stated that this study does not develop any new methodology or algorithm for the individual simulation models but rather focuses on the results when these are dynamically coupled. This thesis describes the developed model and shows its potential for better understanding the dynamics of these interactions. It should also be mentioned that as part of the larger research effort a separate work (Wall 2010) on the local highway system component has been completed. It uses VISSIM© traffic simulator to model the on-road components and communication between the Arena models and VISSIM model is done in “real time”.

CHAPTER 2

LITERATURE REVIEW

2.1 Scope of previous studies

The bulk of the literature on computer based simulations of multimodal freight transportation systems refers to studies carried out on scales that are usually not sufficiently wide to fully capture system level interactions, especially in areas around multimodal ports. These earlier studies have often simulated isolated areas of the system and range in scope from maritime terminal simulations (Asperen et al. 2003, Zaffalon et al. 1998, Mastrolilli et al. 1998) to rail network simulations (Dessouky and Leachman 1995, Lu et al. 2004, Dalal and Jensen 2001, Cheng 1998, Carrey 1994, Komaya 1991), and inland terminal simulations (Rizzoli et al. 2002, Gambardela et al. 2000). Other studies have also focused on projections for container shipments (Wilson and DeVuyst 2007) and container scheduling (Kozan and Preston 1999). A review of the literature was able to find one previous study (Rizzoli et al. 2002) that is set on a scale wide enough to capture some system level interactions. In this study, the simulation was developed as part of the PLATFORM project of the Directorate General VII of the European Community (now called the European Union). Even though their model attempts to integrate the various phases of transporting of an Intermodal Terminal Unit (ITU) these parts are not dynamically coupled and the researchers model the movement of the ITUs on the connecting rail network and truck arrivals at the gates as stochastic or deterministic arrival and departure times. This approach limits the models ability to report on system bottlenecks outside the terminal and the likely response to planned improvements.

2.2 Rail network modeling

Literature on the current state of the art for rail network modeling points to a preference of simulation models over analytical models. There are two likely reasons for this observation. First, some analytical modeling methodologies fail to capture operational constraints due to blockage delay because they model each rail track segment in the network as an independent queue (Dessouky and Leachman 1995). Second, "*analytical models that consists of a network of queues that have general distributions for arrival and service time are difficult to solve*" (Wolff 1991) (Dessouky and Leachman 1995). Previous studies using simulated rail models have been heavily focused on train movement algorithms and the major modeling requirements can be identified as follows:

a) trains going in / out of sidings

b) control logic over junctions, single, and multiple track segments

c) calculation of train movement time

d) avoidance of deadlocks.

Sidings are used as auxiliary tracks where trains can wait temporarily to be passed by other trains moving in the same direction or in an opposite direction. Sidings become necessary for unidirectional tracks when they are used by both low speed and high speed trains. For bidirectional tracks, sidings are necessary in areas of single track configuration. The length of the siding and trains need to be considered in movement algorithms because not all trains can stop at every siding. In between these sidings or passing areas different algorithms are needed to simulate train movement over segments of single-track and double-track rail lines. Most of the studies on single-track rail lines are dominated by analytical models (Frank 1966, Petersen 1974, Greenberg et al. 1988, and Chen and Harker 1990).

The model by Frank uses a sequential allocation of resources (e.g. tracks and sidings) and also assumes that only one train can occupy a single-track segment between sidings. This same assumption is still used in current simulation models for single-track lines. The disadvantage of this assumption is that it can unnecessarily increase the headway between trains and therefore reduce network capacity. In addition, blockage delays become more significant when there are only same direction requests for movement but trains have to queue up waiting for occupied single-track segments to become free. In an effort to overcome this disadvantage, a simulation study (Dessouky and Leachman 1995) allowed more than one train movement in the same direction to occupy a segment of single-track line between sidings which were assumed to have an infinite capacity. This was an assumption needed to avoid a network deadlock. However, an assumption of infinite capacity for sidings has not been popular; therefore, later single-track line simulation studies (Lu et al. 2004 and Fiorini and Botter 2005) still opt for the assumptions in Frank's model.

Train movement over double-track or multi-track segments is comparatively easier to model. The algorithm closely models car movement on a multi-lane freeway; trains traveling in the same direction use the same track. The main approach to double-track movement algorithm is to divide the line between sidings into track segments. The algorithm utilizes the basic idea of sequential resource allocation and the assumption of single occupancy of track segments, however, because the line between sidings can be divided into several track segments it has the advantage of reducing headways and increasing capacity. One of the earliest simulation studies (Petersen and Taylor 1982) using this approach, divided the line into track segments delineated by switches. This approached was extended further in a study (Dessouky and Leachman 1995) which divided the line into track segments of minimum length equal to the length of the longest train using the network. Hence, in their study the headway between trains was equal to one train length. However, a later research study (Lu et al. 2004) restricted the maximum length of a track segment to be the length of the longest train. The rational for this modification is that it can further reduce headways between trains even though it is likely to increase the computational run time because of the possibility of having more track segments in the model.

Review of the technical literature also shows that the level of detail in the calculation of movement time is influenced by whether train movement is over single-track segments or double-track segments. The large headways between trains on single-track lines greatly reduce the number of potential stops a train may make. Therefore, movement times over single-track segments have been modeled in detail and they are usually allocated in a deterministic way with an added random component (Dessouky and Leachman 1995) to account for reaction times for trains moving from stopped positions. However, on double-track segments the division of lines between sidings into multiple track segments and the associated close headways demands a more detailed modeling of movement time because the number of potential stops trains may have to make while waiting for a resource to become available increase greatly. In their study in 1995, Dessouky and Leachman calculated movement times based on whether a train stopped or did not to stop to wait for a resource and if it stopped whether the distance it needs to accelerate to attain maximum speed is shorter or longer than the length of the track segment. Thus, their movement time functions where dependent on:

a) the length of the track segment

b) maximum speed allowed on track segment

c) train’s acceleration rate and

d) reaction time for signaled trains to start moving.

Lu et al. (2004) extended this work on movement time algorithms to account for deceleration of trains when stopping. The study documents a very comprehensive algorithm to calculate move time which is capable of determining the fastest speed a train can travel while observing multiple speed limits on its route. It should be mentioned that whether the train movement is over a single-track segment or double-track segment, the total time is based on the time the rear of the train exits the track segment. Also, in both cases sidings, track segments, and junctions are modeled as resources which must be seized by a train before it can move over them. The algorithms reviewed also showed that waiting queues are normally prioritized on a first-in first-out (FIFO) basis; however, track ownership rights may supersede this priority. The occurrence of deadlock in simulated rail networks is high in single-track segments and multiple-track segments where there are more than two lines in the latter.

Networks consisting of double-track configurations are easy to simulate without deadlocks. Network deadlock occurs when there is more than one train moving in opposite directions requesting the use of a resource that is unavailable. And because trains do not usually move backwards, when such a conflict occurs in an area where passing is not possible a deadlock occurs. Train movement algorithms to avoid deadlocks have been extensively discussed in previous studies (Lu et al. 2004 and Fiorini and Botter 2005). Both studies basically propose the same one-step look-ahead algorithms in which trains have to ensure that the immediate next node (station or siding) where they can stop and the connecting track segment(s) between its current position and this node is free to seize before moving. However, the algorithm in the latter study has also been graphically presented to show how the three main conflicts that can cause deadlocks are avoided. Despite the extensive nature of the technical literature on train movement algorithms, this review finds that the simulation of mechanical defects and repairs has been rarely investigated. Apart from one study (Dessouky and Leachman 1995) which modeled the occurrence at an adjacent rail terminal of a multimodal terminal, actual simulation of this on rail networks were not found. One issue of concern is how to schedule trains for defects and repairs. Should trains in the simulation model be randomly selected or should individual trains be scheduled based on train-miles travelled or train-hours travelled? Another complication with simulating defects and repairs concerns how to handle defective trains. Specifically, should such trains hold on to seized resources until the exhaustion of a randomly distributed maintenance time or should they be delayed only at passing yards and thus free track resources for waiting trains?

2.3 Port terminal modeling

Also, during the literature review for this study two publications were identified (Ottjes et al. 1994 and Zaffalon et al. 1998) on a simulated containerized freight port terminal. Other studies on simulated port facilities have been tailored to specific facilities; a United Nations Committee on Trade and Development (UNCTAD) report in 1978 studied the design of jetties and applied simulation to analyze their capacities, Andrew et al. (1996) modeled crude oil lightering in Delaware Bay, Philips (1976) studied optimization models for a crude oil terminal, Baunach et al (1985) applied simulation to the study of a coal transshipment terminal, and Heyden et al. (1985) developed a decision support system for a grain terminal. The results of these studies have not been generalized (Asperen et al. 2003) and therefore may not be applicable to container port terminals. Ship delays at ports are very expensive in the real world and so simulation studies need to model the correct ship arrival process. This was the finding of a study (Asperen et al. 2003) on the impact of arrival processes on the efficiency of the loading and unloading process for ships. The study applied three arrival processes; stock-controlled, equidistant, and uncontrolled to ships docking at a chemical plant in the port of Rotterdam. Stock-controlled arrivals means that ships are scheduled such that a minimum base stock is maintained in the tanks, equidistant arrivals means that ships arrive at regular intervals spread throughout the year, and uncontrolled arrivals was modeled with a Poisson distribution. For container terminals, a stock-controlled arrival may not be applicable; however, the results showed that an equidistant arrival distribution may provide a more efficient loading and unloading process for ships than a Poisson arrival distribution. Multimodal freight terminals can be either a maritime or inland terminal. Inland terminals can be distinguished first of all by the fact that they are "inland and are nodes in a tightly interconnected network, composed of rail and road networks" (Rizzoli et al. 2002). They may also require different loading / unloading and yard operations because their ITUs can be a mixture of maritime containers, semi-trailers, and swap bodies. Maritime containers are stackable while semi-trailers and swap bodies are not [4]. However, they are similarities which facilitate the exchange of simulation modeling concepts between these two types of terminals. For example, both terminals have yard storage of ITUs, resource (e.g. gantry cranes and forklifts) sharing and allocation challenges, and gates where ITUs enter / depart terminal and their destinations are decided. Also, in both cases a terminal can be viewed as a network for routing ITUs to and from directly incompatible transportation systems.

Simulation modeling research in multimodal terminals has been dominated by inland terminals, with a focus on terminal performance. However, good terminal performance hinges on efficient storing of containers in the yard, scheduling of loading and unloading operations, and allocation of terminal resources. In one study (Rozzoli et al. 2002) the researchers identified the main modeling requirements as the loading / unloading of ITUs onto / from trains, storage of ITUs on the yard, and the arrival / departure of ITUs by truck. Both manual and automatic scheduling approaches have been used to model train and truck arrival at terminals. Manual approaches usually use a fixed timetable which can be developed from fixed travel times plus some stochastic delay value or from previously recorded train or truck events at the gates. On the other hand, the automatic approach uses algorithms to generate stochastic arrivals based on some statistical distribution. For example, in the simulation study (Rizzoli et al. 2002) of a terminal, the researchers used an algorithm to generate their truck arrivals from traffic rates obtained from an external road network model. The allocation of resources and scheduling of loading / unloading operations was the focus of a case study (Zaffalon et al. 1998) of the La Spezia Container Terminal (LSCT) located in the Tyrrhenian Sea in Italy. The researchers in this study developed three connected modules – a simulation model of the terminal, a forecasting model which predicts expected container traffic by analyzing historical data, and a planning model – to efficiently allocate resources and schedule loading / unloading operations. Their planning model uses a complex mixed-integer linear program to find the best combination of resources that minimizes costs, and then uses a Flexible Job Shop (FJS) model to prepare schedule lists for loading / unloading operations. These outputs are treated as management policies and provided as input data for the terminal simulation model. Their computer generated resource allocation and loading / unloading policies were able to achieve 30% savings on the net profit from terminal operations. Simulation models that integrate all phases of intermodal transportation over spatially wide scales can also serve as important tools in the socio-economic evaluation of alternative management strategies adopted by various actors in the freight distribution chain. European researchers have developed such an evaluation methodology (Gambardella et al. 2000) for the PLATFORM project. Their methodology uses a Cost Benefit Analysis (CBA) for commercial entities such as terminal managers, forwarders, and rail operators whose main goal is to increase profits while minimizing costs. However, the methodology uses a Multi Criteria Analysis (MCA) technique to evaluate decisions by actors in the policy arena whose concern is with the community welfare and other non monetary impacts.

Finally, the literature review for this study shows that discrete-event simulation is the most preferred modeling technique for the components of the intermodal transportations system. Also, models that integrate the different modes in the transportation system need to adopt the same level of resolution. The appropriate level of resolution is usually the single ITU because it allows more accurate modeling of a terminal’s inner workings although this resolution level comes at a high computation cost

(Rizzoli et al. 2002). There may be two reasons for this trend despite the likely computational costs, The preference of discrete-event simulation may be due to the fact that a continuous modeling approach especially for rail network simulations may “require the development of motion equations that represent train movement, which are difficult to represent mathematically in the context of train conflicts and deadlock-free movement” (Lu et al. 2004).

CHAPTER 3

DESCRIPTION OF MODEL

3.1 General

This section presents a description of the rail network and maritime terminal models which were constructed using ARENA® simulation software. ARENA's in-built modules allowed these models to be developed to a detail that closely mimics world behavior. The terminal layouts of the three ports are conceptual but the modeled rail network closely follows the CSX railroad originating from Chicago, Washington, DC, and Miami into the southern U.S. Ports of Savannah, Jacksonville, and Charleston in terms of track lengths, location of sidings, and speed limits. Figure 2 and Figure 3 shows the developed rail network and a typical port layout in the model, respectively.

The models were constructed using a nested modeling approach with nodes and links. There are two high-level nodes; the port module and the rail network module. Within these high-level modules are nested sub modules such as dockside operations, customs, inspections, bonded storage, and intermediate stations / sidings. Dynamic module coupling is handled by a top-level framework and individual modules may vary in sophistication depending on overall needs. For example, service times may be a specified value or a variable. The links represent individual transportation facility links between either the nested sub modules or the high-level modules. Figure 4 shows a graphical representation of the modeling coupling as constructed.

Figure 2 A map showing the modeled CSX railroad routes.

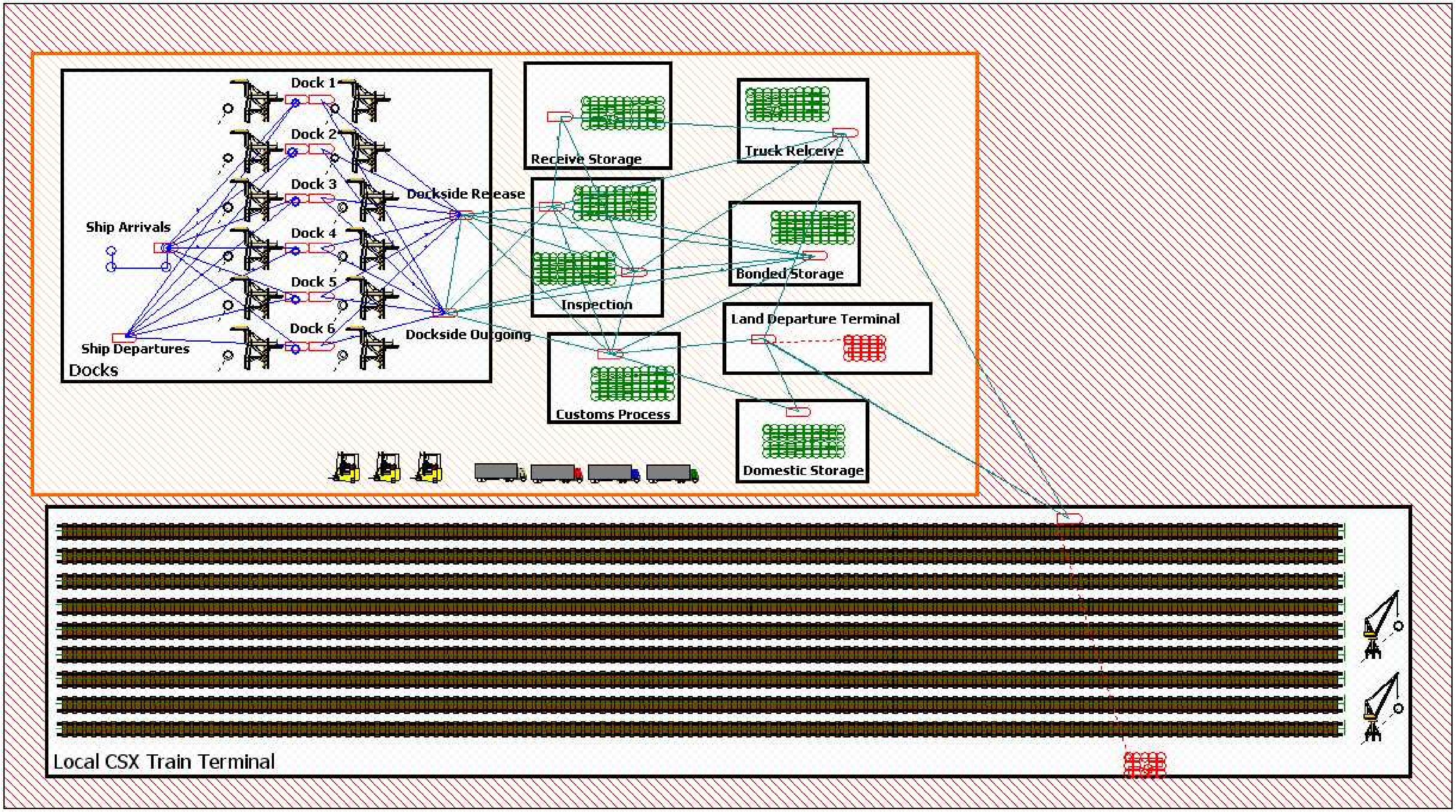

Figure 3 Layout of a typical port terminal showing transport connections between various areas as links.

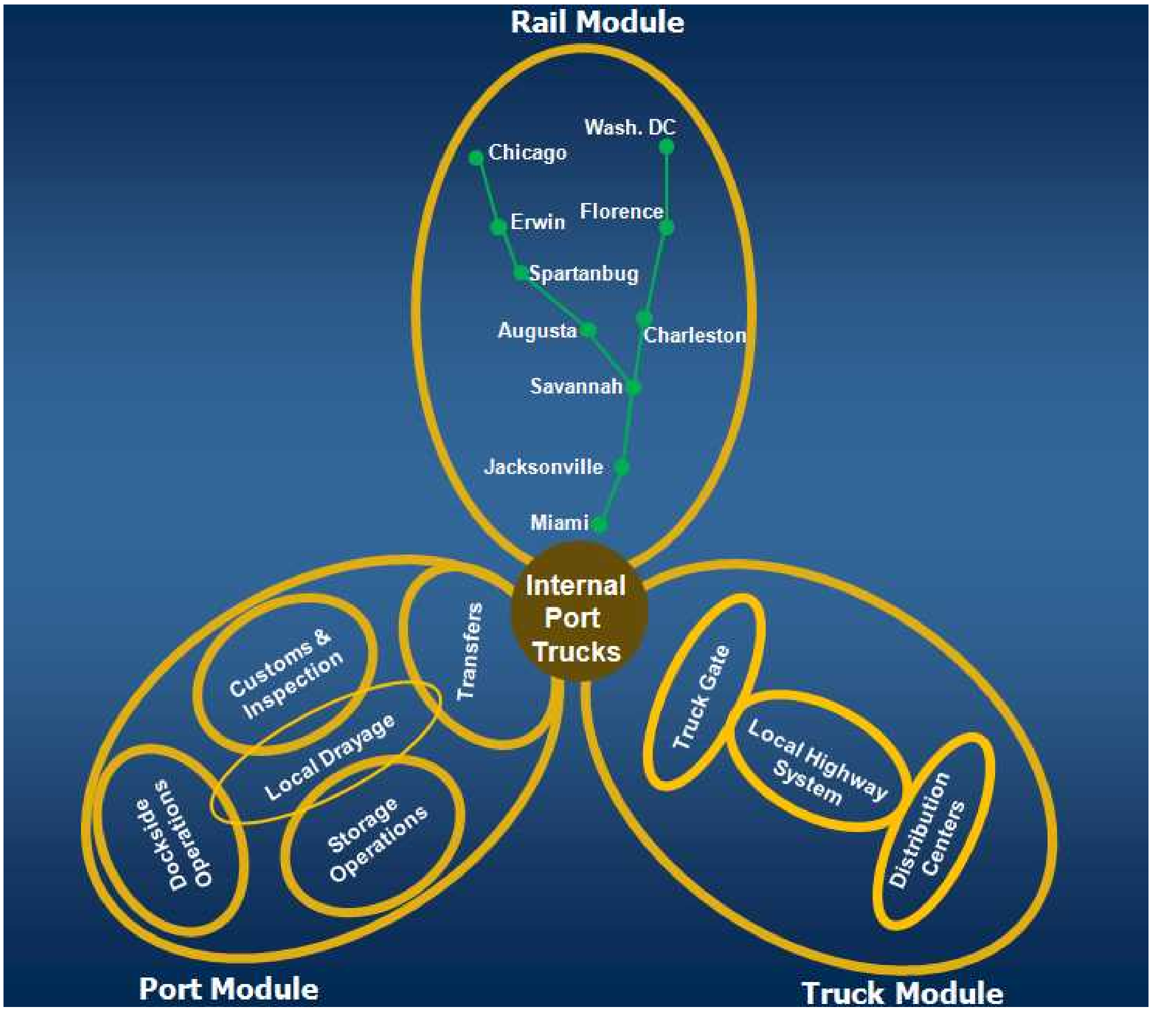

Figure 4 Graphical representation of the dynamic model coupling.

3.2 Rail network model

This section provides a general description of the logic in the various sub-models. For each sub-model representation shown as an Arena® flowchart diagram, a more detailed, stepwise description has been included in Appendix A. The alphanumeric codes on the flowcharts in the figures identify the steps in the logic for a typical train.

3.2.1 Train creation, dispatch, movement, and disposal

The rail network model utilizes a central *train creation and initial dispatch logic* that usually creates trains every hour and randomly dispatches them from six terminals (Erwin, Augusta, Rocky Mountain, Miami, Washington, DC, and Chicago) to 12 receiving terminals according to an exponential probability distribution. Figure 5 shows a schematic representation of the train creation, dispatch and movement logic. The trains are assigned a *Direction ID* and *Route ID* which is used to identify all the track segments and stations / sidings en route to their destinations from an itinerary list. This itinerary list must be provided as an input database file. Apart from the above mentioned attributes, trains are also assigned an *Origin ID*, *Destination ID*, and *Time Stamp* at creation and departure from the terminal. A flowchart representation of this logic is presented in Figure 6.

Trains are also created and dispatched from the adjacent CSX rail terminal at the three ports when all containers assigned to a particular destination and train has been loaded. There are eight destination options from Savannah, three options from Jacksonville, and one from Charleston. For both Savannah and Jacksonville, probability distributions are used to assign the destinations of created trains. Trains created at ports are also assigned the same initial attributes mentioned above. A typical train creation and dispatch logic from a port is shown in Figure 7.

The rail network model also has a central *train movement logic* used to route trains between intermediate stations / sidings. The flowchart representation of this logic is shown in Figure 8. As can be inferred from the flowchart, there are two main branches of this internal logic. The lower branch routes trains whose next station in the route sequence is not the assigned destination while the upper branch routes trains which are one station away from the assigned destination. When a train arrives at an intermediate station on its route, it enters this logic where it releases the previously seized track segment for any other waiting train. If the train is more than one station away from its destination the logic checks whether the next step would require a direction change. This is a very important step because a change in direction may mean that the sequential routing station / siding and track segment identification numbers in the database may change from an ascending order to a descending order. The next step in the logic assigns the train the next track segment and siding / station. However, before it can proceed with its journey it must examine the model to ensure that the following conditions are met:

a) There is an available track segment between its current station / siding and the next one and that no other train has been assigned to the track segment

b) The next siding / station where it can stop has an unoccupied passing area.

If these conditions are met the train seizes these resources and proceeds to move otherwise it waits in a siding until the condition can be met. The priority scheme for trains waiting in a siding for a particular resource to become available is first-in-first-out (FIFO). This logic ensures that at least one track segment in a multi track configuration is reserved for an opposite direction of travel.

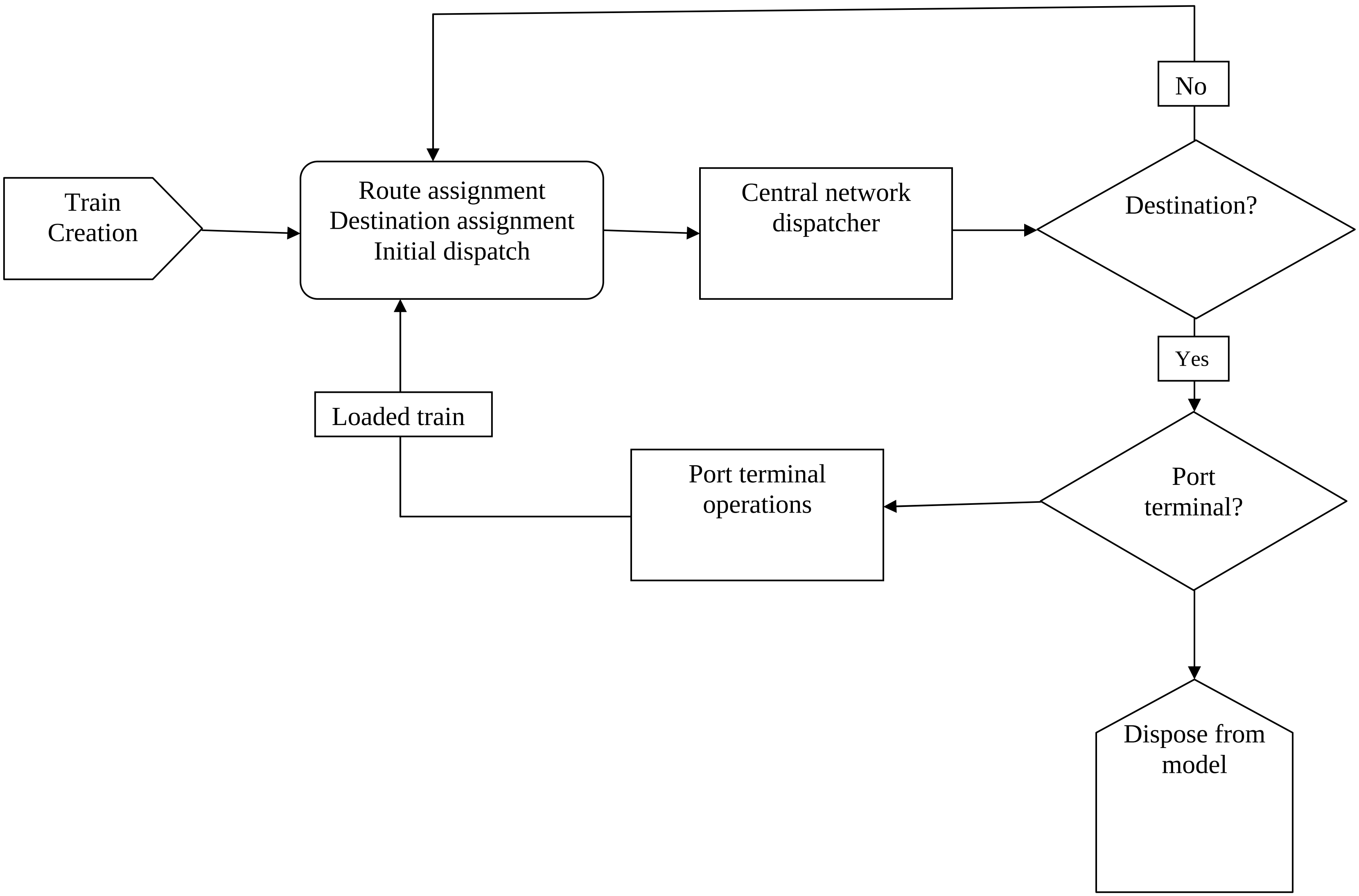

Figure 5 A schematic representation of the train creation, dispatch and movement logic

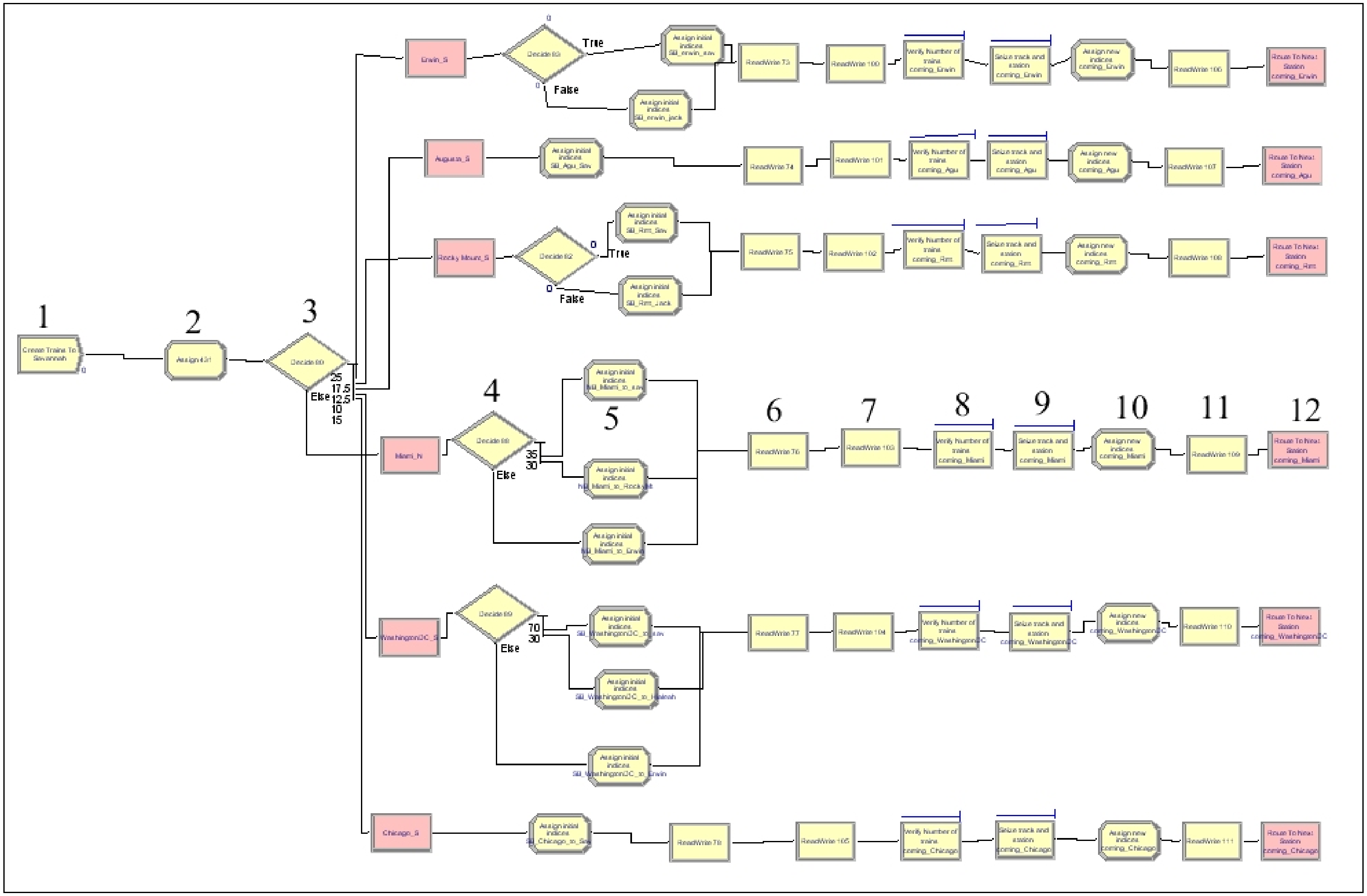

Figure 6 A graphical representation of the train creation and dispatch logic from inland terminals using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

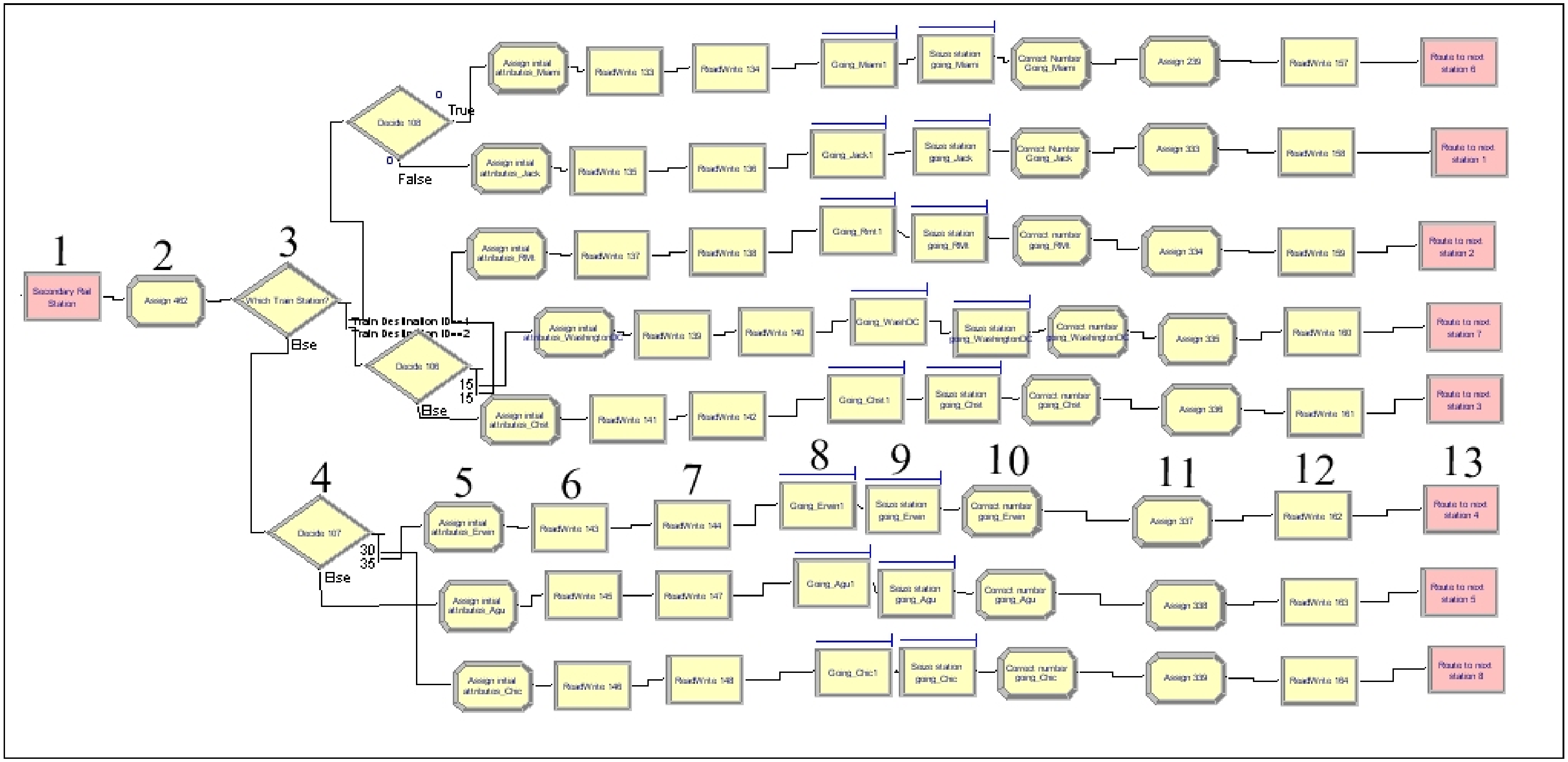

Figure 7 A graphical representation of the train creation and dispatch logic from a typical port terminal using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

All station / siding resources held by a train are released just before it exits the siding / station. If a train arriving at this logic is one station away from its destination it is assigned the next track segment and the model is checked to ensure that the connecting track to the destination is free for it to seize and move, otherwise it waits until it can seize the resource. Trains with an assigned destination that is other than a port are disposed from the model after it is given a time stamp attribute. However, if the train is headed for a port terminal it is sent to the *rail terminal logic*. Figure 9 shows the flow chart representation of the *disposal logic*. In addition, every time a train arrives at this logic it is time-stamped on entry and exit. The *Station ID* and *Train ID* are also written to the database. The *train movement logic* does not yet incorporate the more efficient movement logic for double tracks that has been identified from the review of existing literature. This is because of the computational costs of the adopted discrete process modeling approach. The number of entities created at runtime places significant demands on computer system resources and the extra track segments that need to be created in accordance with the identified double track movement logic adversely affect model performance. Hence, track segment lengths used in the model are the actual physical distances between sidings and / or stations. Each track segment also has an imposed speed limit corresponding to the existing authorized speed. In contrast to previous studies mentioned in literature review section, this study has adopted a rather simple approach to the estimation of train movement time. The hypothesis is that the net effect of all possible combinations of train accelerations, decelerations, and constant speed over track segments could be approximated as average speeds which can be modeled as a bounded range of randomly distributed speeds.

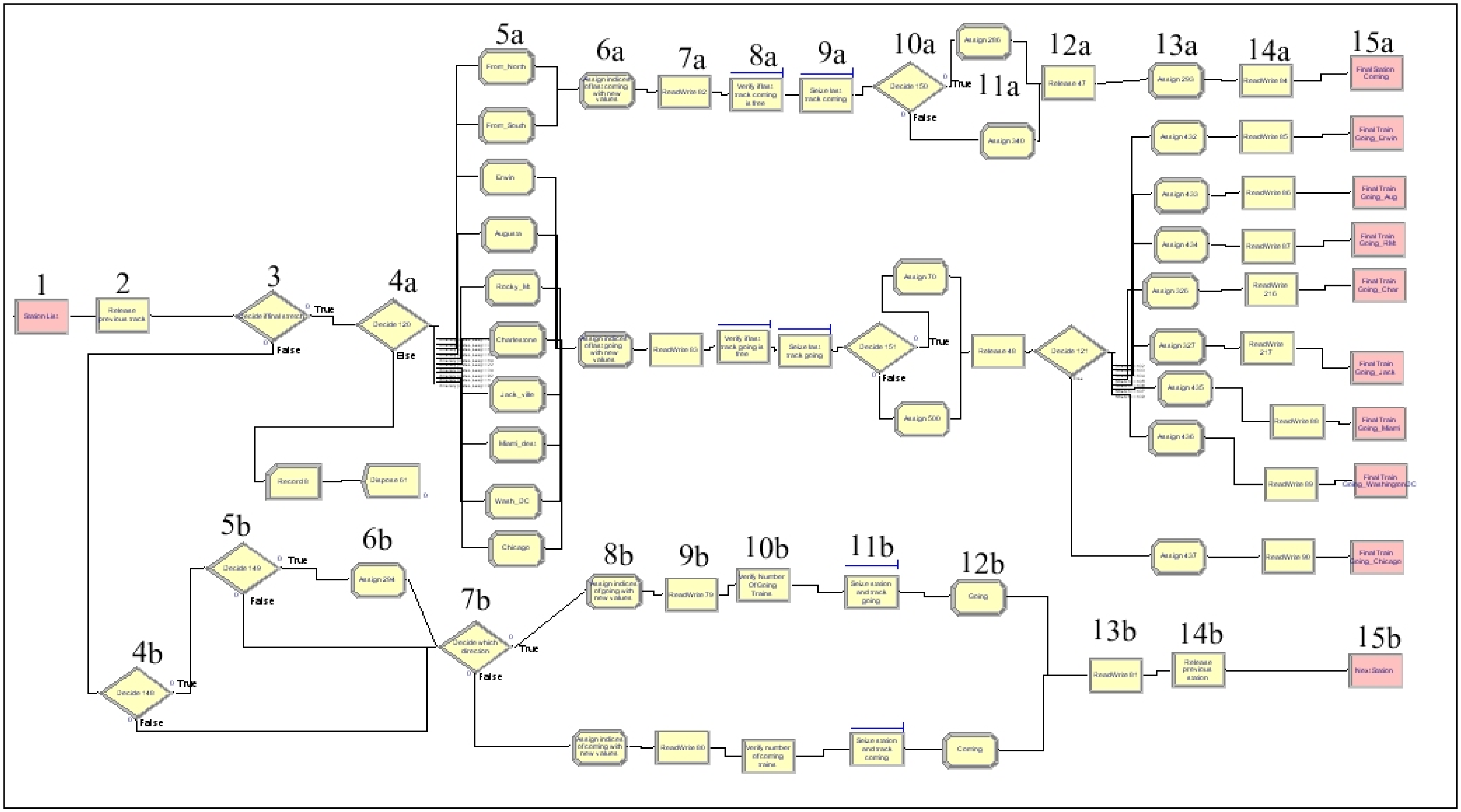

Figure 8 A graphical representation of the train movement algorithm using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

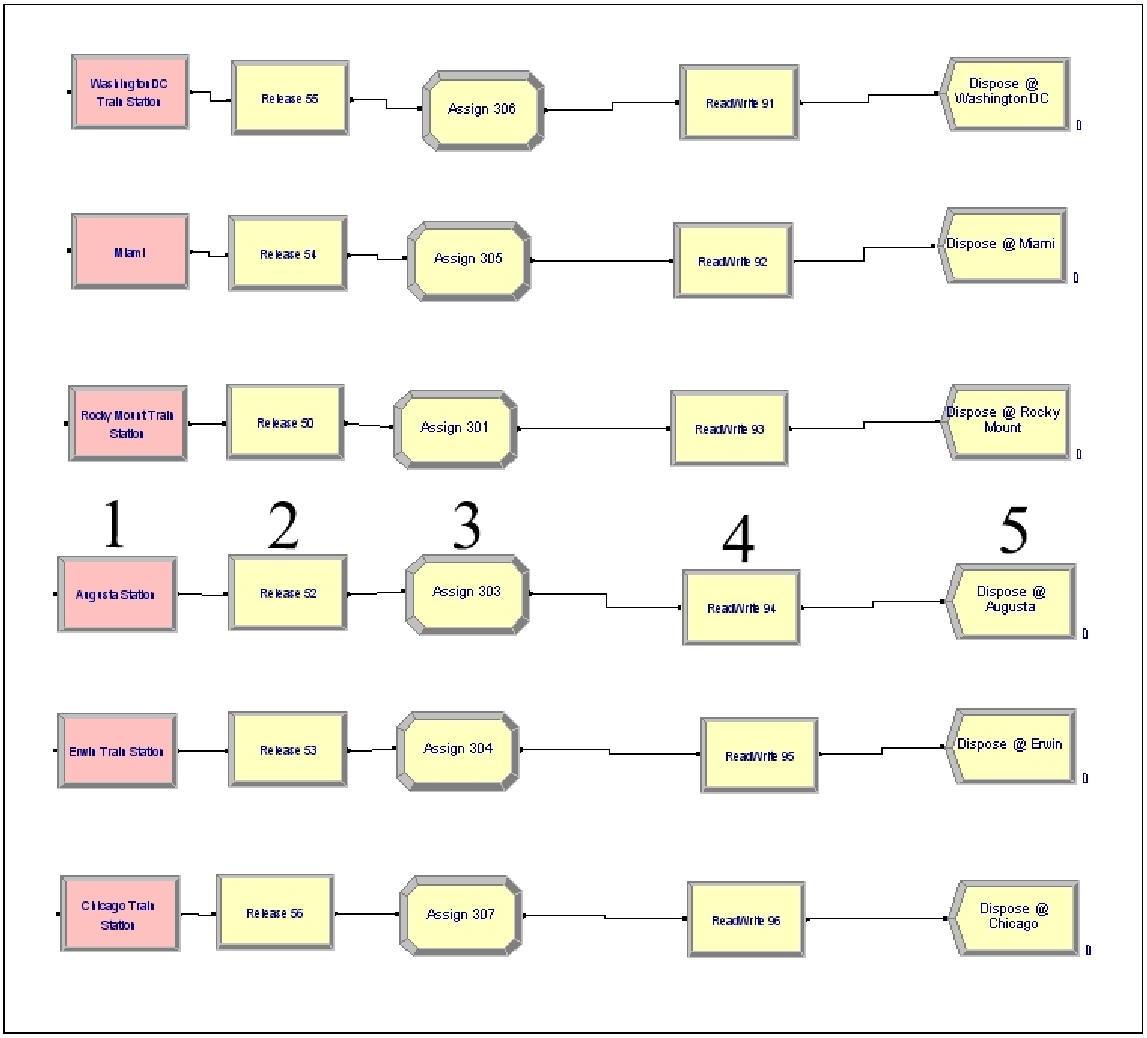

Figure 9 Train disposal logic used at the inland destination terminals. (Numbers reference step-by-step descriptions in Appendix A)

Assuming a range from the maximum authorized speed to 50 % less of this speed and a uniform probability distribution, average speeds can be estimated with the time function given below:

$$Time(min) = \frac{60 * track\ segment\ length(miles)}{Unif(0.5,1.0) * speed\ limit(mph)}$$

In instances of model calibration and validation the bounds on the range, especially the lower bound will have to be adjusted to change network capacity.

3.2.2 Trains at adjacent rail terminals

At any of the CSX rail terminals at the modeled ports a train is first made to release the previously held track segment and before it can proceed to a shunting area where it waits for an available unloading track within the rail terminal. The shunting area has been modeled to have an infinite capacity. Modeling the shunting area with an infinite capacity enables the model to estimate a ‘theoretical’ maximum number of trains waiting behind the rail gates at any time. This modeling approach is more likely to give better estimates of the unconstrained demand on the adjacent rail terminal. When a train departs the shunting area for the rail terminal inside the port area, it goes through the rail gates where the following decisions are made about it and its load:

a) The number of containers to unload

b) The destinations of unloaded containers

c) Container identification number

Container destinations can be one of three local distributer truck centers, outgoing dockside, or long haul rail to another terminal. Inside the terminal two fixed delay times for parking and switching are applied to the train before unloading of containers can start. All the containers to be unloaded are made to queue for an available unloading crane. Crane service times (in minutes) are modeled to fit a triangular probability distribution with a minimum of 2.5, mode of 3, and maximum of 3.5. Containers with a truck center or outgoing dockside destination assignment must request for a distributer truck or an internal port area transport truck respectively before seizing an unloading crane. The unloading crane places these containers on the trucks which are then dispatched either to the port area or to the truck gate and a local distributor truck terminal.

As previously mentioned the trucking facilities have been modeled using VISSIM© traffic simulation software in a separate research study (Wall 2010). A seamless communication between VISSIM© and ARENA® helps to transfer trucks between the two models in real time. Subject to the availability of waiting trains headed to their destination, containers that are assigned to long haul train routing are either immediately loaded onto trains or held in temporary storage. Trains leaving the terminal are made to release previously seized loading tracks before proceeding to the internal logic for dispatching created trains at the port. A flow chart representation of this logic is shown in Figure 10.

3.3 Portside model

The internal logic described in this section is the same for any of the three ports developed in this study. As mentioned earlier the designs of the port terminals are conceptual and may not accurately match the existing situation at any of these ports. In addition, more detailed description of the logic in the various sub-models has been included in Appendix A.

The Receiving Dockside Operations logic models ship arrivals at a port according an equidistant / constant distribution. The inter-arrival times represent the total vessel calls at the ports in 2007. For example, in 2007 there were 1807 containership calls at the Port of Savannah (USDOT BTS 2009), this represents an equidistant distribution of about a ship every 4.8 hours. Created ships are immediately assigned both the number containers to load and unload. The numbers are randomly assigned according to uniform probability distribution within a predefined range.

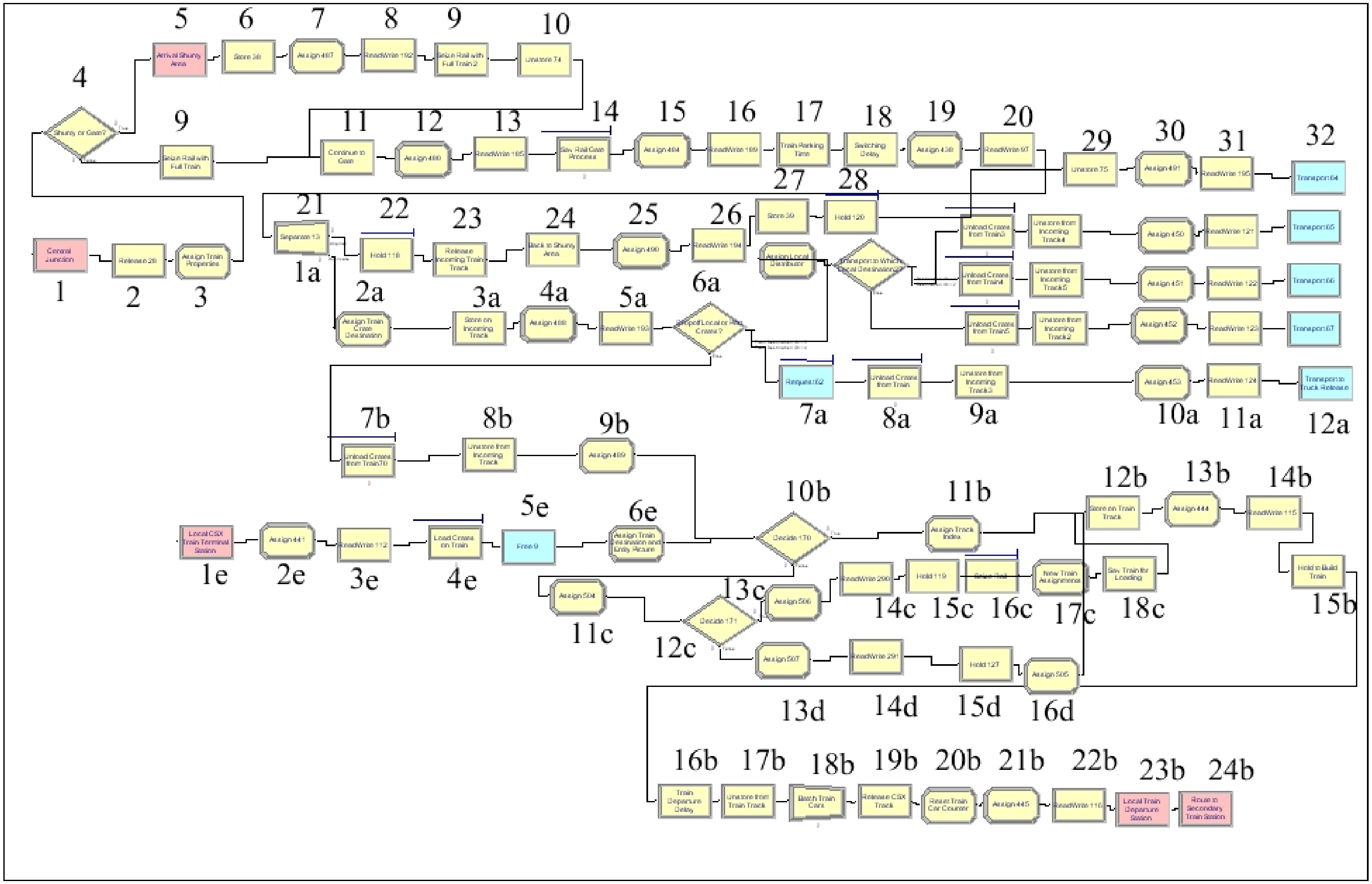

Figure 10 A graphical representation of the processing logic for an adjacent CSX rail terminal at a port using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

A terminal has six docks (jetties or quays) and created ships wait in a FIFO prioritized queue to seize one when it becomes available. When a dock becomes available the ship at the top of the queue seizes it before being routed to it after a fixed routing delay. The model logic then assigns an initial destination to each of the containers to be unloaded. The possible destination options include the adjacent rail terminal, long distance truck terminal, and one of three local truck distribution terminals. As previously explained for the rail network model, all the truck facilities are external to the port area and have been simulated with VISSIM© traffic simulation software. The containers wait in their dock specific queues for the chance to seize one of two cranes dedicated to each of the six docks. The crane service times are randomly assigned according to a triangular probability distribution. The minimum possible service time is 2 minutes, the maximum is 3 minutes, and the most likely is 2.5 minutes. Unloaded containers are temporarily held on the dock while waiting for a requested forklift to transport them to the inspection area of the port. These requests are given the highest priority and forklifts are assigned according to the smallest distance rule. This rule is made possible because the port terminal is designed as a network for routing containers from one location to the other. The database has a network file containing distances between various locations and the speed of internal port area transporters such as forklifts and trucks. Therefore, travel time for these transporters is usually derived at runtime based on only distance and speed. However, it is also possibly to model the travel time to include network delays which depend completely on operational characteristics. The appropriateness of these two routing approaches is a programming choice, (i.e., whether transporters in the terminal move as *free paths* or as *guided transporters).* An additional fixed delay is also applied to simulate forklift pickup time. A sample flow chart representing this logic is shown in Figure 11.

Within the Inspection area an extra fixed delay is applied to simulate forklift drop off time. The Inspection area is the first point-of-call in the port for containers en route the ships. Containers must first release their transporters before seizing a set of inspection resources according to a FIFO prioritized queue. An inspection delay modeled as a random triangular probability distribution is then applied after which the seized set of inspection resources is released. A random selection of containers is immediately subjected to further and more rigorous inspection. This second inspection delay has been made larger in order to simulate the more thorough nature of the inspection. A predetermined percentage of containers fail this second inspection and they are disposed from the model. Containers that successfully pass the inspections are either sent to Customs or Bonded Storage. Containers whose custom payment isn’t made on time are sent to bonded storage and this is simulated using a predetermined percentage of the containers. However, before being routed to either of these locations the containers must request for forklifts. All forklift requests in the model have high priority and apply fixed delays to simulate pickup time and drop off time. Figure 12 shows a flowchart representation of the internal logic in the inspections module. The *bonded storage logic* first releases the transporting forklift and then applies a random storage delay that is uniformly distributed within the ranges of 2 to 8 hours. After this delay is exhausted containers request forklifts to transport them to Customs. The flowchart shown in Figure 13 shows the internal logic in the bonded storage module.

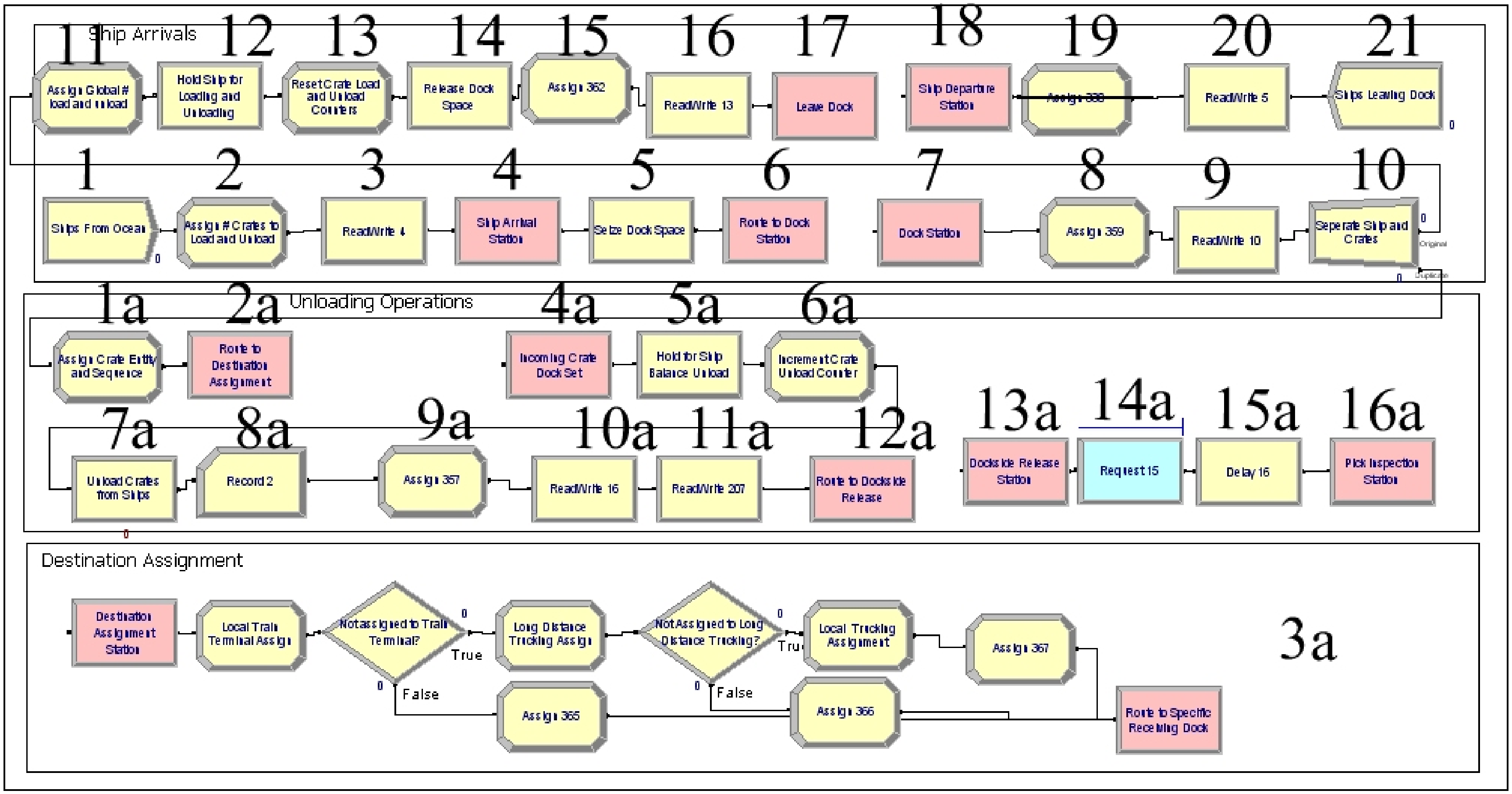

Figure 11 A graphical representation of the Receiving Dockside Operations logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

When containers arrive at customs, they first release their seized forklift and then join a FIFO queue to wait until they can obtain a customs agent. Service times at the customs section is randomly distributed according to a triangular probability distribution. Minimum service time is 3 minutes, maximum service time is 7 minutes, and the most likely service time is 5 minutes. At the end of the service time, a container may be routed to the domestic storage module, land departure module, or dockside outgoing module. Containers assigned for ship departure proceed to dockside outgoing and those assigned to land departure are sent into domestic storage if the temporary storage available at the land departure terminal is full. All containers have to request a forklift to transport them out of Customs to any of these locations. Figure 14 shows a flowchart diagram of the Customs module. The internal logic of the domestic storage module is similar to that of the bonded storage module described previously. The only exception is that residence time is determined by availability of temporary storage capacity at the land departure terminal rather than a random probability distribution. Figure 15 gives a flowchart representation of this logic.

Containers arriving at the land departure module are initially stored in available temporary storage. The internal logic here distinguishes containers by their destination ID’s and therefore processes them in three possible ways represented as the three main branches of the flowchart shown in Figure 16. The top branch processes containers assigned to the local rail terminal. These containers first request an internal port transport truck that will take them to the rail terminal. The request is a high priority request and it is based on the smallest distance rule. It then seizes the next available flatbed crane to load it onto the truck and remove it from temporary storage before being routed to the rail terminal. The service time for the flatbed crane is the same as the cranes at the rail terminal. The middle branch handles all containers assigned to local distributor truck terminals. There are three local distributor truck terminals and as such this branch is further split into three separate branches but they use basically the same logic. Containers that come to this logic are designated a local distributor center id before they request for an external truck from the VISSIM© model to transport them. Upon receiving the truck, the container seizes a flatbed crane, it is removed from the temporary storage, and it is routed back to designated local distribution center via VISSIM©. The lower branch is similar to the middle branch except that it is for processing long distance distributions. When containers are sent from land departure to the rail terminal, they first have to seize one of the available loading cranes at the terminal and then release the internal port truck. Then subject to the availability of a waiting train headed for its direction it may be immediately loaded on the train or held temporarily.

As previously mentioned in the description of the rail terminal logic, some containers that arrived by train are sent to the outgoing dock to leave by ship. These containers are initially transported to a truck receive terminal in the port by an internal port truck. At the truck receive terminal, it joins other containers arriving by trucks from the VISSIM© model and they are all processed and assigned a dock id. They then have to seize a flatbed crane to unload it from the truck and release the internal port truck. The internal logic at this module then scans the model to see if any dock as been assigned a number of containers equal to or greater than the number of containers to be loaded onto the ship currently docked there. If such a dock is found then all subsequent containers assigned to that dock id are sent into a temporary storage. Otherwise the containers are routed to inspections. In both cases containers must request for forklifts to transport them. The next series of modules for these containers have already been described except for the dockside outgoing module. In Figure 17 we have shown the flowchart representation of the logic at the truck receive module.

When these containers arrive finally at the outgoing dockside module they release their forklifts and are routed to the specific outgoing dock. A fixed routing delay is applied to them as a consequence. They then have to seize one of two cranes dedicated to the specific dock. When ship loading operations are done for the specific ship docked, the ship is signaled to leave. After a fixed routing delay the ship releases the dock and is disposed from the model. A flowchart representation of this internal logic is shown in Figure 18. It should be mentioned that in every internal logic module, containers are assigned attributes of time on entry / exit and a location within the port.

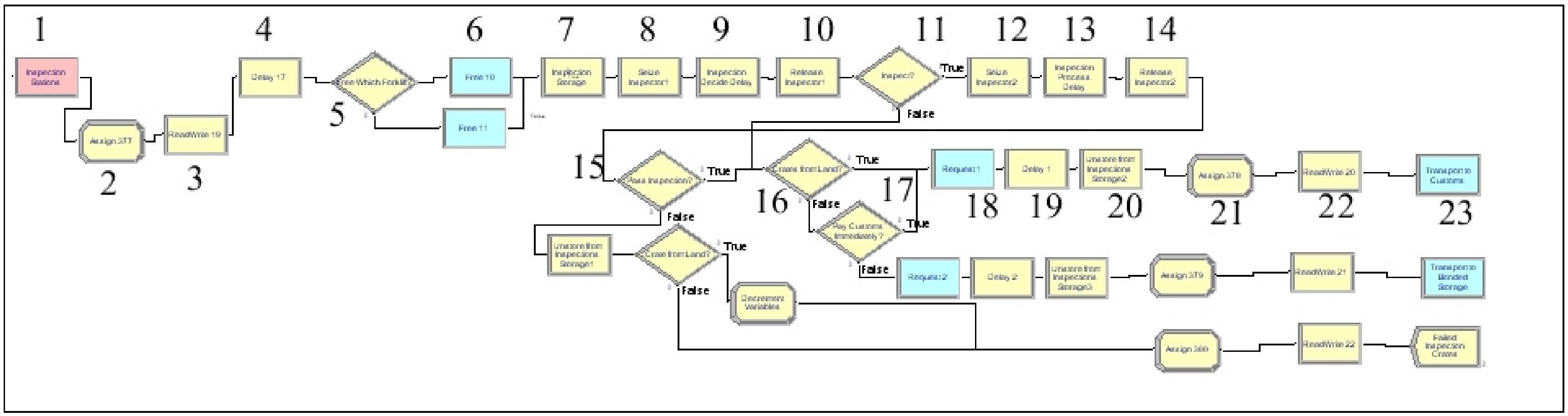

Figure 12 A graphical representation of a typical port's Inspections logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

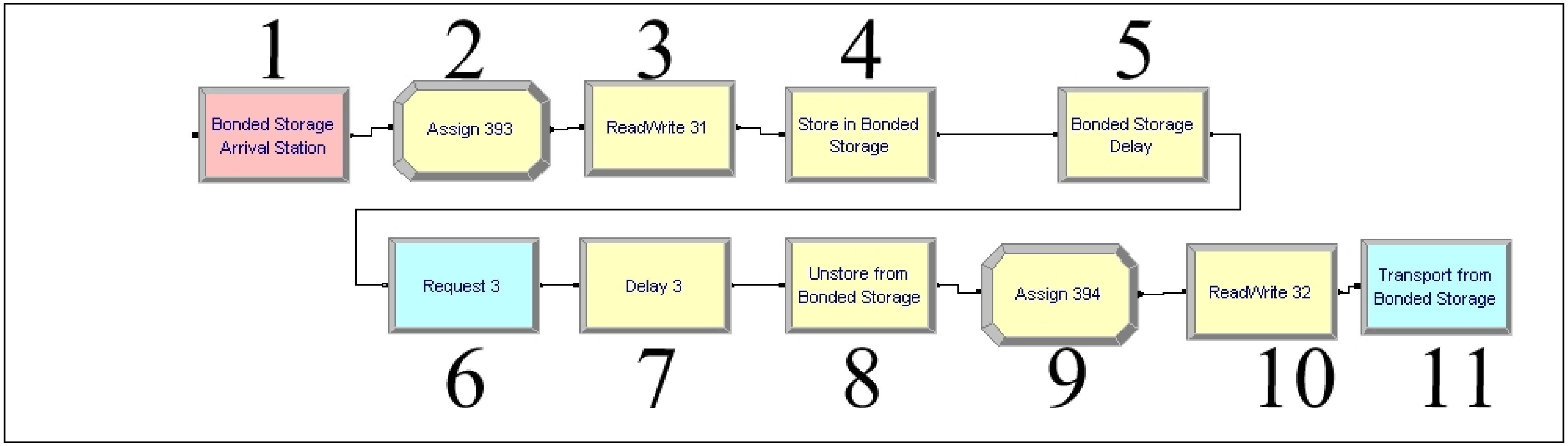

Figure 13 A graphical representation of a typical port's Bonded Storage logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

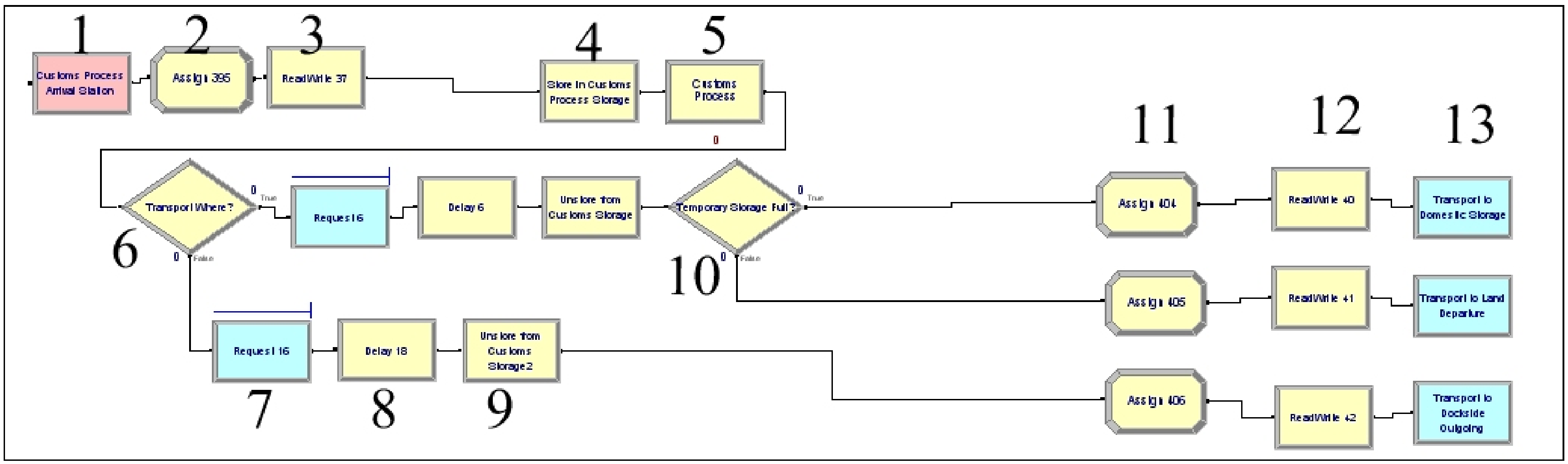

Figure 14 A graphical representation of a typical port's Customs process logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

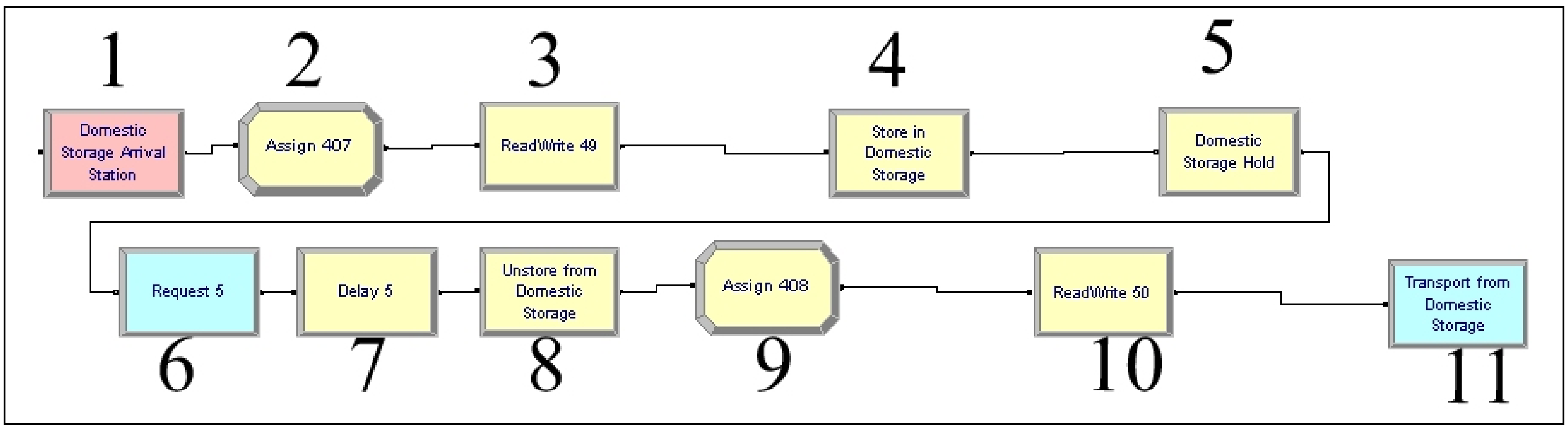

Figure 15 A graphical representation of a typical port's Domestic Storage logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

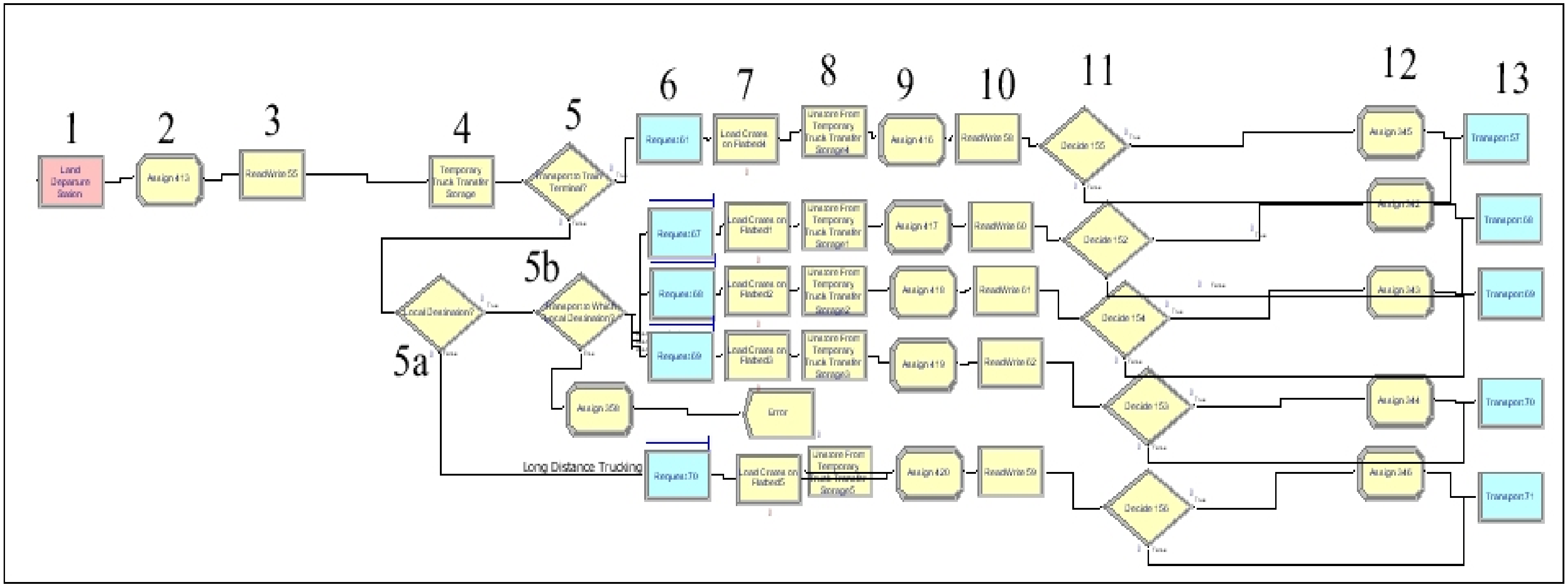

Figure 16 A graphical representation of the Land Departure Terminal logic using Arena’s inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

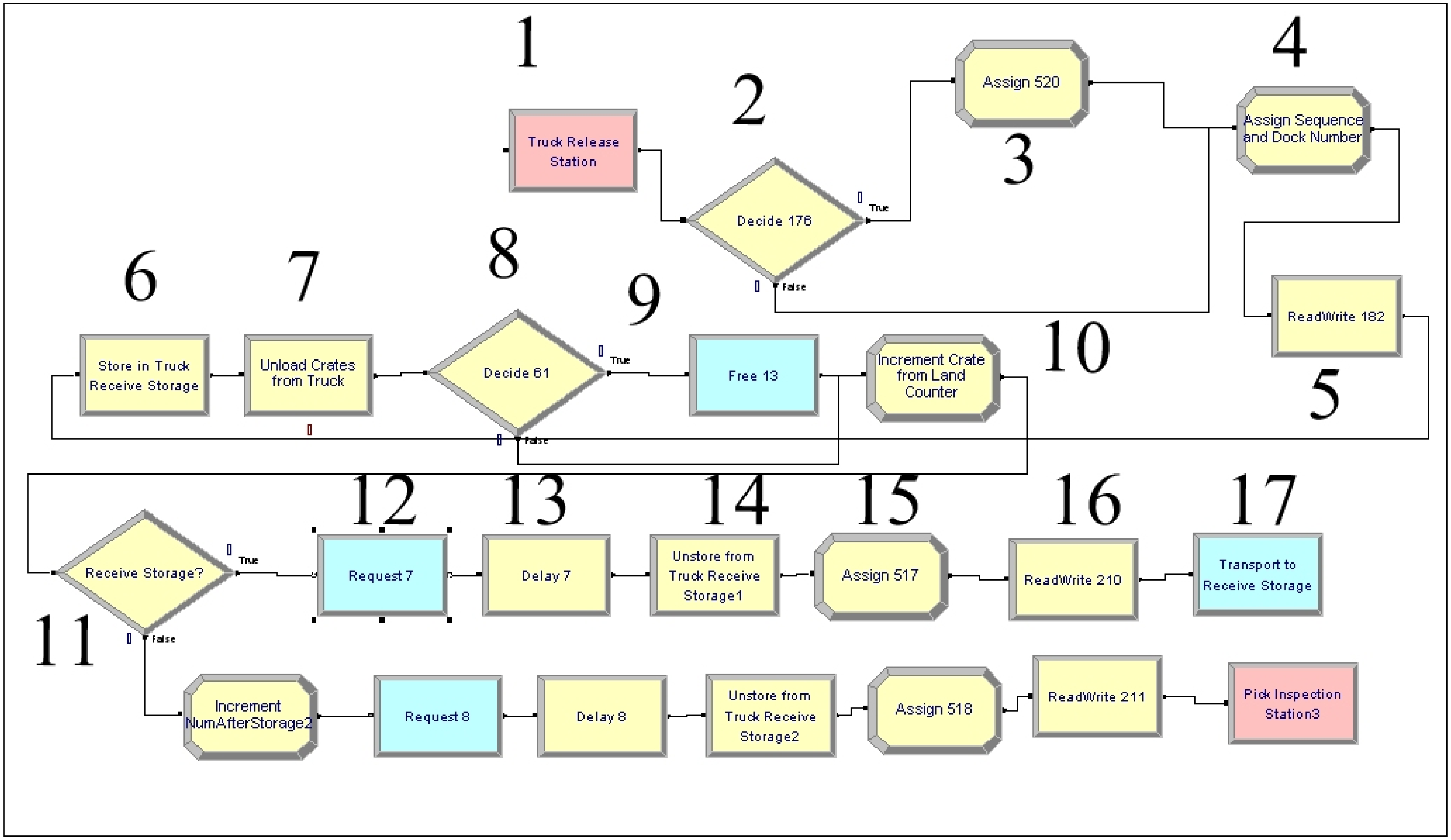

Figure 17 A graphical representation of the Truck Receive Terminal logic using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

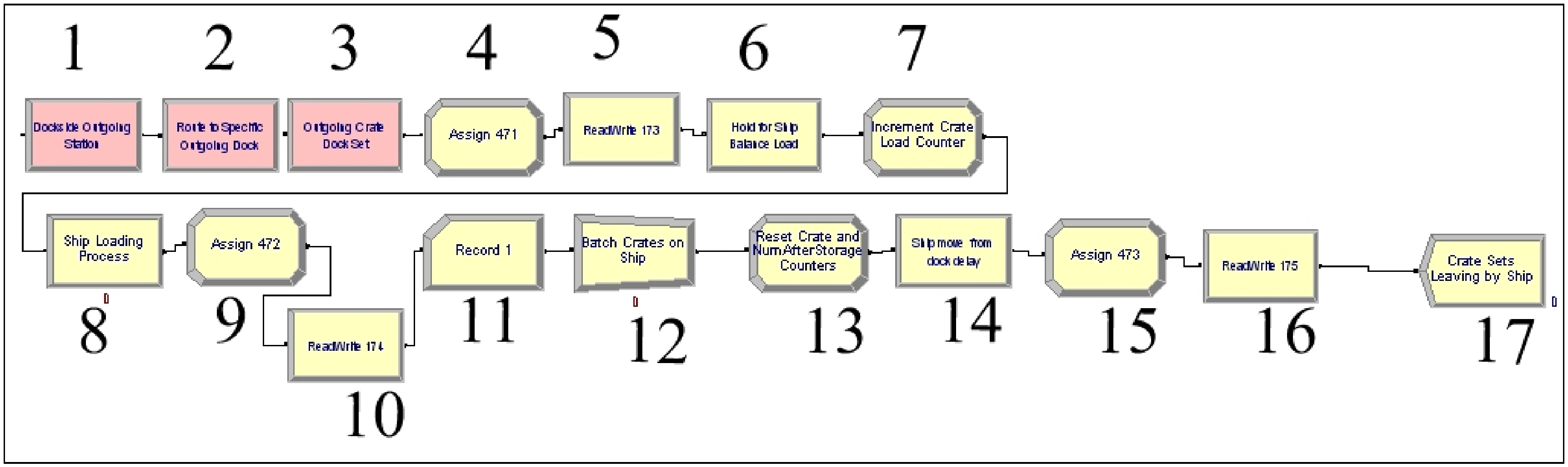

Figure 18 A graphical representation of the Outgoing Dockside operations logic at a typical port using Arena's inbuilt symbols. (Numbers reference step-by-step descriptions in Appendix A)

CHAPTER 4

SAMPLE SIMULATION OUTPUTS

4.1 General

The results of the simulation run presented in this section are based on data gathered from the model for the Port of Savannah, GA. Since the layouts of the ports are conceptual and similarly structured, the distinguishing variable in the modeled port operations is the rate at which containers arrive at the port. Therefore, analysis on only one port would suffice to demonstrate some of the information that can be obtained from the dynamically coupled modeled. In addition, the model has not been fully calibrated and validated with observational data. It should also be mentioned that since the port layouts are only conceptual, and actually observed train generations rates have not be also been used, the output results discussed in this section should not be viewed as real system performance indicators but rather as a demonstration of what can be obtained from the model.

4.1.1 Model setup parameters

The bulleted information below describes the setup information for the rail and port models.

- Trains are created into the model according to a random exponential distribution with a mean of 1 hour and the trains are dispatched from the listed 6 stations according to the probabilities shown in parenthesis.
    a. Chicago (0.15)
    b. Erwin (0.25)

c. Augusta (0.125)

d. Washington, DC (0.10)

e. Rocky Mountain (0.175)

f. Miami (0.20)

- Trains are also created and dispatched into the network from the three multimodal rail terminals at the ports.
- There is a single rail gate with average processing time of 10 minutes into the multimodal rail terminal.
- The adjacent multimodal rail terminal at the port has the following characteristics

  a. A shunting area with infinite capacity.

  b. 6 unloading and 3 loading platforms each with a crane of a single-container-lift capacity.
- There are 6 docking areas in the port. Each dock has an infinite length and so ship length does not influence dock availability. In addition, each dock has one unloading crane and one loading. The cranes are of single-container-lift capacity.
- There are 15 Customs agents at the port. The port also has two Inspections facilities and each Inspections facility has 5 agents.
- The type and number of the port transport vehicles are as listed below.

  a. 10 flatbed cranes

  b. 100 trucks

  c. 600 forklift vehicles

4.2 Model verification

The main sub-models – the railroad network and port terminal – were tested to ensure that their logic works as expected. A fixed number of entities – trains for the railroad and containers for the port terminal – were created into these sub-models and various counters were checked to see if the number-in equals the number-out. The time at which these numbers become equal was also recorded.

4.2.1 Verification of railroad sub-model's logic

In order to check the railroad sub-model it was decoupled from the port terminal model and a triangular train creation distribution was used to replace the port terminal's train creation process. Two train turnaround time categories were used in the triangular distribution. Category 1 has a minimum of 2 hours, maximum of 12 hours, and mean of 6 hours. Category has a minimum of 4 hours, maximum of 16 hours, and a mean of 10 hours. The results of the verification runs are shown in Table 1.

Table 1: Railroad sub-model logic verification results

No. of trains created	No. dispatched from stations	No. arriving at destination	Turnaround time category	Time taken (days)
500	500	500	1	17.7
600	551	398	1	too large
600	600	600	2	22.2
700	700	700	2	24.8
800	800	800	2	29.7
900	848	701	2	too large

The data presented in Table 1 show that the railroad network is working as expected and not resulting in deadlock situations. The issue with the second and sixth simulations is likely to be a case of demand being greater than the network capacity. This is because when the turnaround time is of the second simulation is made longer in the third simulation all 600 hundred trains are able to arrive at their assigned destinations. However, further verifications would be made in the future to ensure that the inference drawn is correct. In addition, verifications of all the sub-models included in the railroad model would be undertaken in the future.

4.2.2 Verification of port terminal sub-model's logic

A similar procedure to the one described for railroad sub-model was completed for the port terminal sub-model. However, in this case the models were not decoupled. The results of the verification runs are shown in Table 2.

Table 2: Port terminal sub-model verification results

Number of containers entering port	Number. of containers leaving port	Time taken (days)
5000	5000	5.6
10000	12341	8.6
20000	24298	18.9
50000	50000	41.7

The verification results in Table 2 show that the model is working as expected. It should be mentioned that no container was allowed to fail inspections in order to ensure that number-in would equal number-out.

4.3 Model Outputs

4.3.1 Container traffic through port terminal

The model was run for 26 continuous days, generating 141 containership calls, and 147 highway truck deliveries at the Port of Savannah. The number of trucks generated is notably small and this is due to the truck delivery setup parameters used. It also created 900 trains out of which 363 successfully traveled from various inland and port terminals to the Savannah multimodal rail terminal. All three transportation modes delivered a total of 28,738 containers to the Port of Savannah. A plot of container throughput characteristics is shown in the chart displayed in Figure 19.

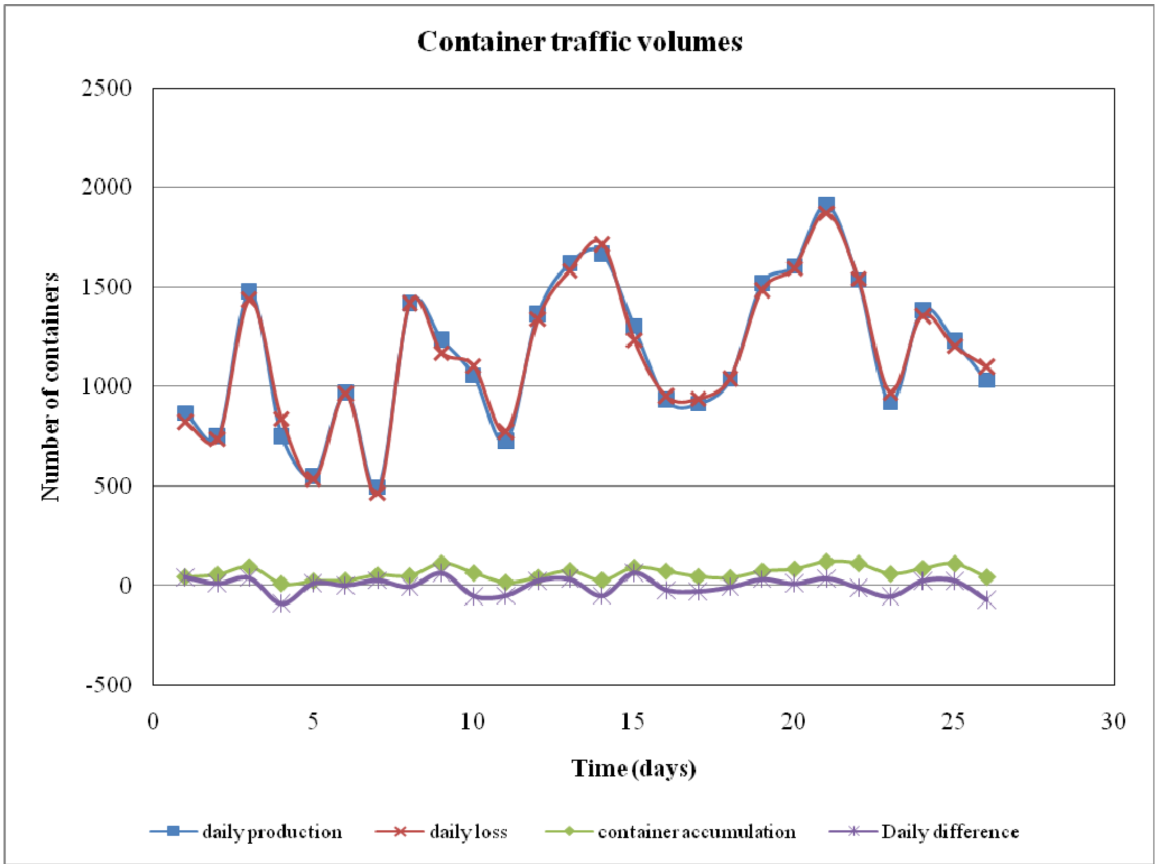

Figure 19 Simulated volumes of containers entering (production) or leaving (loss) port. Containers may be lost to other modes not included in model such as outgoing ships

The chart shows four different characteristics; the daily rate of production, the daily rate of loss, the daily difference in production and loss, and container accumulation volume on the port terminal. It can be inferred from the chart that the port is operating under capacity. This is because the daily rate of loss closely matches the daily rate of production and so almost all the containers produced in a day leave the port resulting in very little accumulation on the terminal. A number of factors in the model's setup might explain this observation. It is possible that there may not be a sufficient number of containers coming from landside to meet the capacity provided by the ships and a high turnout of containers from Oceanside. Increasing the number of containers unloaded from trains would likely place additional demand on ships. Another reason may be that the average numbers of containers unloaded from ships is well under capacity. A third reason may be that inter arrival times for ships is too long even though the number of containers they unloaded is at capacity. But this is unlikely to be the case since model uses a uniform rate that equals the total containership ship calls at the port in 2007.

4.3.2 Train movement trace plots

The model tracks the movement of each train over the rail network. Whenever a train stops, the duration of the stop and the location of the train are recorded in a database from which an entire trajectory of the journey can be plotted. Sample trajectory plots of some trains are shown in Figure 20, Figure 22, and Figure 21.

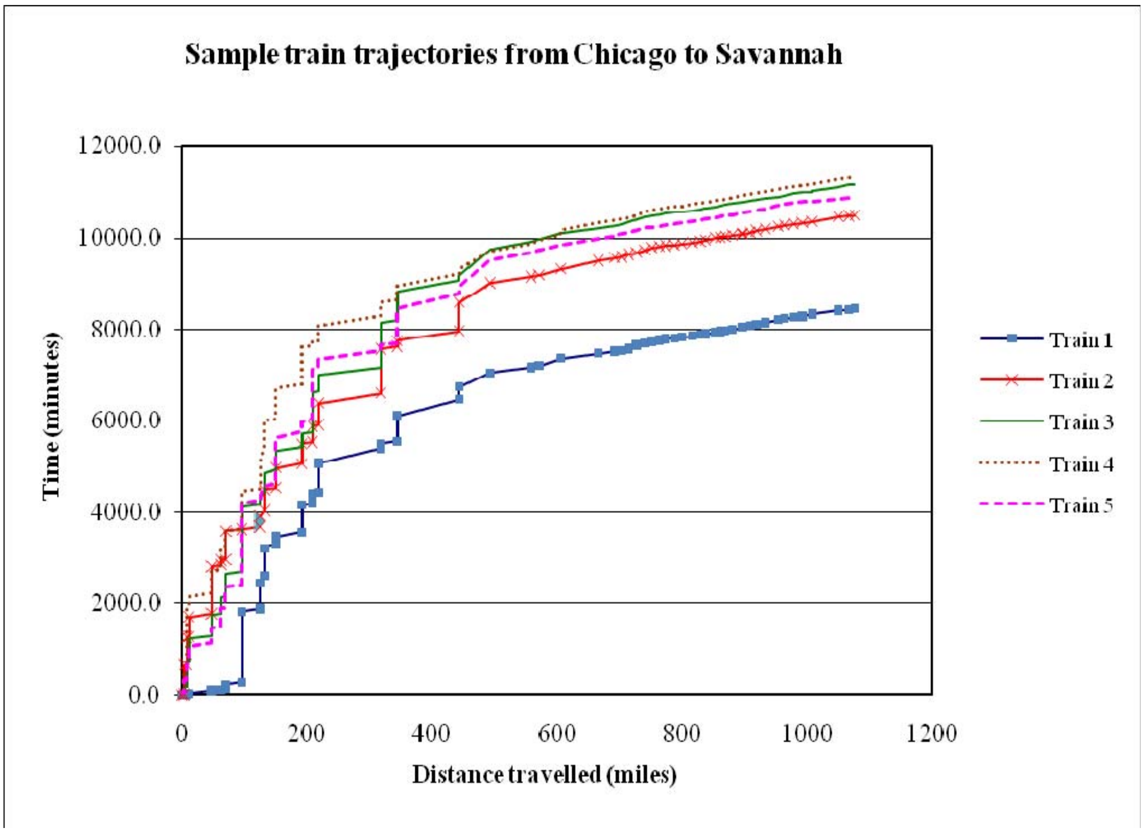

Figure 20 Time-Distance plots of sample trains from Chicago to Savannah

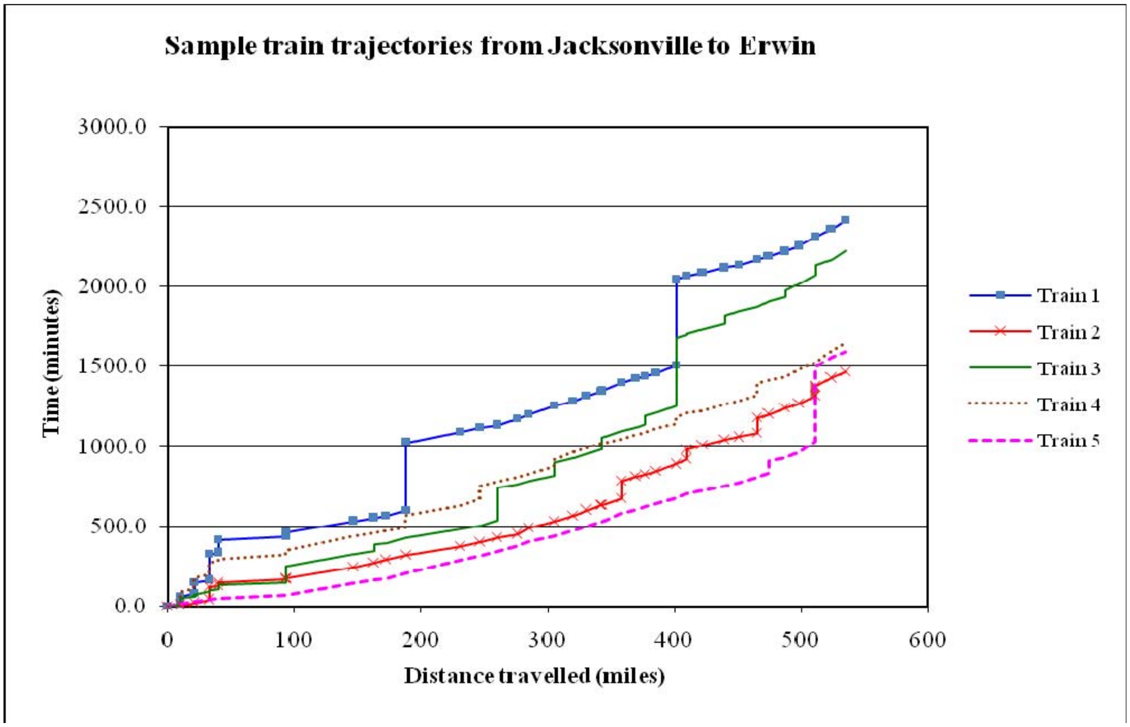

Figure 22 Time-Distance plots of sample trains from Jacksonville to Erwin

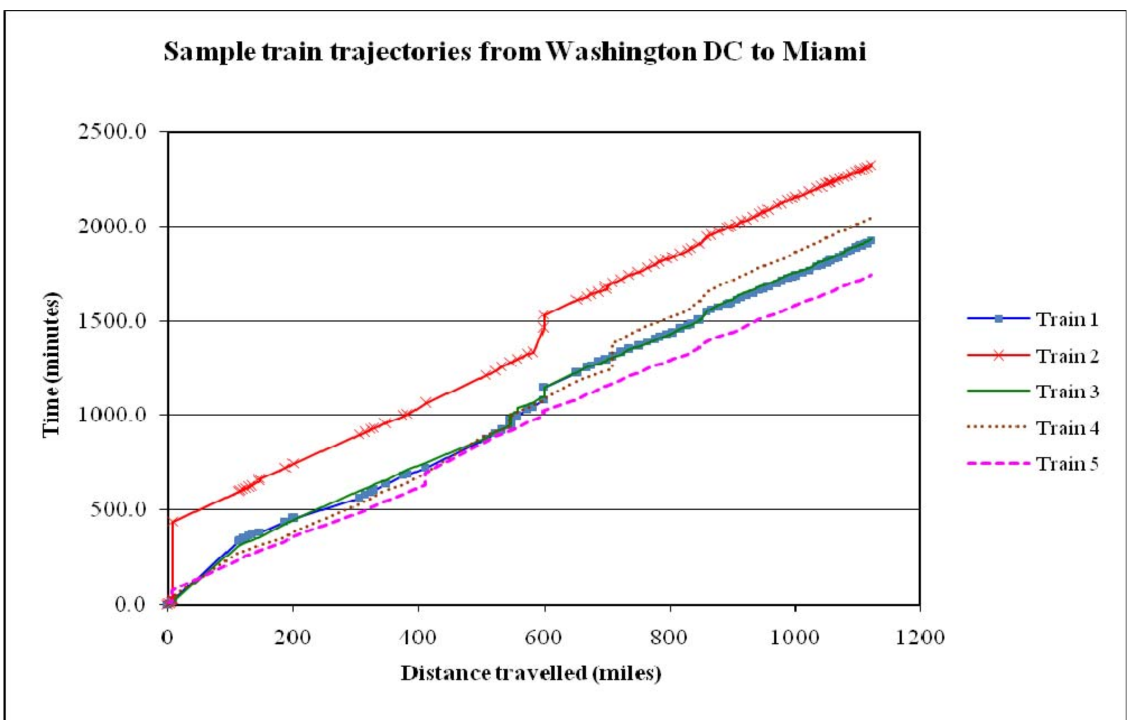

Figure 21 Time-Distance plots of sample trains from Washington, DC to Miami

The numeric suffixes attached to the trajectory names in the legend indicate the relative train creation times for the route. For example, Train 1 was created before Train 2. The vertical curves in the trajectory plots indicate locations where the train stopped to wait for the next available track segment or station on its route and the length of the curve on the vertical axis indicates the duration of the stop. Therefore, if multiple plots of train trajectories on a route are made they can help identify "bottlenecks" in the rail network.

4.3.3 Train idle and travel time

The output data used to generate the trajectory plots can be further aggregated to produce statistics on the train idle time as a percentage of the travel time. Table 3 presents some of these statistics on some selected train routes.

Table 3: Descriptive statistics on train idle time and travel time

Route	Statistic (mins)	Min.	Max.	Mean	Mean idle time / mean travel time
Savannah - Chicago	Travel time	3878	11697	8680	0.67
	Idle time	870	8579	5855	
Savannah - Erwin	Travel time	793	2330	1054	0.20
	Idle time	0	1494	209	
Savannah - Augusta	Travel time	62	727	263	0.38
	Idle time	0	541	99	
Savannah - Miami	Travel time	657	2347	1083	0.36
	Idle time	0	1627	391	
Savannah - Jacksonville	Travel time	53	819	107	0.40
	Idle time	0	752	43	
Savannah - DC	Travel time	510	2159	1201	0.24
	Idle time	0	1283	290	
Savannah – Rocky Mt.	Travel time	371	1079	616	0.33
	Idle time	0	708	205	
Savannah - Charleston	Travel time	97	603	222	0.50
	Idle time	0	497	111	

Figure 23 through Figure 30 presents a distribution of train idle time as a percentage of travel time on the selected routes.

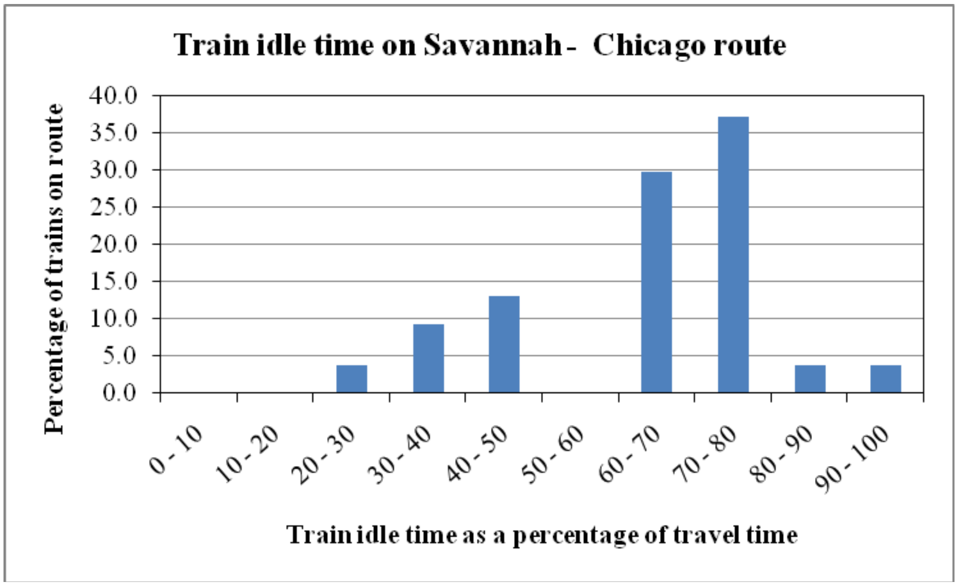

Figure 23 Distribution of train idle time as a percentage of travel time on Savannah – Chicago route

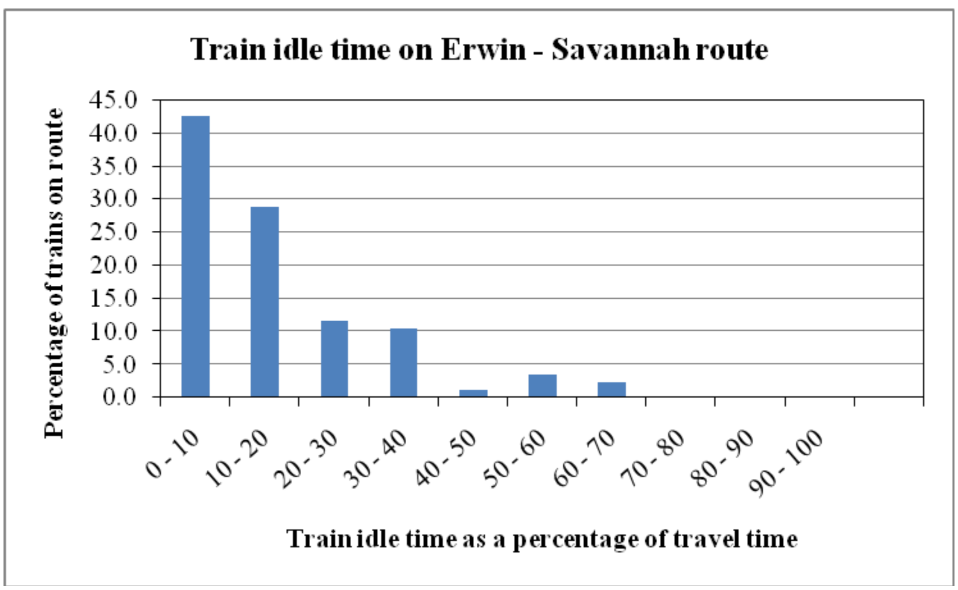

Figure 24 Distribution of train idle time as a percentage of travel time on Savannah – Erwin route

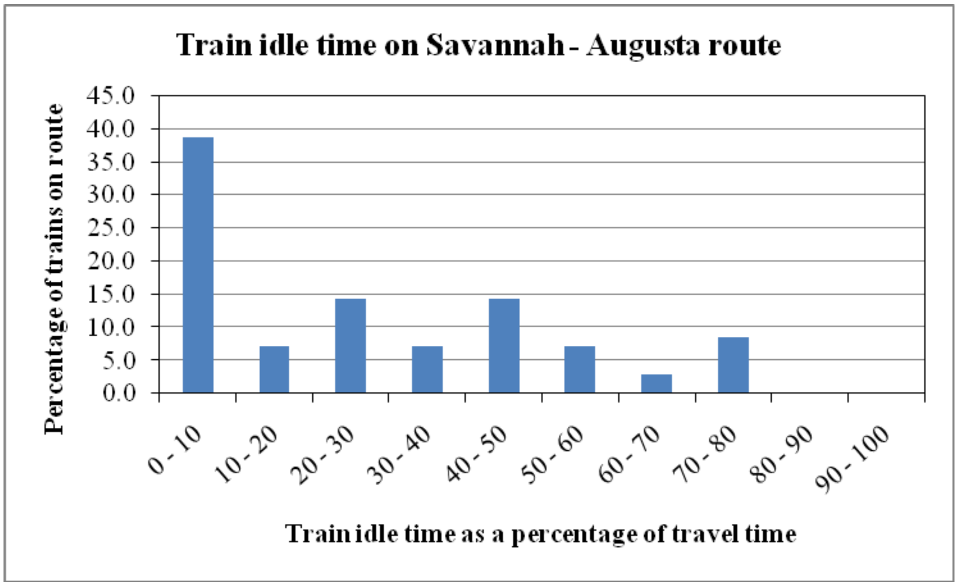

Figure 25 Distribution of train idle time as a percentage of travel time on Savannah – Augusta route

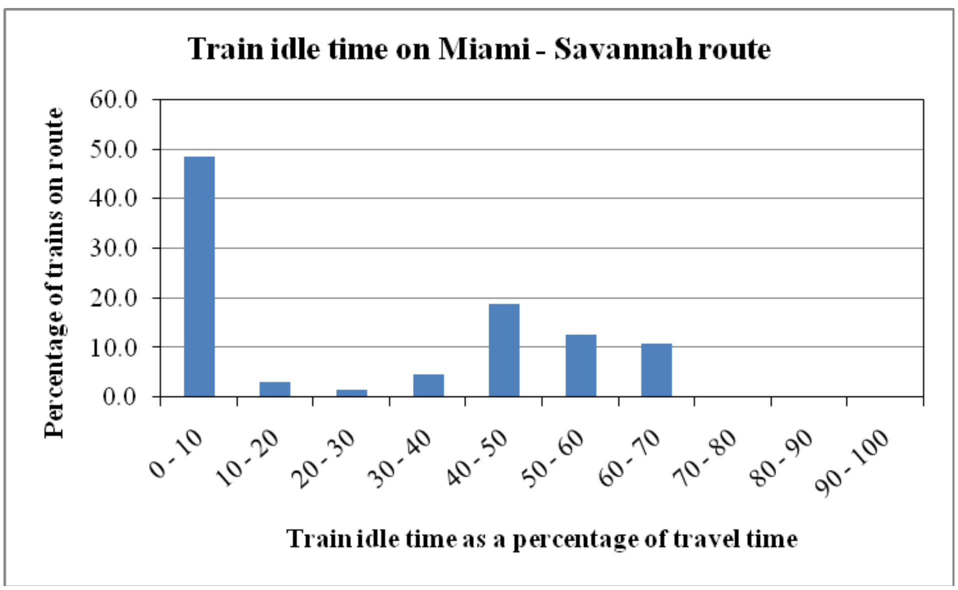

Figure 26 Distribution of train idle time as a percentage of travel time on Savannah – Miami route

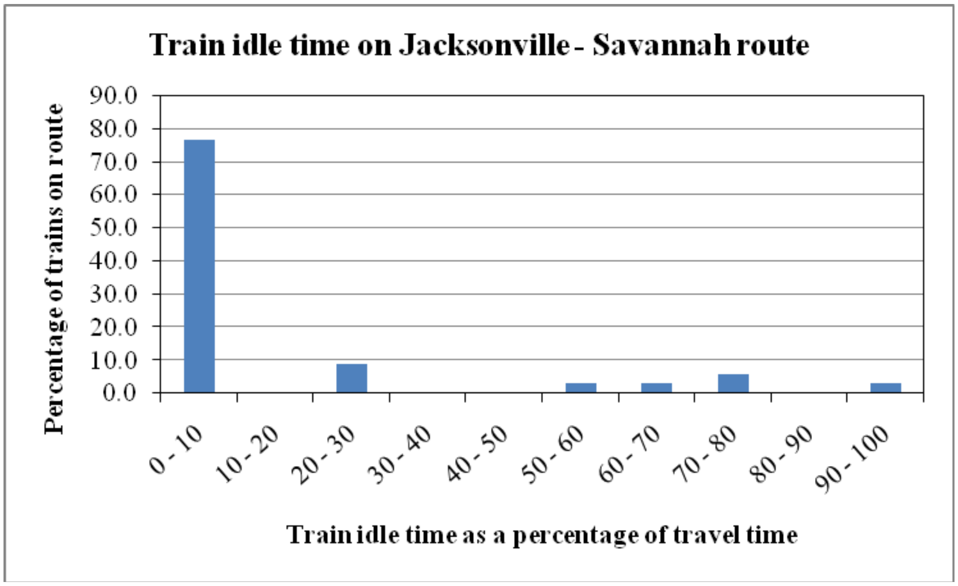

Figure 27 Distribution of train idle time as a percentage of travel time on Savannah – Jacksonville route

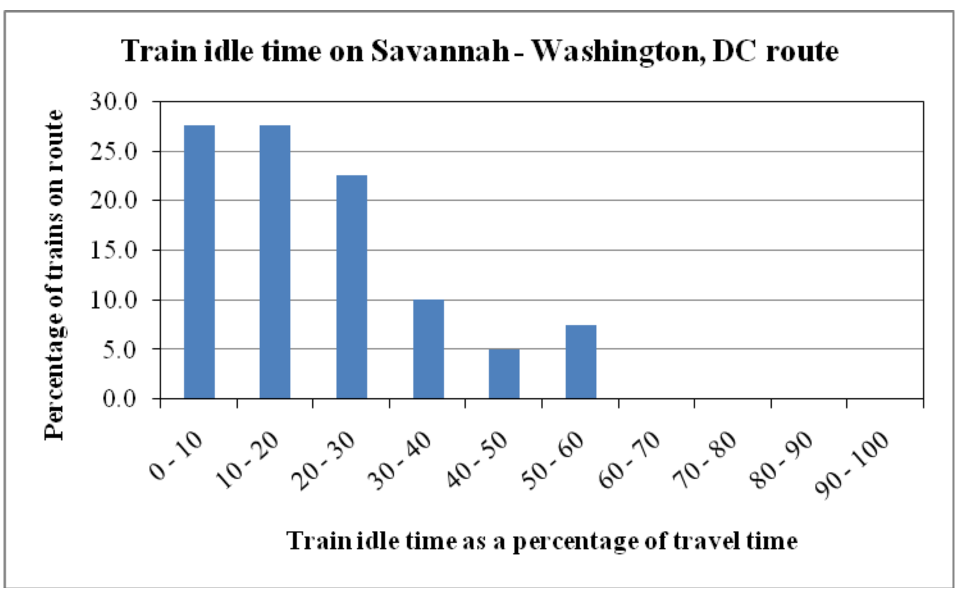

Figure 28 Distribution of train idle time as a percentage of travel time on Savannah – Washington, DC route

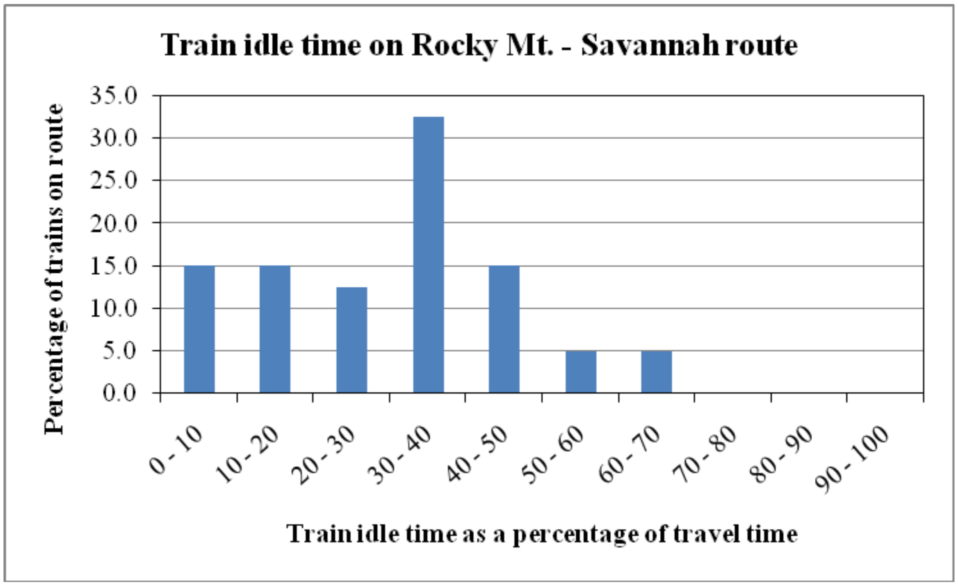

Figure 29 Distribution of train idle time as a percentage of travel time on Savannah – Rocky Mountain route

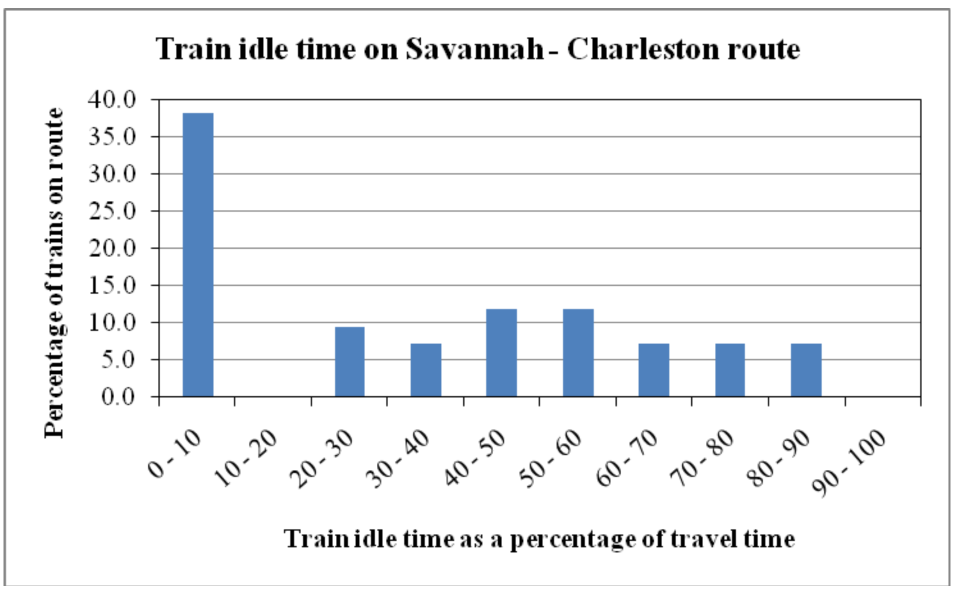

Figure 30 Distribution of train idle time as a percentage of travel time on Savannah – Charleston route

It seems odd that there are some data columns in the charts missing. For example, the 50 – 60 percent and 10 – 20 percent columns of Figure 23 and Figure 30 respectively. One reason may be that the number of trains analyzed on those routes is too small. In the future, the model would be run for a longer duration to obtain a larger set of trains on the routes for the idle time analysis.

4.3.4 Train speed profiles

The model also records the travel speeds of a train over every track segment on its route. The model's setup ensures that these speeds are in the range of 50 to 100 percent of the maximum permissible speed on the track segment. The speed profiles for some sample trains are shown in Figure 311 through Figure 3333.

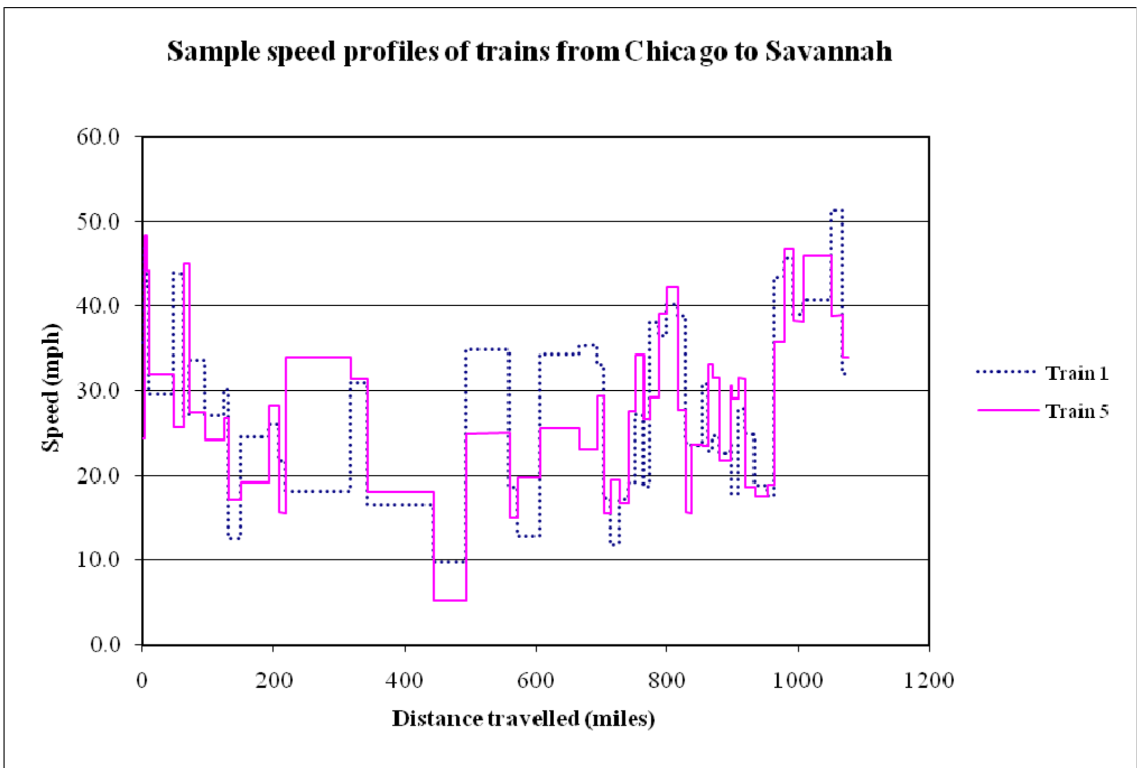

Figure 31 Sample speed profiles of trains from Chicago to Savannah

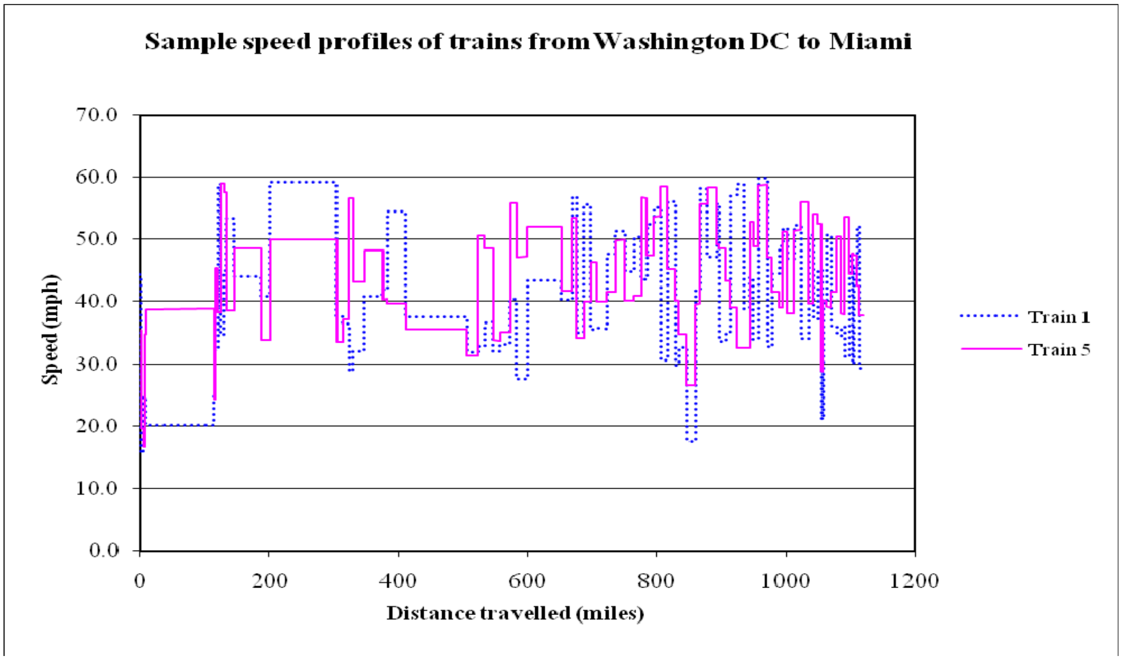

Figure 32 Sample speed profiles of trains from Jacksonville to Erwin

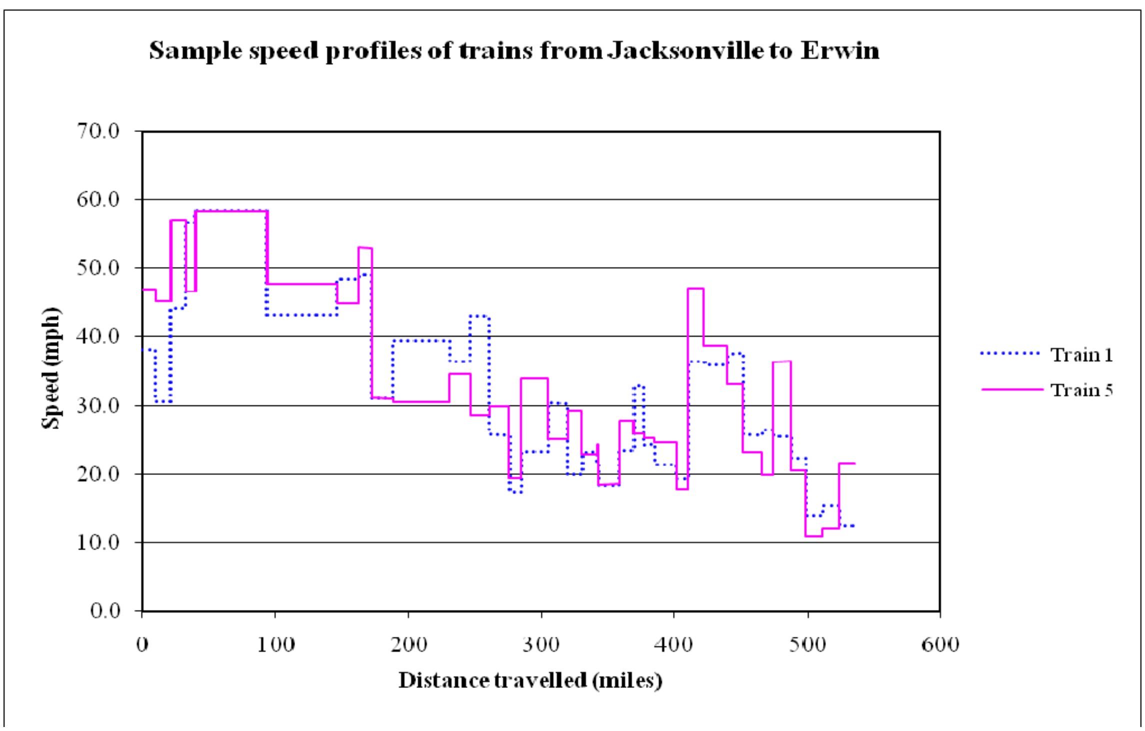

Figure 33 Sample speed profiles of trains from Washington, DC to Miami

The plotted speed profiles are shown as speed blocks with abrupt drops in speeds because the model does not currently consider train accelerations and decelerations. This is due to the assumption of a simplified travel time function as explained in section 3.2.1 of the previous chapter.

4.3.5 Train demand on multimodal rail terminal

Trains arriving at a multimodal port terminal have to wait at the shunting area for an available unloading crane before entering the rail terminal. In addition, unloaded trains may also wait in the shunting area until their designated containers are ready. Thus, a good way to estimate the train demand on the rail terminal would be to keep track of the number of trains waiting in the shunting area at a given time interval, say every day. The daily train parking demand in the shunting area is presented in Figure 3434. The plotted chart covers the entire duration of the simulation.

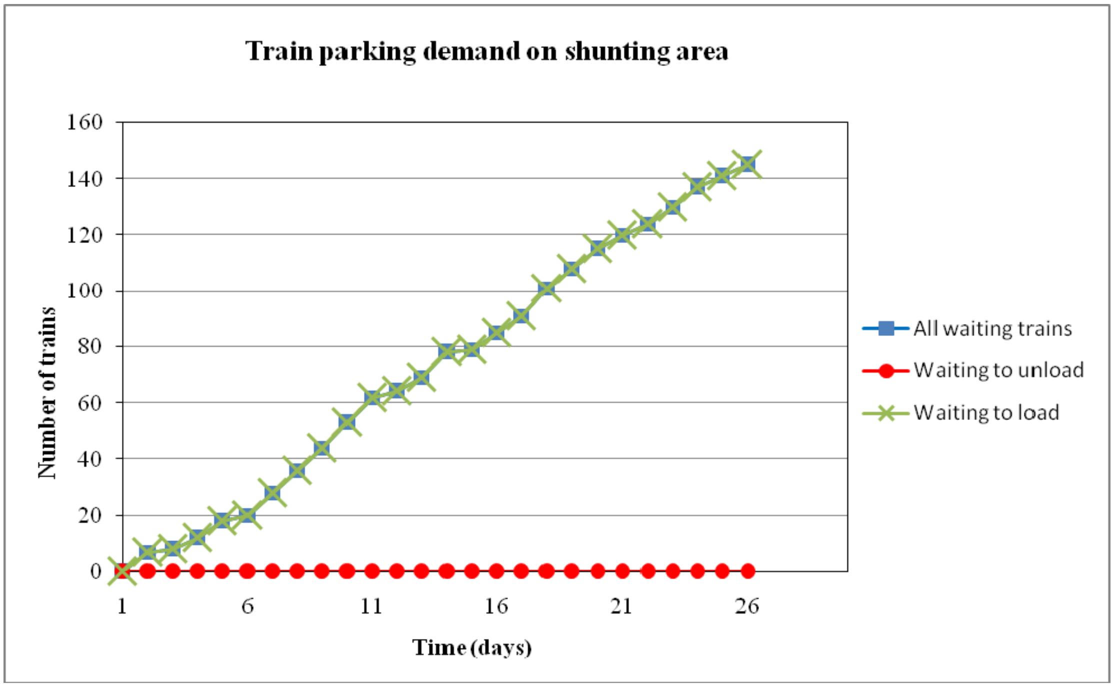

Figure 34 Train parking demand on shunting area at multimodal rail terminal

It can be inferred from the chart in Figure 3434 that parking demand within the simulation period of 26 days continues to increase. Disaggregating the parking demand into trains waiting to unload and trains waiting to load shows that the entire demand is being created by trains waiting to load. This observation agrees with the earlier of the observation of little container accumulation on the port terminal. It is obvious from these two charts that there is an imbalance and the model setup doesn't match demand between trains and containers. Future studies on the model should seek observed container and train demand data from the port to refine the model's setup parameters.

4.3.6 Ship servicing time

The model stores information on a ship's port residence time, i.e., interval between arrival and departure from the port. It also records the time between a ship's arrival and berthing at an available dock. Out of the 141 containership calls at the port during the simulation run, 84 ships were successfully loaded with containers and departed from the port. The rest of the ships were in queue waiting for an empty dock to berth. The descriptive statistics on port residence time and time taken to berth for the 84 ships is shown in Table 44.

Table 4: Descriptive statistics on ship servicing times at Port of Savannah

Statistic	Time (days)				
	Number. of ships	Min.	Max.	Mean	Standard. deviation
Port residence	84	2.8	6.0	4.45	0.98
Time to berth	90	1.7	4.9	3.23	2.98

The distribution of port residence time and time to berth is also shown in Figure 35.

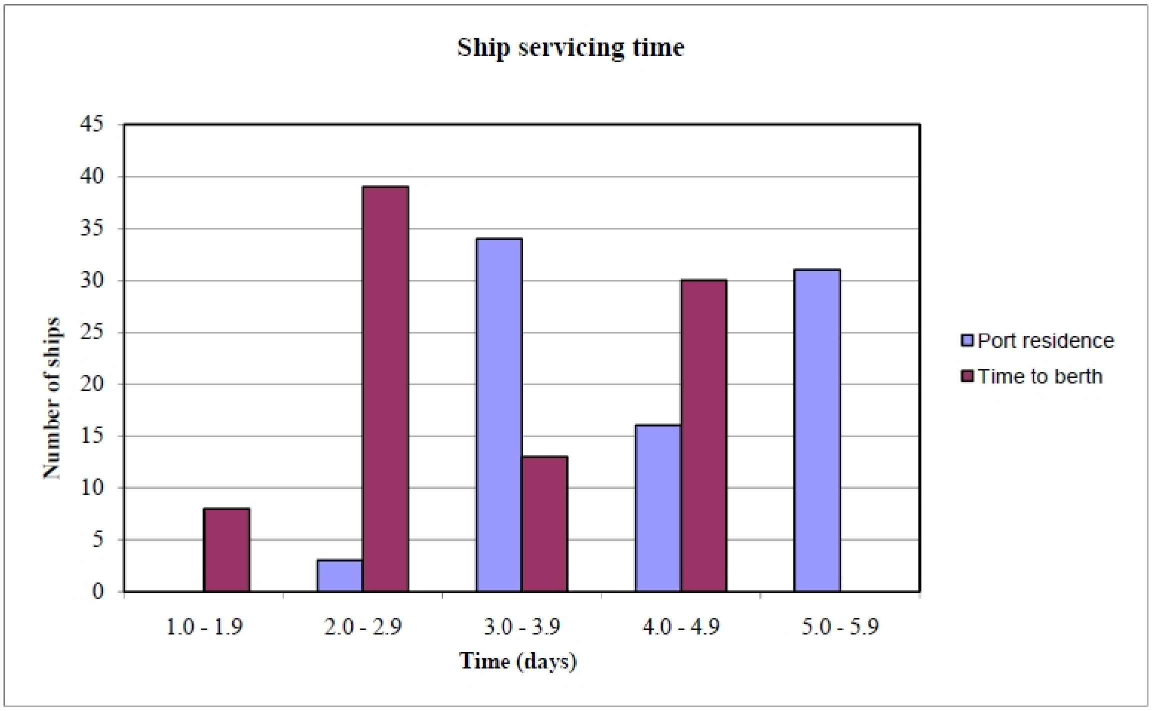

Figure 35 Distribution of ship servicing time at the modeled Port of Savannah

4.3.7 Container residence times

The model also collects data on the length of time containers stay within the port or rail terminal before being loaded onto a ship or train respectively. This wait time between arrival and loading has been termed "residence time" in this report and it should be noted that it doesn't include time spent aboard a ship or train that is waiting to depart. Table 55 shows some descriptive statistics on container residence times within the port terminal and adjacent multimodal rail terminal.

Table 5: Descriptive statistics on container residence times within port and rail terminals

Statistic	Number. of containers	Min.	Max.	Mean	Standard. deviation
Port terminal residence time (hours)	23514	0.2	8.92	0.89	1.61
Rail terminal residence time (mins)	23781	2.0	154	18.45	15.33

The statistics presented in Table 55 shows that container residence times on both terminals vary widely. Within the port, it takes about one hour on average to load a container onto a ship. However, the same process could take between 0.2 to 9 hours. Containers delivered to the rail terminal may also spend as little as 2 minutes or as much as 2.5 hours before being loaded onto a train, however, containers are likely to spend 18 minutes on average. Figure 3636 shows the distribution of container residence times within the port. It can be inferred from the bar chart that most containers – 62 percent – have a residence time of less than 30 minutes. This implies the container residence time distribution within the port is skewed to the right with a median value less than mean. Figure 37 also shows the distribution of container residence times within the rail terminal.

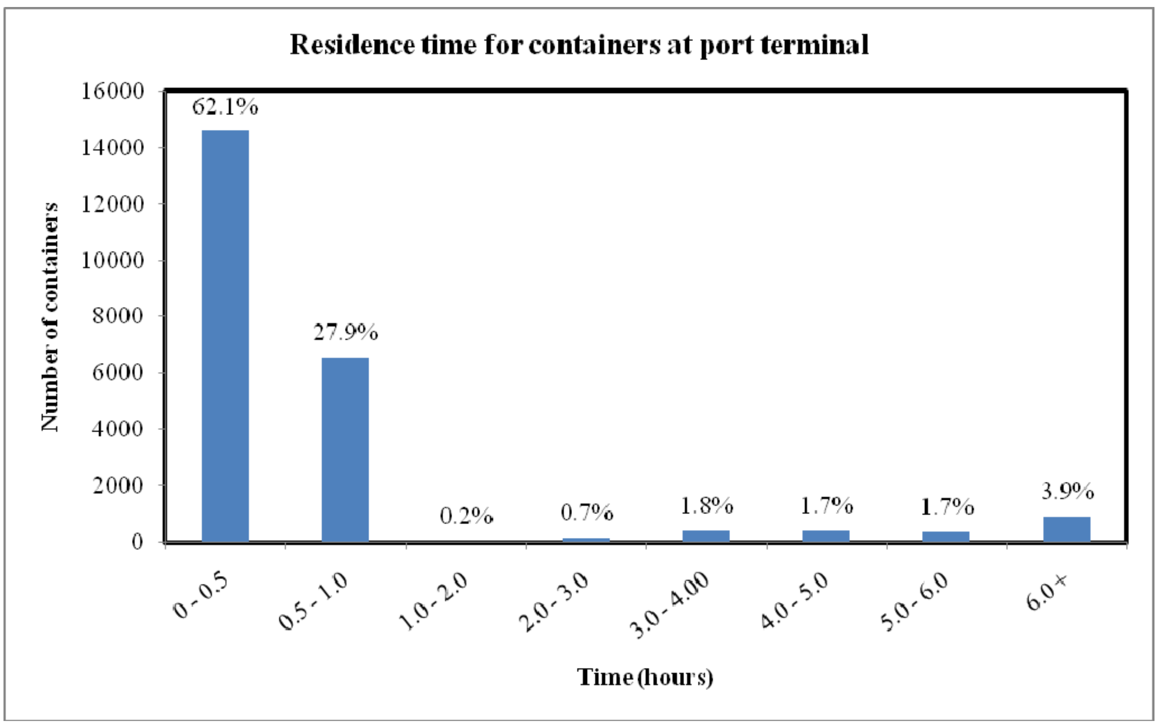

Figure 36 Distribution of container residence times at port terminal

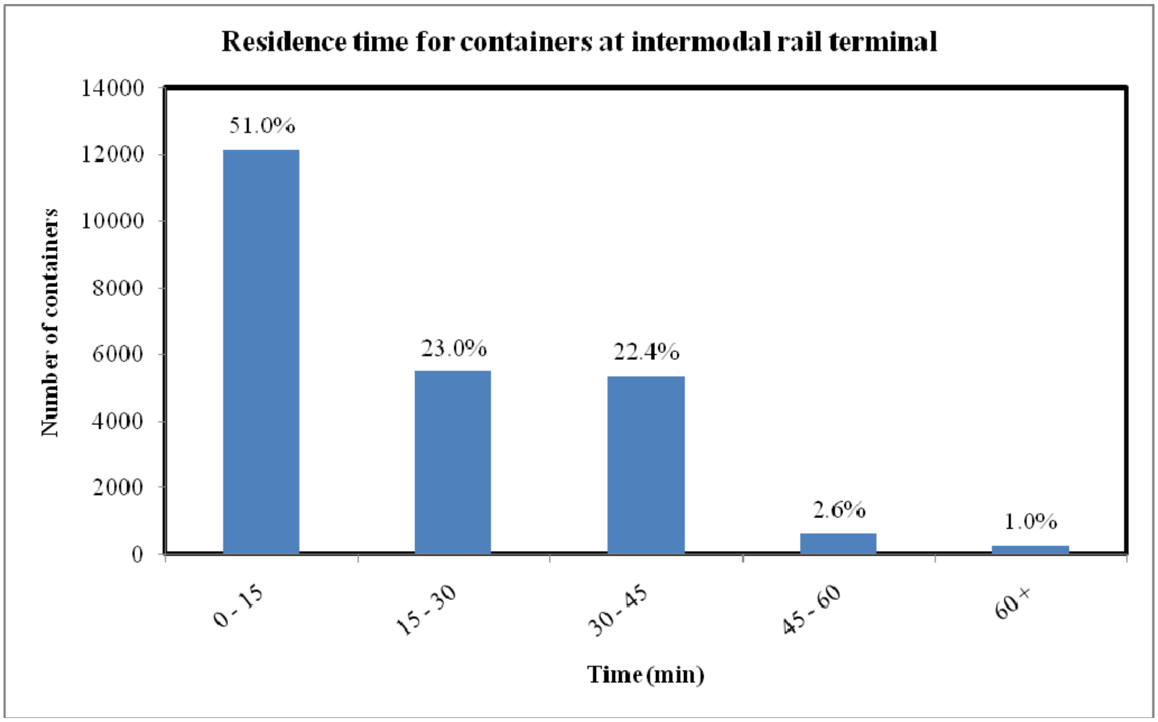

Figure 37 Distribution of container residence times at multimodal rail terminal

The model also collects data on the amount of residence time that a container spends idle inside the port. And idle container may be one that is temporarily held in storage or one that is sitting in queue at the Inspections facility. Table 66 presents some descriptive statistics on container idle time and Figure 388 shows the distribution of idle time as a percentage of terminal residence time.

Table 6: Descriptive statistics on container idle time inside port terminal

Statistic	Number. of containers	Percentage of container residence time			
		Min.	Max.	Mean	Standard. deviation
Idle time	23514	70.10	99.30	84.59	8.68

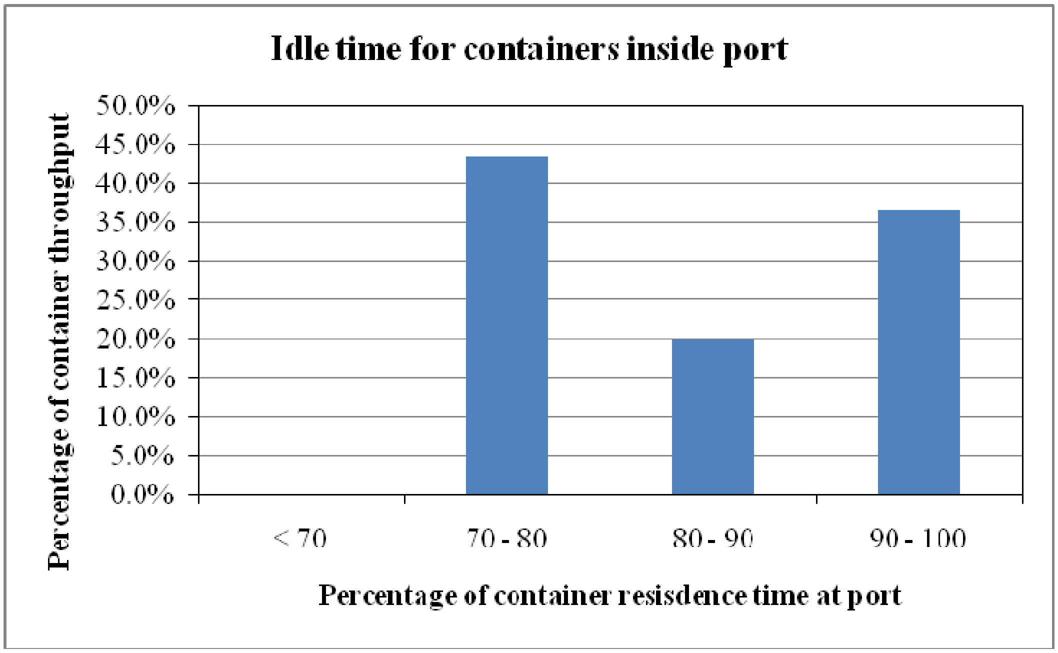

Figure 38 Distribution of container idling time as a percentage of residence time

Given that Figure 3636 shows that 90 percent of the containers are subject to less than 1 hour residence time, it seems surprising that Figure 388 is implying that about 36 percent of containers spend about 90 to 100 percent of their residence time idle. It should however be noted that the current port layout in the model is only conceptual. For example, in the port layout used the length of transportation link between the Dockside and Inspections is only 500 ft, and with an average forklift speed of 10 mph (880 ft per min.) it takes only about 34 seconds to transport an unloaded container from the dock to Inspections. When these notably short travel times are compared to say a 5 minute processing times at on-port facilities the cumulative idle time can be a large proportion of total residence time. In addition, this section is only intended to demonstrate some of the information gathering capabilities of the model and not to represent actual port performance. Obtaining a more accurate estimate of actual port performance will require extensive calibration and validation of the model with real-world observations at specific ports.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

5.1 Data gathering capabilities of model

This research study has developed a dynamically coupled model of the multimodal freight transportation system. The developed railroad network, multimodal rail terminal operations, and seaport operations have been modeled using Arena ® simulation software.

The railroad network modeled covers most of the CSX railroad network in the southeastern U.S. while the multimodal terminals modeled are made up of the Ports of Savannah, GA, Charleston, NC, Jacksonville, FL, and their adjacent CSX multimodal rail terminals. The modeled port layouts are only conceptual and as such the output from the simulation runs do not actually inform on port performance but only serves to demonstrate the data gathering capabilities of the model.

Several charts and tables have been developed from the model run data to demonstrate some of the data gathering capabilities of the developed model. Plotted train trajectories trace the entire journey of trains from origin to destination and they help identify locations where trains had to stop as well as the duration of the stop. The data on train trajectories have also been aggregated to show train idle times as a percentage of travel times. Train speed profiles graphically show the simulated travel speeds of trains. However, because the model does not currently consider train acceleration and deceleration the speed profiles appear as speed blocks with abrupt drops or increases in speeds. Statistics on train parking demand on the shunting area connected to the multimodal rail terminal is also presented. The plot on parking demand shows that with the current model setup, a simulation run longer than 26 days would be required before a fairly constant demand can be observed.

Ship waiting statistics have also been presented on the distribution of waiting times for ships to berth and depart from the port. These statistics are likely impacted by the efficiency of loading / unloading operations, the inter-arrival time for trains, and/or the container capacity and destination of trains arriving at the port.

Container throughput statistics have also been shown. The analyses presented covers the daily production and loss of containers in the port terminal, the residence time of containers within the port or rail terminal, and the percentage of residence time at the port that containers spend idle.

5.2 Future research needs

There are several limitations of the present model which should be addressed by future research in order to develop a more robust multimodal freight transportation model. The main limitation is the adopted train movement logic which does not distinguish between single-track and multi-track configurations. This causes reduced rail network capacity, especially in areas of high traffic volume, and rippling compound delays. Future research should modify the train movement logic to incorporate a more efficient multi-track train movement algorithm such as the one proposed by Lu and co-workers (Lu et al. 2004).

In addition, future research should also attempt to use the actual track control segment on the rail tracks, i.e., the actual Direct Traffic Control (DTC) or Automatic Block Signal (ABS) blocks. This information could be obtained from published CSX

Subdivision Timetables. The combination of these track segments and improved movement logic would improve the accuracy of network capacity estimates obtained from model.

The next limitation of the model is the large computational costs due to the discrete process modeling methodology. The use of a single container resolution level, and size of the modeled rail network generates millions of discrete entities whose collective memory requirements comes at a high computation cost. Further track segmentation as described above would augment this limitation and might seriously hamper the model's ability to efficiently simulate such large scale dynamically coupled multimodal freight transportation systems, except on computers with very high system resources. Future research should investigate the resolution of this challenge through the use of a combined discrete and continuous process modeling methodology. This combined modeling methodology might be carried out in two logical stages. In the first stage the multimodal freight transportation system should be modeled only within a certain radius around terminals to capture the system dynamics in the immediate areas and generate throughput statistics for the terminals. Then in the second stage the generated throughput statistics of the various multimodal terminals can be used to create continuous process models of the terminals which can then be used to replace the discrete process models of the terminals in the large scale multimodal transportation system. By this approach, many of the limitations of the continuous approach can be avoided while preserving higher model throughput. This would allow for the use of replicated trials to augment the statistical strength of the results. However, this combined approach might result in a trade-off between computational cost and the model's feedback capability, especially if the port terminal is the source of capacity constraint in the multimodal system. Therefore, the merits of the proposed methodology must be critically assessed in future research work.

Another limitation of the model is that the current ship arrival and dock space allocation logic does not adequately discriminate between containerships. The length of a ship must influence the availability or allocation of dock space for berthing. This lack of discrimination affects the accuracy of gathered statistics on ship waiting times and dock resource allocation. Future research should seek to build a discrimination algorithm into the model. In addition, there is a need to optimize the allocation of storage space on the terminal yard. Currently model does not optimize yard storage space. However, this is essential to ensure that the allocation and utilization of resources such as forklifts and cranes can be better studied. A yard storage space allocation and optimization module should be added to the model's internal logic in future research.

The next limitation to the present model is that port layouts and details are only conceptual and may not accurately represent what actually exists at the three ports. This makes it impossible to calibrate and validate any results from model simulation. Future research should obtain real data that can be used to modify the ports' layouts, calibrate the model and then check the validity of the outputs.

Last, future research should apply ARENA's template development to develop a stand-alone template that can be deployed on any computer to run a multimodal terminal and rail network simulation. However, the identified limitations will need to be addressed before any template development can commence.

APPENDIX A

STEPWISE EXPLANATION OF VARIOUS SUB-MODELS LOGIC

Logic for sub-model shown in Figure 6

The various branches in the flowchart have the same logic. This section outlines the steps within a typical branch.

Step 1: Creates trains into the model according to a random exponential distribution with a mean of 1 hour.

Step 2: Assigns the created trains a unique ID

Step 3: Decides the origin station to dispatch trains from. The decision is according to a random probability distribution and there are 6 possible origin stations.

Step 4: Decides the final destination station of the trains

Step 5: Assigns a route ID with which it can identify all the track resources on its route from the database. Also assigns a time stamp to show when train was created at the station.

Step 6: Writes the train ID, status, and time stamp information into a database.

Step 7: Writes the train ID, assigned origin station and destination stations into another database.

Step 8: Holds the train and scans the model to ensure that the network deadlock avoiding conditions are met before releasing the train

Step 9: In this module the train waits to seize a connecting track to the next station / siding. It also seizes a track at the next station / siding before it can leave origin station.

Step 10: Sets a number of counters, assigns new attributes, and updates the train status and time stamp information.

Step 11: Writes train information assigned in the previous step to different databases.

Step 12: Routes the train from the origin station to the next station. Also calculates the travel time required for the trip.

Logic for sub-model shown in Figure 7

The various branches in the flowchart have the same logic. This section outlines the steps within a typical branch.

Step 1: Holds a set of general route IDs which are matched to the general route IDs assigned to the trains from the port terminal. There are three general routes / branches shown in the flowchart.

Step 2: This step assigns a unique train ID and time stamp to the trains. The time stamp shows the time of creation at the terminal.

Step 3: Decides the general route or logic branch the train should follow. The decision is based on the matched route IDs in step 1.

Step 4: At this step the destinations of trains are decided based on a probability distribution and the number of modeled terminals on the general route decided in step 3.

Step 5: Assigns a train a more specific route ID, updates the status, origin station, and destination station attributes. The specific route ID is used to identify the track resources it needs for the entire trip from the database.

Step 6: Writes the train ID, status, and time stamp information into a database.

Step 7: Writes the train ID, assigned origin station and destination station into another database.

Step 8: Holds the train and scans the model to ensure that the network deadlock avoiding conditions are met before releasing the train

Step 9: In this module the train waits to seize a connecting track to the next station / siding. It also seizes a track at the next station / siding before it can leave the rail terminal

Step 10 and Step 11: Sets a number of counters, assigns new attributes, and updates the train status and time stamp information.

Step 12: Writes train information assigned in the previous step to different databases.

Step 13: Routes the train from the rail terminal to the next station. Also calculates the travel time required for the trip.

Logic for sub-model shown in Figure 8

There are two possible paths for trains entering this logic. The first path (A) is for trains that are one station away from their assigned destination stations. The second path (B) is for trains that are further away from their assigned destination stations.

Path A

Step 1: Trains enter an assigned station / siding at the end of a trip through this block. The block holds a set of all possible stations / sidings on the railroad network.

Step 2: Release the previously held track segment resource

Step 3a: Routes trains to either path A or path B

Step 4a: Looks for the station ID of the train's assigned destination.

Step 5a: Assigns trains the ID of the track segment leading into the station.

Step 6a: Updates the trains status and time stamp information.

Step 7a: Writes the some previously assigned train attributes to the database.

Step 8a: Holds the train and scans the model to ensure that the network deadlock avoiding conditions are met before releasing the train

Step 9a: In this module the train waits to seize a connecting track to the next station / siding. It also seizes a track at the destination station before it can leave its current station.

Step 10a: Checks for the correct entrance gate based on the trains direction in reference to the station.

Step 11a: This block is used to update and assign other train attributes.

Step 12a: At this step the train releases its currently seized station / siding resources before proceeding onto the connecting track segment leading to the destination station.

Step 13a: Updates train's status and other attributes

Step 14a: Writes train information to the database

Step15a: Routes the train to the destination station.

Path B

Step 4b: Checks if the train's current position is a junction.

Step 5b: If the train is at a junction then this blocks checks if the route would require a change in direction.

Step 6b: If a change in direction is required then this block updates train's attributes such as direction of movement.

Step 7b: Checks the trains assigned direction of movement and decides on which part of the logic to use. This is necessary because of the sequential numbering of resources in the database. For example, a northbound train may have a descending order of assigned track resource IDs while a southbound train may have an ascending order of assigned track resource IDs.

Step 8b: Updates the train's status and time stamp information.

Step 9b: Writes the some previously assigned train attributes to the database

Step 10b: Holds the train and scans the model to ensure that the network deadlock avoiding conditions are met before releasing the train to proceed.

Step 11b: In this module the train waits to seize a connecting track to the next station / siding. It also seizes a track at the next station / siding before it can leave its current station

Step 12b: Updates and assigns other train attributes.

Step 13b: Writes train information to the database

Step 14b: Releases its currently seized station / siding resources before proceeding onto the next connecting track segment on the route

Step15a: Routes the train to the next station siding.

Logic for sub-model shown in Figure 9

This logic applies to trains reaching destinations other than ports

Step 1: Trains enter their destination stations through this block

Step 2: Releases the connecting track into the station.

Step 3: Updates the train's attributes

Step 4: Writes train information to the database

Step 5: Removes train from the model.

Logic for sub-model shown in Figure 10

This logic has one path for trains and 5 different paths for containers.

Path A (for trains in the logic)

Step 1: Trains enter the logic through this block

Step 2: The train releases the held connecting track into the station

Step 3: Assigns and updates other train attributes

Step 4: Determines whether there is an available unloading deck and crane in the terminal. If the answer is yes the train proceeds to step 9 otherwise it is sent to the shunting area to wait.

Step 5: Trains enter the shunting area through this module

Step 6: Updates the storage counter in the shunting area

Step 7: Updates the train's status and other attributes

Step 8: Writes train information to database

Step 9: Trains at this block wait in a FIFO queue to seize an unloading deck and unloading crane.

Step 10: Removes train from shunting area and updates the storage counter

Step 11: Train proceeds to rail gate after a fixed delay

Step 12: Updates train's status and other attributes

Step 13: Writes train information to database

Step 14: Train undergoes gate processing

Step 15: Updates train’s status and other attributes

Step 16: Writes train’s information to database

Step 17: Train incurs parking time delay on track

Step 18: Applies additional delay for switching

Step 19: Updates train’s status and other attributes

Step 20: Writes train information to database

Step 21: The train’s cargo is unbatched into an assigned number of containers

Step 22: Holds train until all containers are unloaded

Step 23: Releases the track at the unloading deck

Step 24: Train proceeds to shunting area to wait until it is signaled to proceed to loading deck.

Step 25: Updates the train’s status and other information

Step 26: Writes train information to database

Step 27: Stores train in shunting area and updates the storage counter

Step 28: Holds train until it is signaled for loading

Step 29: Removes train from shunting area and updates the storage counter

Step 30: Updates the train’s status and other attributes

Step 31: Writes train information to database

Step 32: Train proceeds to loading deck.

Path B (for containers arriving by train and departing by truck or ship)

Step 1a: Unbatches into the number of assigned containers.

Step 2a: Assigns containers to a temporary storage on train.

Step 4a: Updates a container's status and other attributes

Step 5a: Writes container information to database

Step 6a: Uses the container's assigned destination ID to select the branch of logic to follow. There are 6 possible destinations; 3 local truck distribution centers, 1 long distance truck center, another terminal via a different train, or port terminal.

Step 7a: A container assigned to the port terminal must request for a truck to take it to the Truck Receive terminal. The request for a truck is the only activity that is different from the logic for other landside destinations via highway trucking.

Step 8a: Unloads container from train onto truck.

Step 9a: Removes container from temporary storage on train and updates the storage counter.

Step 10a: Updates container's status and other attributes.

Step 11a: Writes train information to database

Step 12a: Transports container to assigned destination

Path C (for containers arriving by train and leaving by another train to another terminal)

The initial part of the logic is the same as Step 1a through Step 6a.

Step 7b: Unloads container from train

Step 8b: Removes container from temporary storage on train and updates the storage counter.

Step 9b: Updates the container's status and other attributes

Step 10b: Checks the availability of a loading train assigned to the container's general route ID.

Step 11b: If there is an available train from Step 10b, then this block assigns a specific route ID and updates the train load counter.

Step 12b: Stores container on train.

Step 13b: Updates the container's status and other attributes

Step 14b: Writes train information to database

Step 15b: Holds the train until all assigned containers are loaded.

Step 16b: Applies a train departure delay

Step 17b: Removes train from track storage and updates the track storage counter

Step 18b: Batches the loaded containers into one entity type

Step 19b: Releases the loading track and deck

Step 20b and 21b: Updates the container's status and other attributes

Step 22b: Writes train information to database

Step 23b: Updates train's attributes

Step 24b: Routes train out of rail terminal

If no train was available from the check performed in Step 10b then the following occurs

Step 11c: Updates the train's information

Step 12c: Checks to see if there is an empty loading track or deck available to signal a train from the shunting area to proceed to that deck

If there is an empty loading track or deck then …

Step 13c: Updates the container's status and attributes.

Step 14c: Writes the container's information to the database

Step 15c: Scans the model to see if there is a shunting train waiting to be loaded.

Step 16c: Seizes the empty track for its general route ID.

Step 17c: Updates some counters and assigns the number of containers to be loaded on the track.

Step 18c: Signals for a shunting train to be released to the loading deck. From here the logic joins Step 12b.

If there wasn't an empty loading track or deck from Step 12c then …

Step 13d: Updates the container's status and attributes

Step 14d: Writes the container's information to the database

Step 15d: Holds the container in a temporary storage until a train and loading deck is later assigned to its general route ID

Step 16d: Updates some counters and container attributes. The logic then joins Step 12b.

Path D (for containers arriving from seaside)

<u>Step 1e</u>: The containers enter the rail terminal through this block

<u>Step 2e</u>: Updates the container's status and other attributes.

<u>Step 3e</u>: Writes the container's information to the database

<u>Step 4e</u>: Unloads the container from the transporter

<u>Step 5e</u>: Releases the transporter

<u>Step 6e</u>: Assigns the container a general route ID. From here the logic joins Step 10b.

Logic for sub-model shown in Figure 11

There is one path for ships and path for containers in this logic

Path A (for ships)

<u>Step 1</u>: Creates ships into the model

<u>Step 2</u>: Assigns some attributes such as number of crates to load and unload, unique ship ID, ship status, and time stamp.

<u>Step 3</u>: Writes ship information to database

<u>Step 4</u>: Updates the ship's station name attribute

<u>Step 5</u>: Ship queues to seize a dock to berth

<u>Step 6</u>: Routes ship to the seized dock.

<u>Step 7</u>: A ship arrives at a dock through this module

<u>Step 8</u>: Updates ship's attributes and also assigns new ones

<u>Step 9</u>: Writes ship's attributes to the database

<u>Step 10</u>: Unbatches the ship cargo into the assigned number containers to unload

<u>Step 11</u>: Updates counters for loaded and unloaded containers at the dock

<u>Step 12</u>: Holds the ship until the assigned values of the dock's loaded and unloaded container counters are reached and then signals ship to depart.

<u>Step 13</u>: Updates ship and dock attributes

<u>Step 14</u>: Releases the dock space

Step 15: Updates ship's status and time stamp

Step 16: Writes ship information to database

Step 17: Routes the ship from the dock to the departure station

Step 18: A ships enters the departure station through this module

Step 19: Updates the ship's status and other attributes

Step 20: Writes ship information to database

Step 21: Removes ship from the model

Path B (for containers)

Step 1a: Assigns the container an entity type called crate

Step 2a: Sends containers to the destination assignment logic.

Step 3a: Assigns each container a destination ID. There are 5 destinations; train terminal, long distance trucking center, and 3 local trucking centers. After assigning the destination IDs containers are sent to the incoming container logic

Step 4a: Containers arrive at the incoming container logic through this block.

Step 5a: Holds container and checks that the assigned crane for the dock is available before allowing it to proceed

Step 6a: Updates the container unloading counter

Step 7a: Unloads container from ship

Step 8a: Counts the number of unloaded containers

Step 9a: Updates container's status and other attributes

Step 10a: Writes container information to the database

Step 11a: Writes other container information to a different database

Step 12a: Routes container to the Dockside Release station

Step 13a: Containers arrive at the Dockside Release station through this module

Step 14a: Requests for a forklift transporter

Step 15a: Applies a forklift pickup delay

Step 16a: Routes containers to inspections

Logic for sub-model shown in Figure 12

There is only one path for containers in this logic

Step 1: Containers enter inspections through this block

Step 2: Updates container's status and other attributes

Step 3: Writes container information to database

Step 4: Applies a forklift dropoff delay

Step 5: Checks whether container was from seaside or rail landside

Step 6: Releases the appropriate forklift based on the container's origin as checked in previous step

Step 7: Stores container in inspections storage

Step 8: Container seizes an inspector resource

Step 9: Applies an inspection delay

Step 10: Container releases the inspector resource

Step 11: Decides whether container would need further inspections. If further inspection is decided then go to Step 12 otherwise go to Step 15

Step 12: Container seizes an inspector resource

Step 13: Applies an inspection delay

Step 14: Container releases the inspector resource

Step 15: Fails a defined percentage of containers. The failed containers are destroyed from the model. The containers assigned a "pass" proceed to Step 16

Step 16: Checks if container is from landside or seaside. If the container is from the seaside then it may be sent to bonded storage otherwise container would proceed to customs. The logic to bonded storage or customs is the same

Step 17: If container is from seaside then send a defined percentage to bonded storage.

Step 18: Request a forklift transporter to next destination

Step 19: Applies a forklift pickup delay

Step 20: Removes container from inspections storage and updates the storage counter

Step 21: Updates the container's status and other attributes.

Step 22: Writes container information to database

Step 23: Transports container to the assigned destination

Logic for sub-model shown in Figure 13

There is only one path for containers in this logic

Step 1: The containers enter bonded storage through this module

Step 2: Updates the container's status and other attributes

Step 3: Writes container information to database

Step 4: Stores in container in bonded storage and updates the storage counter

Step 5: Applies a storage delay

Step 6: Requests a forklift transporter

Step 7: Applies a forklift pickup delay

Step 8: Removes the container from bonded storage and updates the storage counter

Step 9: Updates the container's status and other attributes

Step 10: Writes container information to the database

Step 11: Transports the container to customs

Logic for sub-model shown in Figure 14

There is only one path for containers in this logic

Step 1: This is the entry point for containers to customs in the model

Step 2: Updates the container's status and other attributes

Step 3: Writes container information to database

Step 4: Stores container in customs and updates the storage counter

Step 5: Container seizes, delays, and releases a customs agent resource

Step 6: Checks if the container is from landside or seaside. Containers from landside will be sent to the outgoing dockside. Containers from seaside will be sent to land departure terminal unless there is lack of temporary storage space then they will be sent to domestic storage.

Step 7: Requests a forklift transporter

Step 8: Applies a forklift pickup time delay

Step 9: Removes container from storage and updates the storage counter

Step 10: Applies to only containers from seaside. Checks for the availability of temporary storage space at the land departure terminal.

Step 11: Updates the container's status and other attributes

Step 12: Writes container information to database

Step 13: Transports container to one of the three possible destinations.

Logic for sub-model shown in Figure 15

Step 1: The containers enter domestic storage through this module

Step 2: Updates the container's status and other attributes

Step 3: Writes container information to database

Step 4: Stores container in domestic storage and updates the storage counter

Step 5: Holds the container until there is space at the temporary storage facility at the land departure terminal.

Step 6: Requests a forklift transporter

Step 7: Applies a forklift pickup delay

Step 8: Removes the container from domestic storage and updates the storage counter

Step 9: Updates the container's status and other attributes

Step 10: Writes container information to the database

Step 11: Transports the container to land departure terminal

Logic for sub-model shown in Figure 16

Containers entering the land departure terminal all use similar logic.

Step 1: This is the entry point for containers into the land departure terminal

Step 2: Updates the container's status and other attributes

Step 3: Writes container information to database

Step 4: Stores container in temporary truck transfer storage and updates the storage counter

Step 5: Check to see if the container is assigned to the rail terminal. If so then go to Step 6. Otherwise go to Step 5a.

Step 5a: Check if the container is assigned to a long distance truck distribution center. If so then go to Step 6. Otherwise go to Step 5b.

Step 5b: Assigns container a destination IDs from one of the three local truck distribution centers. Then go to Step 6.

Step 6: Request for a truck to pickup container.

Step 7: Loads container onto truck using a flatbed crane

Step 8: Removes container from temporary truck transfer storage and updates the storage counter

Step 9: Updates the container's status and other attributes

Step 10: Writes container information to the database

Step 12: Updates some counters and container attributes

Step 13: Transports container to the designated destination

Logic for sub-model shown in Figure 17

Step 1: This is the entry point for containers received from both the rail terminal and the highway trucks.

Step 2: Checks if the container is from the rail terminal or highway truck.

Step 3: If the container is from highway truck then assign it container attributes such as a unique ID and status.

Step 4: Assign the outgoing dock number as an attribute

Step 5: Writes container information to the database

Step 6: Stores container in temporary truck storage and updates the storage counter

Step 7: Unload container from truck

Step 8: Check the origin of the container

Step 9: If it is from the rail terminal then free the truck transporter.

Step 10: Updates container's status and other attributes

Step 11: Check the number of containers assigned to the outgoing dock. If it is equal to the assigned number of containers to be loaded by the berthed ship then route to Receive Storage otherwise send container to inspections

Step 12: Requests a forklift transporter

Step 13: Applies a forklift pickup delay

Step 14: Removes container from temporary truck storage and updates the storage counter.

Step 15: Updates the container's status and other attributes

Step 16: Writes container information to the database

Step 17: Route the container to the assigned destination.

Logic for sub-model shown in Figure 18

There is one path for containers in this logic

Step 1: Containers enter the outgoing dockside through this module

Step 2: Routes containers to their assigned docks

Step 3: This block holds the IDs of dockside stations

Step 4: Updates the container's status and other attributes

Step 5: Writes container information to the database

Step 6: Checks to ensure that the loading crane on the dock is free and the number of loaded containers is less that the assigned number for ship to load.

Step 7: Updates the loaded container counter

Step 8: Loads container onto ship

Step 9: Updates the container's status and other attributes

Step 10: Writes container information to the database

Step 11: Records the number of loaded containers

Step 12: Permanently batches the loaded containers into one cargo.

Step 13: Updates some counters related to the dockside activities

Step 14: Applies a ship routing time from dock to departure station

Step 15: Updates the container's status and other attributes

Step 16: Writes container information to the database

Step 17: Removes ship and containers from model.

REFERENCES

Andrews S., Murphy F., Wang X., and Welch S. (1996). *Modeling crude oil lightering in Delaware Bay.* Interfaces 26, No. 6, pp 68-78.

Asperen, E. van, Dekker, R., Polman, M., de Swaan Arons, H., and Waltman, L.R., (2003). *Arrival Processes for Vessels in a Port Simulation*. Research Paper ERS-2003-067-LIS Revision, Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus Uni.

Baunach G., Wibberley E., and Wood B. (1985). *Simulation of coal transshipment terminal: Batam Island, Indonesia.* Mathematics and Computers in Simulation 27, pp 115-120.

Carey M. (1994). *A model, algorithm and strategy for train pathing, with choice of lines, platforms, and routes.* Transportation Research 28 (B), 333 – 353.

Chen B. and Harker P. (1990). *Two moments estimation of the delay on single-track rail lines with scheduled traffic.* Transportation Science, Vol. 24, No. 4, pp 261-275.

Cheng Y. (1998). *Hybrid simulation for resolving resource conflicts in train traffic rescheduling.* Computers in Industry 35 (3), 233 – 246

Dalal M., and Jensen L. (2001). *Simulation modeling at union pacific railroad*. Proceedings of Winter Simulation Conference 2001, 1048-1055.

Dessouky, M., and Leachman, C. (1995). *Simulation Modeling Methodology for Analyzing Large Complex Rail Networks.* Simulation, v. 65 n. 2pp. 131 – 142.

Erie S. (2004). *Globalizing L.A. : Trade, Infrastructure and Regional Development*. Stanford 2004.

Fioroni, M. and Botter, R. (2005). *An Algorithm for Train Movement Simulation*. Proceedings of the ICIL'2005, p.113-120, Montevideo, Uruguay.

Frank O. (1966). *Two-way traffic in single line of railway.* Operations Research, Vol. 14 pp 801-811.

Gambardella L., Rizzoli A., Fornara N., Alessandrini A., and Epifani C. (2000). *The use of simulation in the socio-economical evaluation of the management of an intermodal terminal*. In Proceedings of the International Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, Portofino - Italy, pages 71–76. Society for Computer Simulation, 2000.

Greenberg et al. (1988). *Predicting dispatching delays on a low speed, single track railroad.* Transportation Science, Vol. 22, No. 1, pp31-38.

Heyden W. and Ottjes J. (1985). *A decision support system for the planning of workload on a grain terminal.* Decision Support Systems, 1, 293-297.

Komaya K. (1991). *A new simulation method and its application to knowledge-based systems for railway scheduling.* In Proceedings of the IEEE / ASME Joint Railway Conference, New York, NY, May 1991, St. Louis, MO, 59 – 66.

Kozan E. and Preston P. (1999). *Genetic algorithms to schedule container transfers at multimodal terminals.* International Transactions in Operational Research, 6: 311 – 329.

Levinson M. (2006). *The Box: How the Shipping Container Made the World Smaller And the World Economy Bigger.* Princeton, N.J.: Princeton University Press, 2006.

Lu, Q., Dessouky M. and Leachman, R.C. (2004). *Modeling train movements through complex rail networks*, ACM Transactions on Modeling and Computer Simulation, 14(1) 48-75.

Mastrolilli M, Fornara N, Gambardella L M, Rizzoli A E, and Zaffalon M. (1998). *Simulation for policy evaluation, planning and decision support in an intermodal container terminal*. In: Merkuryev Y, Bruzzone A, Novitsky L (eds) Proceedings of the International Workshop 'Modeling and Simulation within a Maritime Environment', Sep 6–8, pp 33–38. Society for Computer Simulation International, Riga, Latvia.

Ottjes J., Hengst S., and Tutuarima W. (1994). *A simulation model of a sailing container terminal service in the port of Rotterdam.* In proceedings of the 1994 Conference on Modeling and Simulation. Guasch and Huber (Eds.), 876-880.

Petersen E. (1974). *Over-the-road transit time for a single track railway.* Transportation Science, Vol. 8, pp 65-74.

Petersen E. and Taylor A. (1982). *A structured model for rail line simulation and optimization.* Transportation Science, Vol. 16, pp 192-206.

Philips O. (1976). *Optimization models for a crude oil storage export system.* Ph.D. thesis. Penn. State University.

Rizzoli A., Fornara N., and Gambardella L. (2002). *A simulation tool for combined rail/road transport in intermodal terminals*. Mathematics and Computers in Simulation, 59(1- 3):57–71, 2002

Strom, H. (1972). *Containerization: A Pandora's Box in Reverse?*. Transportation Journal, 12(2), 46-57.

U.S. Department of Transportation (USDOT), Federal Highway Administration (FHWA) (2008). *Freight Story 2008*. Washington, DC. Available at http://ops.fhwa.dot.gov/freight/freight_analysis/freight_story, as of Apr. 7, 2009

U.S. Department of Transportation (USDOT), Maritime Administration (MARAD). (2005). *Report to Congress on the Performance of Ports and the Intermodal System*. Washington, DC. Available at http://marad.dot.gov/documents/Rpt_to_Congress- Perf_Ports_Intermodal_Sys-June2005.pdf, as of Apr. 7, 1009.

U.S. Department of Transportation, Research and Innovative Technology Administration, and Bureau of Transportation Statistics, (2010). *National Transportation Statistics* http://www.bts.gov/publications/national_transportation_statistics/

U.S. Department of Transportation, Research and Innovative Technology Administration, and Bureau of Transportation Statistics, (2009). *America's Container Ports: Freight Hubs That Connect Our Nation to Global Markets* Washington, DC.

Wall T. (2010). *A federated simulation approach to modeling port and roadway operations.* http://hdl.handle.net/1853/33928

Wilson W. and DeVuyst E. (2007). *Optimization models of container shipments in North America: Spatial competition and projections (methodology).* A draft report for review submitted on December 31, 2007.

Zaffalon M, Rizzoli A.E., Gambardella LM, and Mastrolilli M. (1998). *Resource allocation and scheduling of operations in an intermodal terminal*. In: ESS98, 10th European Simulation Symposium and Exhibition, Simulation in Industry, Nottingham, UK, pp. 520–528, October, 1998