

An Object-Oriented Architecture for the Simulation of Networks of Cargo Terminal Operations

Reejo Mathew

rmath002@odu.edu

James F. Leathrum, Jr.

leathrum@ece.odu.edu

Saurav Mazumdar

smazu001@odu.edu

Taylor Frith

taylor.frith@gmail.com

Department of Electrical
and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Joseph Joines

Military Surface Deployment
and Distribution Command
Transportation Engineering Agency
(SDDC-TEA)
Newport News, VA 23606
joseph.joines@tea.army.mil

A new object-oriented architecture to simulate a network of cargo terminals is presented. The architecture simulates the complete flow of military cargo at an entity level, from multiple U.S.-based installations to a set of destinations within a theater of operations located anywhere in the world. The architecture deals with resource and infrastructure allocations and competition within each terminal in the network, as well as between the terminals, to identify conflicts that may arise. The architecture is a hierarchical, nodal model moving individual pieces of cargo through a network of cargo terminals, and within an individual cargo terminal through a network of physical processing areas. The architecture is capable of simulating cargo terminals around the world under various scenario conditions. It is impractical and expensive to conduct actual exercises to study the efficiency of a deployment. Simulating this process provides a viable alternative for testing these strategies and for evaluating technologies. It also enables an analysis of the simulation results for improved efficiency and throughput within the confines of resources and infrastructure availability. The new architecture is described as well as an example case study to demonstrate the capabilities of the architecture. The architecture focuses on resources and infrastructure within a cargo terminal, and on transportation resources between cargo terminals.

Keywords: Network of cargo terminals, nodal networks, port simulation, cargo logistics

1. Introduction

A long logistics supply chain exists in the flow of military cargo from installations in the continental United States to a theater of operations outside of the continental United States. Historically, simulation of this supply chain has involved various simulations to handle different components such as the individual nodes (military installations, seaports, airports) and the individual transportation legs (highway, rail, sea, air). To address the end-to-end nature of the problem, results from different simulations are used as input for other simulations. Alternatively, an effort to create a framework around the existing simulations to allow sharing of data has been conducted [1, 2, 3]. The work presented here resulted from one of the simulations used in the end-to-end process, *PORTSIM* [4, 5, 6], which identified the competition of transports shuttling cargo between cargo terminals as a critical resource.

An end-to-end simulation architecture is presented here that focuses on the availability of resources between and within cargo terminals. This architecture is not concerned with the transportation infrastructure between cargo terminals except as a delay factor. This architectural model was designed with the following requirements:

- *Operation of a network of cargo terminals:* The architecture supports the operation of multiple cargo terminals connected in a network to enable the simulation of an end-to-end (origin to destination) cargo flow.
- *Reconfigurable cargo terminals:* An individual cargo terminal has the ability to be initialized independent of other cargo terminals. The cargo terminals are also reconfigurable; that is, the infrastructure and the resources in a cargo terminal as well as the sequence of cargo flow through the areas in a single terminal are changeable. This is important since the military regularly does not receive full access to a seaport as ongoing commercial shipping activities are maintained.
- *Concurrent deployment and redeployment:* Previous efforts focused on the flow of cargo to the theater of operations (deployment). However, for sustainment, cargo will eventually be moving in a bidirectional manner. To support this, the architecture supports concurrent inbound and outbound flow of cargo within a single cargo terminal and bidirectional flow between terminals.
- *Simulation of one million pieces of cargo:* The architecture can support the simulation of the complete flow of one million pieces of cargo (with

the cargo being modeled as individual pieces) within a reasonable execution time.

While past analyses have studied individual cargo terminals and transportation network infrastructure, little effort has been made to study the complete end-to-end process at the cargo entity level, including detailed processing within the cargo terminals. This effort presents a complete model with a high level of fidelity in operational data and processes, while still maintaining high performance and scalability in problem size. The model develops a high-fidelity model for the cargo terminals modeling all internal resources (container handlers, stevedores, drivers, etc.) and infrastructure (cargo loading/offloading space, staging space, etc.). Internodal connections focus on modeling the impact of transportation resources (trucks, trains, ships, etc.) without worrying about the transportation infrastructure (roads, rail, waterways, etc.). The impact of transportation infrastructure is modeled in the form of time delays between nodes. This paper will present the network architecture supporting the simulation. It will then illustrate how the architecture enables the simulation of a unique military scenario that includes new technology (as yet untested) in a scenario involving approximately one million pieces of cargo.

The architecture described here is object oriented. The different cargo terminals in the end-to-end network simulation are defined as instances of a single abstract class. The same is true for individual terminal areas within a cargo terminal. This supports abstraction and reuse of individual model components, thereby providing a reusable nodal network. The simulation can be used to model any airport or seaport in the world, operating in any military operational mode. The nodal structure of the simulation supports connecting diverse terminals, enabling simulation of a complete military logistics chain.

2. Related Work

2.1 Military Transportation Logistics Simulations

Transportation logistics planning for military operations is used to improve the flow of military cargo through a cargo transportation network. The network is comprised of a set of cargo terminals interconnected by a transportation infrastructure. The cargo terminals include points of origin and destination, intermediate transfer points for transportation mode changes, and/or points of intermediate storage. The *Port Simulation Model (PORTSIM)* [4, 5, 6] addresses two modes of military cargo operations: seaports of debarkation (SPOD) and seaports of embarkation (SPOE). The SPOE mode deals with the arrival of cargo at the port via rail and highway transports, staging of cargo,

and loading of a ship with cargo. The SPOD mode focuses on the activities of unloading cargo from a ship, staging, parking and inspection of the cargo, and clearing the cargo from the port using rail and highway transports. The *Configurable Port Simulation (CPortS)* [7–10] supports *PORTSIM* by providing the SPOD capability and is the foundation of the work presented here. *PORTSIM* suffers from mutually exclusive SPOE and SPOD processes and fixed cargo flow. Other examples of cargo terminal models are *TRANSCAP* [11, 12], which models offloading at installations, and *TLoads* [13], which attempts to assess the capability of tactical and sea-based distribution systems.

Past work has examined modeling the actual transportation segments of the defense transportation system. Such efforts include *ELIST* [14], which models theater rail and highway infrastructure, and *MIDAS* [15] and *JFAST* [16], which model the strategic lift segments. However, none are stochastic models. These models study the restrictions of the transportation infrastructure rather than the cargo terminals (nodes in a network). They also aggregate the cargo, often dealing with it as quantities in terms of weight, area, or number of pieces rather than individual items. The *AMP* model [17] acts as a shell to interconnect these models into an end-to-end logistics model, but at a low level of fidelity, again focusing more on the links than the nodes within the network.

2.2 Commercial Transportation Logistics Simulations

Commercial cargo terminal simulation models that are currently described in the literature [18–31] differ widely in their objectives, complexity, level of detail, and operational factors taken into consideration. These differences are due to the variation in the questions that a particular model attempts to answer as well as the model's fidelity. Most of these models are limited as they are designed around the commercial activities of a specific cargo terminal with a limited level of detail concerning the operational behavior of the terminal. Most of the past research puts forward the argument that a model can be extended to include improved levels of detail.

Object-oriented modeling is applied in a wide variety of transportation domains to simulate real-world problems and to suggest solutions. Object-oriented approaches toward simulating complex transportation systems are found in application areas such as urban transportation [32, 33], rail [34], and road traffic [35, 36, 37]. These simulations demonstrate the benefits of object-oriented modeling in the transportation domain, focusing on the physical interconnect structure and relationships between various model components,

including the integration of GIS information. The work presented in this paper extends this work into a model focusing on the intermodal exchange points in the transportation network and their interaction rather than the physical transportation infrastructure.

3. Problem Description

During a military operation, military units and their corresponding equipment are continually deployed and redeployed between their home station (origin) and the theater of operations (destination). This results in a bidirectional flow of cargo and resources within the network, with cargo competing for a limited number of resources within a cargo terminal and transports between cargo terminals.

Modeling the processes required at various nodes in the transportation network, while presenting its own difficulties, is relatively straightforward in that the processes are static in nature from one scenario to another. The difficulty and increased complexity is in modeling the interconnection of the nodes in the network to allow a configurable flow of cargo and transports for each unique scenario. The interconnection problem appears at both the macro and micro levels. Each of these levels requires modeling the internal processes for handling cargo.

At the macro level, the flow of cargo between cargo terminals must be modeled. Cargo is transported in a multimodal fashion, with different types of cargo moving by different modes based on cargo priority and transport capacity. A typical flow of cargo moving to a theater of operations starts at a set of installations in the United States. Cargo is then moved by highway and rail to seaports and airports of embarkation where the cargo is staged and loaded onto airplanes and ships accordingly. Cargo is then transported to seaports or airports of debarkation near the theater of operations where the cargo is offloaded, staged, and then loaded for transport by highway and/or rail to either an intermediate staging base or to its final destination. On each leg of the transportation network, transports are cycled to keep the cargo moving. These transports are often a limiting factor in the throughput of the network.

At the micro level, the cargo flow within specific terminals may differ greatly from one another. For example, military operations utilizing convoys may be staged outside of the gate at the port of Ad Dammam, in Saudi Arabia, while at the port of Livorno, in Italy, the convoy construction area is located within the port property. This reordering of the terminal areas can create an impact on the operating capabilities of each individual cargo terminal.

At a micro level, this architecture is designed to

support the concurrent and simultaneous SPOE and SPOD operations within a cargo terminal. It supports any configuration of a specific cargo terminal and is suitable for handling the point-of-origin cargo terminal, seaports, airports, and the destination cargo terminal. At a macro level, the architecture supports the interconnection between cargo terminals to facilitate the end-to-end movement of cargo. It is important to note that this architecture is focused on modeling and analyzing cargo terminal capabilities at the cargo transportation network nodes, including the resource requirements to transport cargo between cargo terminals within the network. The models described in the previous section (*ELIST*, *JFAST*, and *MIDAS*) should be used to model the strategic lift and theater transportation infrastructure segments, and then the appropriate delays between segments should be used as input to the new simulation.

For rapid deployment of military forces in a theater of operations within 96 hours, the simulation should execute within a reasonable amount of time. The Military Surface Deployment and Distribution Command–Transportation Engineering Agency (SDDC-TEA) identified the following performance requirements to execute within a “reasonable time”:

- Operate on a PC, minimally 2 GHz with 2 GB of RAM [38, 39],
- Simulate an operation with 100,000 pieces of cargo [38, 39],
- Handle at least 30 multiple runs on the same scenario for statistically meaningful results [38, 39, 40], and
- Execute experiments involving on the order of 150 variations [38, 39].

The term “reasonable time” is difficult to quantify and changes as the perception of machine power changes. In an effort to put a limit on this, consider a worst case scenario, where decisions must be made in days. Assume that in crisis action mode, an upper limit on the amount of cargo to be simulated is 25,000 pieces. Allowing for time to define the experiments and evaluate the results prior to making a decision, an upper limit of 24 hours execution time seems reasonable to meet planning requirements within 96 hours. In this case, 150 variations of the scenario with each variation requiring 30 runs to gather statistics requires something on the order of 4,500 executed runs. Therefore, the simulation should complete a single run in 19.2 seconds on average. Larger operations on the order of 500,000 pieces of cargo or more would involve a longer analysis time and thus would be expected to require longer execution run times; although distributing the network would reduce computing time.

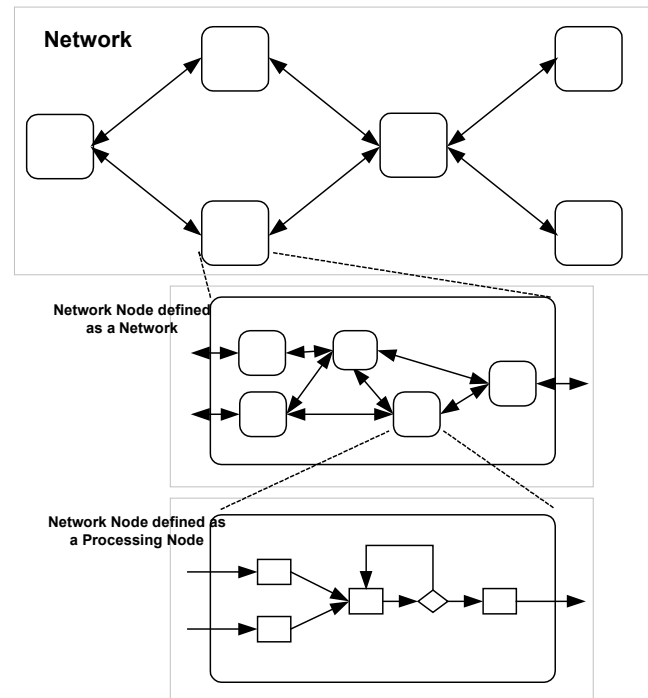


Figure 1. Hierarchical network structure with processes defined within nodes

4. Architecture Description

Our architecture defines the nodal and interconnection structure for a cargo transportation network and facilitates the management of moving entities between the different nodes within the network. Our architecture is hierarchical in nature and allows an individual node within a network to be defined as a network itself. In this manner, the same architecture can support the management of the total cargo transportation network as well as the network internal to a single cargo terminal [41].

The architecture is defined as a network of nodes, with each node defined by a process or by another network, thus allowing a hierarchical structure as shown in Figure 1. (All objects are represented by rounded boxes in this document, while methods of individual objects are represented by ovals.) Nodes represent cargo terminals within the cargo transportation network as well as the individual terminal areas within a single cargo terminal. The architecture also supports grouping together similar nodes into sets for management and selection purposes. A particular node can be selected from a group of similar nodes based on its characteristics, capacity, availability, and its ability to handle a particular type of entity.

The architecture also supports the routing of entities through the network. A route is defined by a starting

node and a destination node with a sequence of nodes to be traversed en route. Once a node that can handle a given entity is identified, the entity must request space in the selected node before transiting to the node. This is important to avoid infinite queuing in the interconnection segments and to provide a level of flow control, thereby assisting in the avoidance of deadlock. Military operations statically define the sequence of cargo terminals that cargo traverses, while the sequence of terminal areas visited within a cargo terminal is selected based on available capacity and capability. All these issues are described in depth in the following subsections.

4.1 Nodes

The architecture is defined around a nodal network concept. A network is partitioned into nodes and the interconnections or edges between the nodes. Each individual node has a similar interface with a set of predefined entry and exit points. Entry points accept loaded or empty transports from the interconnection network. Exit points send loaded or empty transports out of the node to the interconnection network. Each node also has a request for space capability that inhibits the flow of cargo should insufficient space or resources exist to handle it. The request for space capability requires the requesting node to wait until the space is available at the next node before proceeding. Each node defines either the underlying network for network nodes or the underlying processes for process nodes.

The architectural model is a shell around the cargo processing model, which is unique to each cargo terminal as well as to each area within the terminal. The architectural model acts as an interface for the cargo processing model with the outside world. At the top level, the flow of cargo between cargo terminals is modeled (where cargo terminals are modeled as nodes in the network). The cargo transportation network is comprised of a set of cargo terminals interconnected by transportation infrastructure. This transit interconnection interface is modeled using the *TransitInterface* class. The *TransitInterface* class is used as a common interface between various objects that are used in the simulation. Each cargo terminal used in the simulation as well as any object that represents infrastructure within the cargo terminal inherits the *TransitInterface* class. This allows the interconnection of cargo terminals to model an end-to-end flow of cargo with the *ExitPoint* of one node connected to the *EntryPoint* of the next node within the network. The same *TransitInterface* class is used to model the transit of resources and transports through a single cargo terminal.

4.1.1 CargoTerminal Class

The *CargoTerminal* class provides a wrapper around a single terminal. The *CargoTerminal* class defines the input and output operations to and from the terminal, as well as the internal characteristics of the terminal. Functionally, the primary purpose of the *CargoTerminal* class is to provide interfaces between the cargo terminal and the external cargo transport infrastructure. This class also provides an interface between the network of cargo terminals and the network of terminal areas within a single cargo terminal. These interfaces are broken into the available transport modes of sea, highway, rail, and air. Each transport mode has an associated entry and exit operation. The *EntryPoints* and *ExitPoints* pass off transports, which can be ships, trains, and trucks, each of which may be carrying one or more pieces of cargo. The *EntryPoint* is called by the external transportation infrastructure for transports arriving at the cargo terminal. The *ExitPoint* is called internally from the cargo terminal to handle transports leaving the terminal. This results in a call to the external transportation infrastructure to pass the transport off to the next node in the network. Internal to the cargo terminal, the *CargoTerminal* class defines the attributes that identify the actual operations within the terminal. This includes identifying operations (cargo terminal areas), terminal resources, and the infrastructure required to move within the terminal. The cargo terminal areas are physical areas where cargo is handled and stored. Terminal resources are those resources, such as container handlers and forklifts used

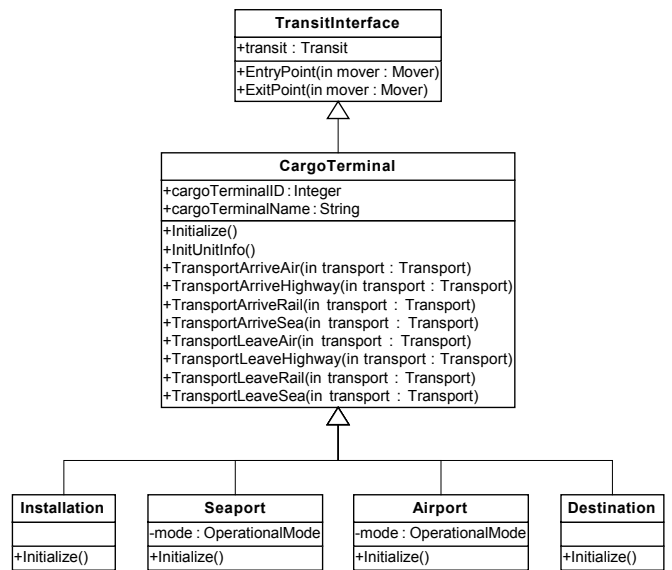


Figure 2. Cargo terminals

for cargo operations, that are shared between multiple terminal area types. (Other resources may be allocated directly to a cargo terminal area.) The different cargo terminals are derived from the *CargoTerminal* class as shown in Figure 2.

4.1.2 CargoTerminalArea Class

Each cargo terminal is further segmented into nodes to describe the internal infrastructure. A cargo terminal is comprised of areas within the terminal where cargo operations are performed. These areas include but are not limited to:

- *Loading/Offloading*: Areas where cargo is loaded onto transports to leave the terminal or offloaded from transports arriving in the terminal;
- *Staging*: Areas where cargo is stored while waiting for appropriate transports to move the cargo out of the terminal—other operations such as cargo inspection may also be performed here;
- *Transport processing*: Frequently, transports will require storage or assembly into a larger or smaller transport set—examples are interchange yards for trains and convoy construction areas for highway transports;
- *Transit conflict*: Terminal operation areas can be used to model conflicts in the transportation network within a terminal—examples could be road intersections or rail crossings.

The different operational areas within a cargo terminal are modeled using the *CargoTerminalArea* class. It is a parent class to be inherited by the individual cargo terminal areas. The structure of this class is shown in Figure 3. The *CargoTerminalArea* class has various attributes to identify itself and to establish its association with a parent cargo terminal. It also has methods to signal the arrival of cargo and resources into the cargo terminal area as well as methods to reserve, allocate, and release space in the cargo terminal area. It is intended that the methods in this class be overridden with any specific code necessary

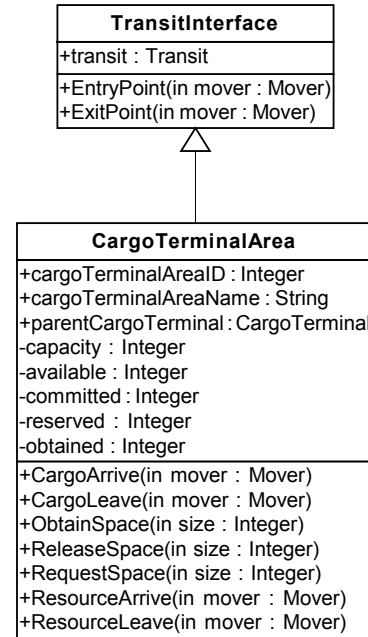


Figure 3. CargoTerminalArea class

within the cargo terminal areas. However, there is some basic bookkeeping done in the *CargoArrive* and *CargoLeave* methods. This bookkeeping enables the tracking of any cargo stranded in the cargo terminal area upon completion of the simulation. An SPOD, for example, can be segmented into the following sets of process nodes (cargo terminal areas):

- *Cargo reception*: Anchorage areas, berth areas;
- *Cargo staging*: Staging areas; and
- *Cargo clearance*: Loading areas, convoy construction areas and gate areas for highway cargo, rail spur areas and interchange yard areas for rail cargo, helicopter reassembly areas, helicopter staging areas, and helicopter takeoff areas for helicopter cargo.

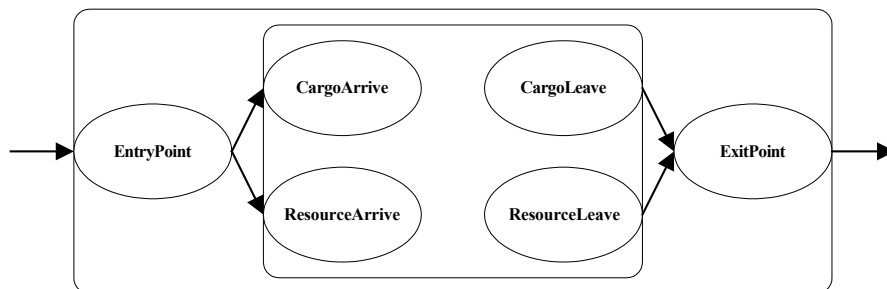


Figure 4. Cargo terminal area entry and exit

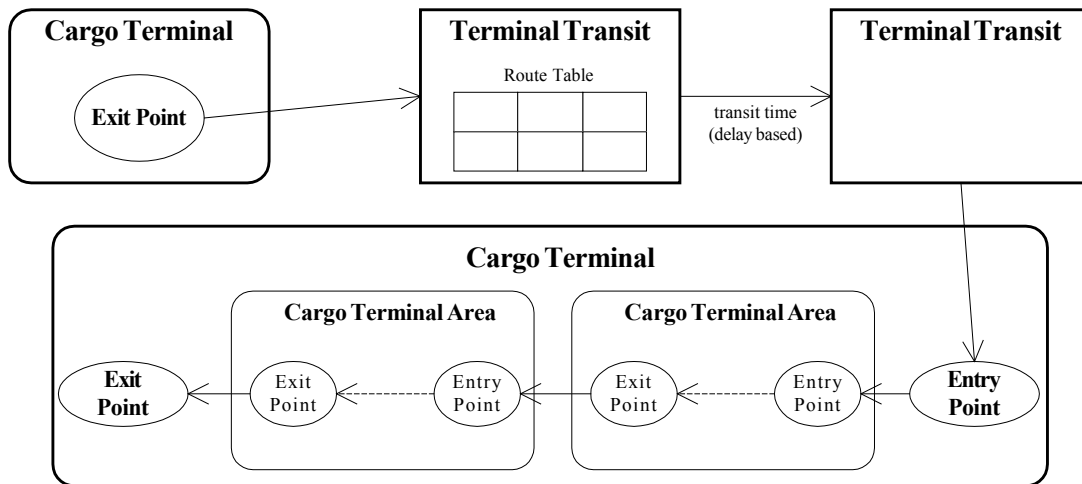


Figure 5. Terminal interconnection network

Both transports and resources (referred to as movers) arriving with cargo, as well as arriving empty to carry cargo out of the cargo terminal area, enter the area through the *EntryPoint*. They are then partitioned with those carrying cargo calling the *CargoArrive* method to enter the area and those that are empty calling the *ResourceArrive* method. Corresponding methods for movers leaving the area are *CargoLeave* and *ResourceLeave*, which each call the common *ExitPoint* method. The extra wrapping around the cargo terminal area by the *TransitInterface* provides a single point of entry and exit through the area. Figure 4 shows the relationship between these methods.

In order to allow finite capacities within the cargo terminal areas and still avoid deadlock, the model requires space to be available in a destination cargo terminal area prior to securing the resources necessary to move the cargo to that area. Space capacity may be viewed as a size (such as square feet or length) or as a quantity (such as number of pieces of cargo). When empty, all space is made available. When a request for space is made, the request must wait until appropriate space is available before the request is serviced. Requests are served in first-come-first-serve order. When a request for space is serviced, the space is made unavailable. The space is defined as *reserved* until the requesting entity arrives. When the entity arrives at the cargo terminal area, the space allocation status changes from *reserved* to *obtained* and returns to *available* when the entity leaves. This triggers any outstanding requests for space that may be satisfied by the newly available space.

4.2 Sets of Nodes

A set of nodes identifies nodes with like type, characteristics, and capabilities. The nodes are grouped together for management and selection purposes. This supports the matching of entity characteristics to specific nodes that can handle the entity. At a macro level, cargo terminals of the same type are grouped together even though they may be far apart geographically. For example, all seaports within a scenario are grouped together, while all airports fall within another set. At a micro level, cargo terminal areas of like type within a cargo terminal are grouped together. For example, at a seaport, berths, staging areas, and rail interchange yards are all grouped under different sets.

4.3 Network Interconnection

Each node within the network is unique such that transit times between nodes vary and even the sequence in which the nodes are traversed varies from terminal to terminal. To accommodate this, a layered communication and interconnection model was developed and is shown in Figure 5. The programmable *Transit* class handles the transit of entities (transports, resources, and cargo) within the cargo transportation network. Each network node has a *Transit* object associated with it. The *Transit* object interacts with the nodes within the network via the *TransitInterface* class which each network node inherits. At the macro level, the *Transit* object associated with a cargo terminal (called *TerminalTransit* object) is responsible for all transit and routing between cargo

terminals. The *TerminalTransit* object interacts with its associated cargo terminal via the *TransitInterface* class methods *ExitPoint* and *EntryPoint*. The *ExitPoint* method is the only place where the *TerminalTransit* is referenced when a transport is required to leave the cargo terminal. The *TerminalTransit* of the destination cargo terminal can only call the *EntryPoint* method of the destination cargo terminal to allow the entry of the transport.

At the micro level, another *Transit* object is responsible for all routing and transit within a single cargo terminal, between cargo terminal areas of the terminal. Once an entity enters a cargo terminal through the *EntryPoint*, it is processed through the various cargo terminal areas following a programmable sequence. This sequence can differ by entity type and usually depends on the processing required for a particular type of cargo and the transport that is needed to carry the cargo out of the cargo terminal. In a particular cargo terminal area, cargo and transports enter through the *EntryPoint* and are processed accordingly. Cargo and transports then leave the cargo terminal area through the *ExitPoint* and are transported to the next cargo terminal area in its sequence through the *EntryPoint* of that cargo terminal area.

The *Transit* class is comprised of two components: a routing component and a transit time component. The routing component enables a transport to move from its current location to the next appropriate location in an attempt to reach a final destination. The transit time component allows for a processing time to be applied to the movement between the two locations.

4.3.1 Routing

Military practice ensures that each piece of cargo has an associated route. Route assignments are based on military unit identifiers, cargo type transport mode, destination, and required delivery date. A route is defined by a starting node, a destination node, and a sequence of nodes traversed en route within the network. The routing component within the *Transit* class is implemented with the *RouteTable*, which is programmable to define unique routes for different scenarios. The *RouteTable* determines the "next node" in the route for the entity to transit to, given its current location and its final destination. The *Transit* class also supports the scheduling of events (termed *programmable events*) that enable route changes during the simulation.

4.3.2 Transit Time

To support transit times between nodes, a programmable random variate model is employed [42]. The model supports complex distributions including dependence between events. The capabilities of this model include the following:

- *Single mode distributions*: A wide variety of distributions is available including exponential, triangular, beta, Weibull and constant.
- *Multimodal distributions*: A multimodal distribution defines a random variable by multiple modes, with each mode including a distribution and probability of occurrence.
- *Ordered distributions*: An ordered distribution guarantees that subsequent events cannot get out of order by preventing the next event from completing prior to the current event. In this way, transits can prevent passing en route.
- *Blocking distributions*: A blocking distribution is a multimodal distribution where once a given mode is entered, all following calls will remain in that mode for a period defined by another distribution. This can be used to model situations such as the transit being blocked by a rail crossing.

The complex distributions available in this model allow the times taken to transit between nodes to be modeled without a high level of detail. However, should more detail be desired, other models such as *ELIST* or *JFAST* could be utilized to replace the transit time computation. Alternatively, the random variate model inside the *Transit* class can be replaced by the appropriate model, thus keeping the routes between the nodes.

The *TransitTimeTable* implements the transit time component within the *Transit* class. It defines the time required to traverse between a pair of nodes where such travel is direct (no other nodes are visited in between). It also keeps track of processes defining the transit time between a pair of nodes in the network.

Transit times between terminals are based on a 24-hour clock, while transit times within a terminal (between cargo terminal areas) can be adjusted to the work shift of a particular cargo terminal (anything less than 24 hours per day). Depending upon a given work shift, if a transit time crosses over into the next workday, the transit of that particular entity can be scheduled at the start of the next workday. Similarly, if the time required for an activity within the terminal crosses over into the next workday, the activity is scheduled at the start of the next workday. This enables the accurate modeling of activities that might be scheduled based on whether or not they can be completed during the current workday.

5. Example Scenario

An example simulation study is detailed to demonstrate how the architecture is used to model an end-to-end military deployment. This study is an expanded version of the scenario described and analyzed by Leathrum et al. [43] to include the complete military deployment process from installation to destination. It is important to state here that this scenario is completely fictional, is not based on any past or planned U.S. military operations, and is used purely as an example. This ensures that no security issues are violated.

5.1 Scenario Description

In the scenario, it is necessary to deploy a U.S. mechanized division from its home base in Hawaii to south-central South Korea. The division proceeds by convoy from the installation to two terminals at the port of Pearl Harbor: the FISC Area and West Loch. The division is loaded onto large cargo transport ships and moved to a sea-based intermediate staging base (SISB) located at Naha, Japan. An SISB is utilized to stage cargo which will be transferred from one ocean-going transport to another. In this case, it is employed to transfer cargo from large ocean-going vessels that manage the long sea leg to smaller vessels that will shuttle the cargo to small ports. At Naha, the division is loaded onto theater support vessels (TSVs) and transported to four SPODs with reduced

capability located on the southern and eastern coasts of South Korea for rapid deployment in the theater of operations via an *intratheater sealift operation*, described by Leathrum et al. [43]. The reduced SPODs are undeveloped fishing ports or commercial seaports reduced in capability due to attack (the latter case being used in this scenario). The TSVs are a new sealift technology. These are high-speed (40 knots), low-draft (~15 feet) vessels that can utilize small ports where larger cargo vessels are not able to operate. From the reduced SPODs, the division moves by road to an inland staging area within the theater, located approximately 60 miles west of Pohang. The network configuration used for this simulation study is shown in Figure 6. The distances and transit times for the various transports operating between the terminals are shown in Figure 7. This data is largely derived from GIS data provided by SDDC-TEA. Missing data is intuitively assumed by the authors based on the available data.

The division consists of 8,750 pieces of cargo (8,686 convoy pieces and 64 helicopter pieces) partitioned into 42 units. The units are routed through one of the two terminals at Pearl Harbor, the port of Naha in Japan, and one of the reduced SPOD ports in South Korea. All helicopter cargo, irrespective of the unit they belong to, is routed through the West Loch Terminal at Pearl Harbor. The *Algol*, the *USNS Bob Hope*, and the *Antares* are used to shuttle cargo between the FISC Area Terminal and Naha, while the *Denebola*,

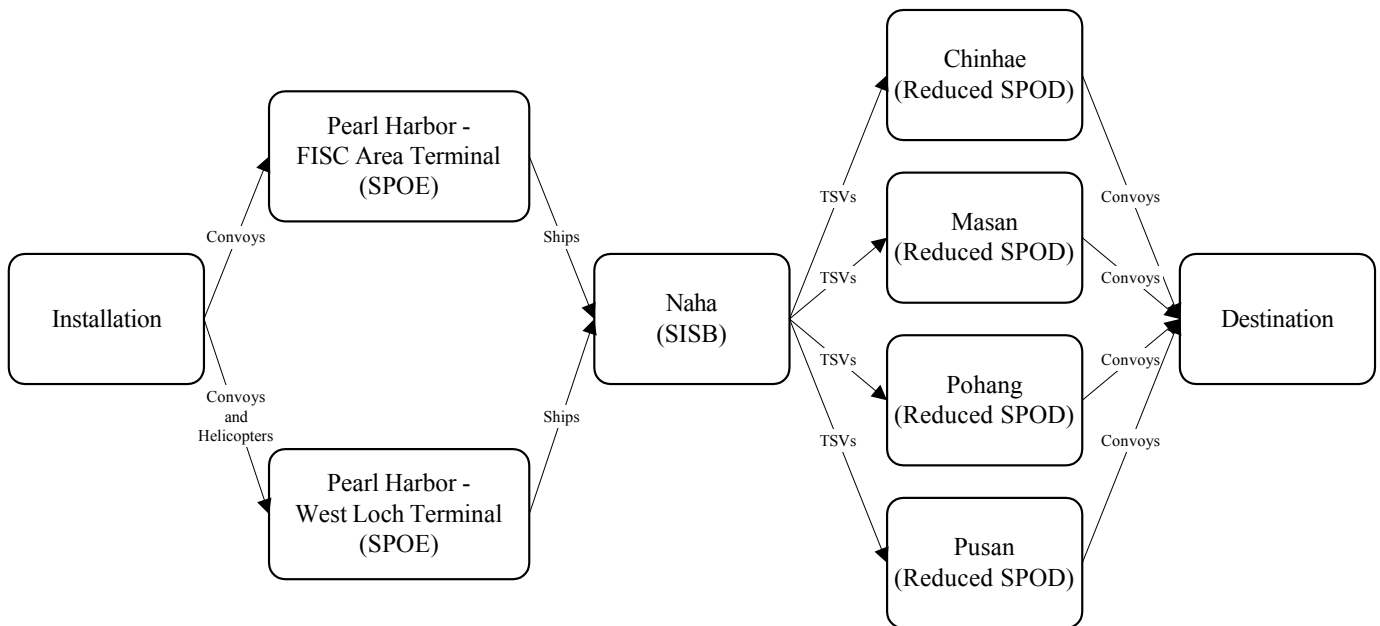


Figure 6. Scenario network configuration

| Source Terminal | Destination Terminal | Distance (miles) | Transit Time | |
|--------------------|----------------------|------------------|--------------|-------------------|
| | | | Transport | Mean Time (hours) |
| Installation | FISC Area Terminal | 40 | Convoys | 1 |
| | | | Helo pilots | 0.5 |
| | West Loch Terminal | 40 | Convoys | 1 |
| | | | Helo pilots | 0.5 |
| FISC Area Terminal | Naha | 4663 | Ships | 202 |
| West Loch Terminal | Naha | 4663 | Ships | 202 |
| Naha | Pohang | 796 | TSV | 17.29 |
| | Chinhae | 721 | TSV | 16.03 |
| | Masan | 726 | TSV | 16.85 |
| | Pusan | 717 | TSV | 15.57 |
| | Destination | 860 | Helo Pilots | 5 |
| Pohang | Destination | 62.5 | Convoys | 0.75 |
| Chinhae | Destination | 140 | Convoys | 1.68 |
| Masan | Destination | 137 | Convoys | 1.64 |
| Pusan | Destination | 133 | Convoys | 1.59 |

Figure 7. Transit times between terminals

the *Pollux*, and the *Regulus* are used to shuttle cargo between the West Loch Terminal and Naha. Twenty-two TSVs, each 430 feet in length with a draft of 15 feet, are used to transport cargo between the port of Naha and the reduced SPODs. Vehicles move from the reduced SPODs in convoys comprised of ten vehicles each.

5.2 Model Description

The model used for the scenario utilizes the following cargo terminal types: installation, seaport of embarkation (SPOE), sea-based intermediate staging base (SISB), reduced seaport of debarkation (reduced SPOD), and destination (Figure 6). Note that this is a unique scenario based on a new technology (TSVs) that has not been tested in operations as shown here.

The macro-level flow of cargo and resources in an SPOE is shown in Figure 8 as an example. The figure illustrates the flow of cargo through the terminal areas and the resources necessary at each stage to support this flow. An SPOE is usually situated in the continental United States and is the place where military equipment and troops are dispatched to be loaded onto large cargo ships to be transported close to the theater of operations. An SPOE deals with the arrival and reception of cargo at the port via rail, air, and highway transports. The cargo is staged and then

loaded aboard the appropriate ship. At an SPOE, military equipment arrives via highway transports (convoys and commercial truck assets), air (helicopters), and rail (trains). Highway transports are offloaded at a loading/unloading area, air assets at a helicopter landing area, and rail assets (trains) are received at an interchange yard and individual railcars are offloaded at a rail spur. The cargo is then staged in a staging area where it waits for an appropriate ship to arrive at the port. When a ship arrives at the port, it enters the anchorage area and requests space to berth at the port. Once a berth area and the appropriate resources (tugs and harbor pilots) are allocated, the ship moves to the berth area and, at the same time, requests the load of cargo that has been created for it. Cargo is allocated for the ship based on the destination cargo terminal through which the cargo is supposed to be moved, while maintaining military unit integrity. The allocated cargo is then sent from the staging areas to the berths to be loaded aboard the ship. The ship then leaves the port and is routed to its destination cargo terminal by the terminal interconnect interface. The macro flow within the other terminal types is internally modeled similar to that of an SPOE.

The SPOE, SISB, and reduced SPOD models are derived from the original *CPortS* model. The majority of processes are a subset of the *CPortS* SPOD model, and processes that are unique to a particular terminal

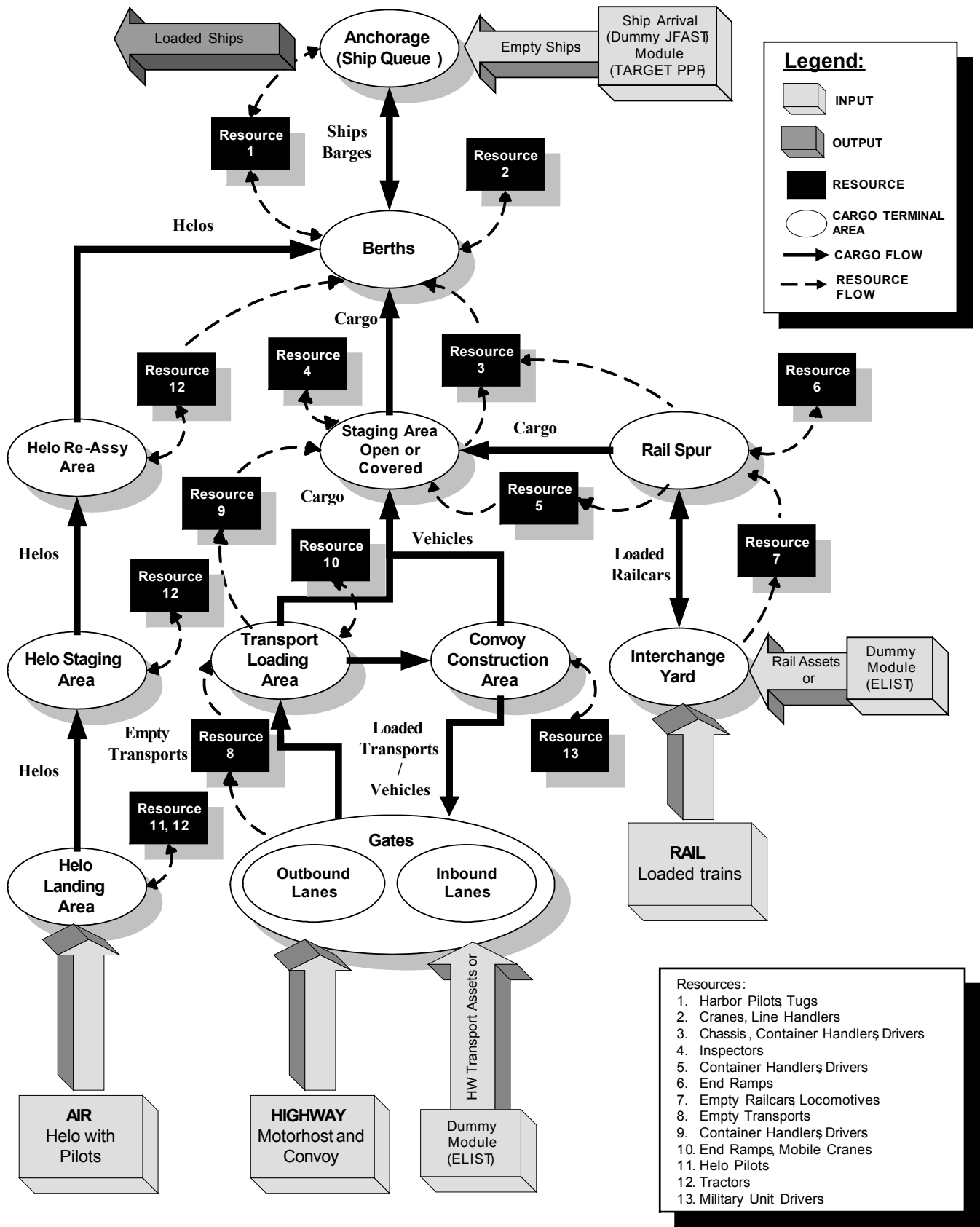


Figure 8. Macro flow for a seaport of embarkation (SPOE)

or a set of terminals are integrated into the main simulation model using the same architecture. In fact, the models are integrated into a single simulation model with a *port operational mode* attribute deciding the actual processes within the terminal as well as their order and sequence. The models also differ in the number of cargo types handled within the model for that particular mode.

The SISB, SPOD, and reduced SPOD models used in the simulation were developed with inputs from SDDC-TEA. These models are detailed and accurately represent constraints imposed by port location, geometry and size, port infrastructure, and the number and availability of various resources within the port. The processes for various operations within these models were approved by SDDC-TEA. The focus during the development of this simulation architecture was the transportation of sea-based cargo. However, the authors strongly believe that the models for air-cargo transportation can be integrated into the network using the same architecture without any major problems.

The routing and transit times both within the terminal as well as between terminals are accurate and detailed as described in the previous section. The transportation links between the various terminal models are less detailed and are modeled by a random delay of mean equal to expected ship transit time.

5.3 Scenario Analysis

The output for the simulation is reported at two levels: the network level and the individual cargo terminal level. At the network level, the flow of each piece of cargo is tracked through the entire network starting at the installation (where the cargo originates) until it reaches the final destination (where the piece of cargo leaves the simulation). The data collected at the network level includes arrival and departure times for each piece of cargo that is routed through each terminal within the network. Details about the transport that brought the piece of cargo into each terminal are also collected. An important result of this collection of data is the ability to track the shuttling of transports between cargo terminals for the duration of the simulation. These profiles help to generate a coarse-grained view of the cargo flow.

At the cargo terminal level, the flow of each piece of cargo is tracked through a particular cargo terminal, from the time it enters the terminal, through individual terminal areas, until it leaves the terminal. This is done by using the output analysis tool developed for the *CPortS* project [44]. This data feeds several graphical displays, such as cargo terminal area usage, number of pieces of cargo in a cargo terminal area at any instant of

time, etc. These profiles help to generate a fine-grained view of the cargo flow.

Figure 9a shows the terminal usage profile for every terminal in the network for the duration of the simulation. From the graph, it is clear that the mechanized division took approximately 50 days to get from its home base to the theater of operations. The clutter at the bottom of the graph from days 20 to 50 is the terminal usage profiles for the reduced SPODs. They are difficult to observe due to the fact that they handle a very low number of cargo items as compared to an SPOE or an SISB.

Figure 9b shows the activities of the ships shuttling cargo between the SPOEs and the SISB. The thick lines indicate the time that the ship was in either of the two SPOEs at Pearl Harbor for loading cargo, while the thin lines indicate the time spent by the ship at the SISB unloading cargo. A similar type of graph can be constructed for the set of TSVs shuttling cargo between the SISB and the reduced SPODs.

The results of this analysis allow military decision makers and planners to ascertain the transportation feasibility of a proposed operation. The results show the impact of the characteristics (infrastructure and resources) of a particular terminal upon the movement of cargo through the complete defense transportation network. Planners can also analyze the impact of increasing transports to shuttle cargo between different cargo terminals. The overall performance of cargo terminals and problem areas that warrant further investigation, such as bottlenecks, are readily available to decision makers.

6. Performance and Scalability

As discussed in Section 3, a single simulation run with approximately 25,000 pieces of cargo should ideally execute within 19.2 seconds to be useful in a real-time military exercise. The simulation meets this requirement executing a scenario with 26,250 pieces of cargo in 16 seconds. In addition, execution time is linear with respect to the number of pieces of cargo. The simulation is capable of handling over a million pieces of cargo, far exceeding SDDC-TEA requirements. These results allow analytical support of logistics planning for critical time scenarios, within the 96-hour requirement specified by SDDC-TEA.

The simulation operates at a high level of fidelity making acquisition of proper unclassified data to validate outside of a currently operating scenario difficult. SDDC-TEA has employed the SPOD portion of the project (*CPortS*) in support of a number of recent military operations. The simulation results closely matched what was observed during the actual operation to the satisfaction of subject-matter experts (SMEs).

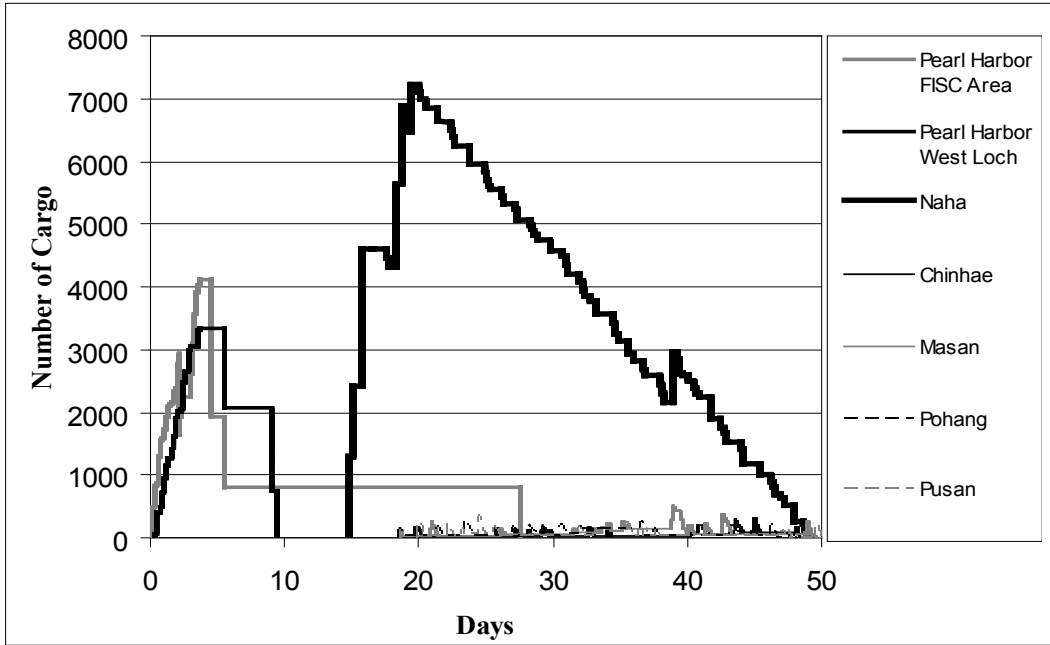


Figure 9a. Port usage profile for the network

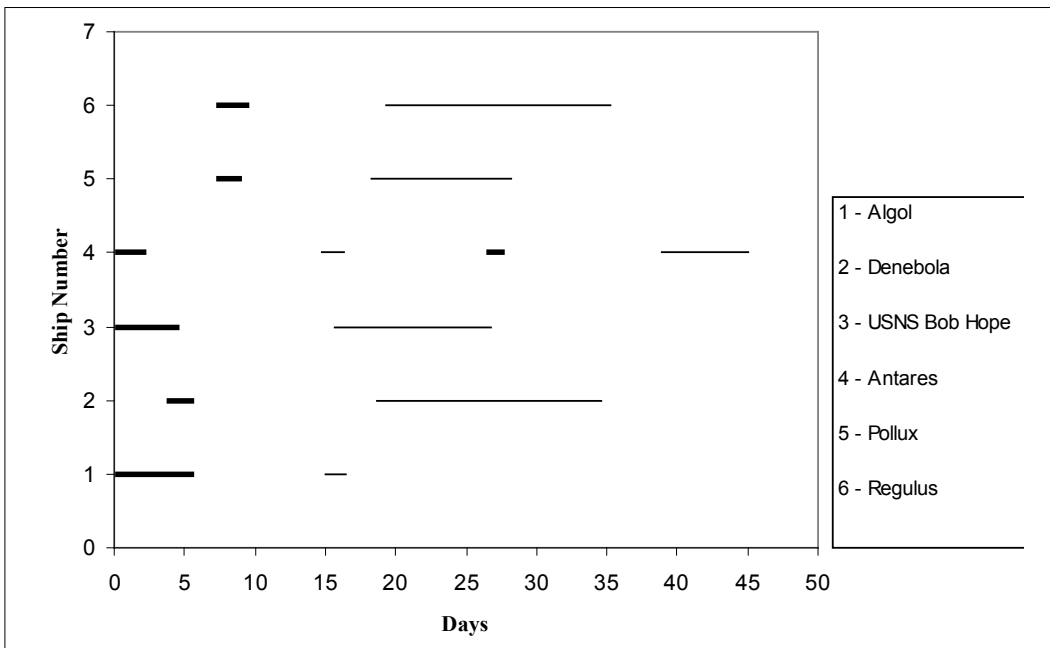


Figure 9b. Ship activity diagram between SPOEs and SISB

While not performed as a proper Turing test [45], the authors were neither in control nor had access to this activity. SMEs have also performed face validations [45] based on the results of fictitious scenarios with unclassified data and determined that the results are appropriate and reasonable. Complete results validation is pending SDDC-TEA approval, funding,

and tasking. However, SDDC-TEA has accepted the simulation and is employing it in real-world planning exercises.

7. Conclusions

A new simulation architecture supporting the simulation of end-to-end logistics for deploying military forces worldwide is presented. Various cargo terminal models are implemented using the same architecture to simulate different terminals involved in transporting a piece of cargo from an installation to the theater of operations. The example study uses three different modes of seaport operation (SPOE, SISB, and reduced SPOD), as well as installation and destination cargo terminal types. The simulation is extremely fast while maintaining a high level of detail. The simulation presents the opportunity to study the impact of cargo terminal infrastructure and resources as well as transportation resources for a network of cargo terminals.

Transportation infrastructure is not modeled, with the assumption that travel times between cargo terminals can be potentially determined through other means, such as the use of another model. For example, while developing a scenario for a particular exercise, highway infrastructure is not a parameter that can be modified; it simply needs to be analyzed. Therefore, transportation infrastructure can be analyzed once for throughput and delay information and then used as an input to the scenario analysis, where the known parameters of the infrastructure can be manipulated for an appropriate capability. This allows potential transportation infrastructures to be analyzed a priori and catalogued for use in high-fidelity simulations during times of crisis action, where the focus is on available transportation resources.

A future extension of the current architecture would be to distribute the architecture over a computer network to satisfy future performance requirements. A prototype of the approach to be utilized is described by Mazumdar et al. [46]. An intuitive method of partitioning the architecture presented in this paper is to partition the network based on cargo terminals since terminal operation areas appear to have reasonable workloads. In a standard environment like the High Level Architecture (HLA), this translates to each cargo terminal being a federate in the end-to-end federation.

The resulting architecture is scalable, and its performance exceeds specified customer requirements, resulting in improved planning time for military logistics. The ability to model different cargo terminal types and operations with the cargo terminal architecture demonstrates how the abstractions used support reuse. The nodal architecture allows the planner to focus on the resources and infrastructure that he has control over within the multimodal transfer points rather than the physical transportation infrastructure. Data from other simulations provide the

time delays resulting from the physical transportation infrastructure capacities and distances, or can be directly interfaced to the architecture. The simulation is the first within SDDC-TEA's set of models that focuses on the actual transportation resources and their interaction between the nodes in the transportation network.

8. References

- [1] Li, Z., C. M. Macal, and M. R. Nevins. "Ownership Transfer for Non-Federate Object and Time Management in Developing an HLA Compliant Logistics Model." Paper presented at the Simulation Interoperability Workshop, Logistics and Enterprise Models Forum, Spring 1998.
- [2] Hummel, J. R., and C. H. Sinex. "Development of a Comprehensive Logistics and Warfighting Simulation System." Paper presented at the Simulation Interoperability Workshop, Fall 1998.
- [3] Tustin, J. P., W. F. Ferguson, C. N. Van Groningen, C. J. Keyfauber, and E. R. Beeker. "Integrating MIDAS and ELIST into the AMP HLA Federation." Paper presented at the Simulation Interoperability Workshop, Logistics and Enterprise Models Forum, Fall 2001.
- [4] Nevins, M. R., C. M. Macal, and J. Joines. "A Discrete-Event Simulation Model for Seaport Operations." *Simulation* 70, no. 4 (April 1998): 213–223.
- [5] Nevins, M. R., C. M. Macal, R. J. Love, and M. J. Bragen. "Simulation, Animation and Visualization of Seaport Operations." *Simulation* 71, no. 2 (August 1998): 96–106.
- [6] Nevins, M. R., C. M. Macal, and J. Joines. "PORTSIM: An Object-Oriented Port Simulation." In *Proc. Summer Computer Simulation Conference (SCSC 1995)*, 160–165. 1995.
- [7] Leathrum, J. F., and T. Frith. "A Reconfigurable Object Model for Port Operations." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*, 693–698. 2000.
- [8] Leathrum, J. F. "PORTSIM: Model Visibility for Training within a Port Simulation." In *Proc. CAORF/JSACC 2000*, L1-1–L1-12. 2000.
- [9] Leathrum, J. F., R. R. Mielke, and T. Frith. "An Integrated Spreadsheet Approach to Simulation Interfaces." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*, 85–90. 2000.
- [10] Leathrum, J. F., and L. Karlberg. "Analyzing the Sensitivity of Simulation Parameters." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*, 151–156. 2000.
- [11] Burke, J. F., R. J. Love, C. M. Macal, D. L. Howard, and J. Jackson. "Modeling Force Development from Army Installations Using the Transportation System Capability (TRANSCAP) Model." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*, 85–90. 2000.
- [12] Love, R. J., J. F. Burke, C. M. Macal, D. L. Howard, and J. Jackson. "The Transportation System Capability (TRANSCAP) Model: A Mixed Language Development Approach for an Army Development Simulation." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*. np. 2000.
- [13] Hamber, B. "TLoaDS Abbreviated Systems Architecture." In *Proc. 2001 Winter Simulation Conference*, 749–757. 2001.
- [14] Braun, M. D., G. Lurie, K. Simunich, C. N. Van Groningen, H. J. Vander Zee, and M. A. Widing. "ELIST8: A Simulation System for Transportation Logistics Planning Support." In *Proc. Summer Computer Simulation Conference (SCSC 2000)*, 693–698. 2000.

- [15] Model for Intertheater Deployment by Air/Sea (MIDAS), Contact Information: OSD-PA&E (Projection Forces), Pentagon Rm. 2D272, Washington D.C., 20301-1800
- [16] DPRA, Inc. *JFAST: Joint Flow and Analysis System for Transportation*. (online at <http://www.jfast.org>)
- [17] Analysis Mobility Platform (AMP), USTRANSCOM TCJ5-SC, Scott AFB, IL.
- [18] Van Hee, K. M., and R. J. Wijbrands. "Decision Support System for Container Terminal Planning." *European Journal of Operational Research* 34, no. 3 (1988): 264-272.
- [19] Hayuth, Y., M. A. Pollatschek, and Y. Roll. "Building a Port Simulator." *Simulation* 63, no. 3 (March 1994): 179-189.
- [20] Ramani, K. V. "An Interactive Simulation Model for the Logistics Planning of Container Operations in Seaports." *Simulation* 66, no. 5 (May 1996): 291-300.
- [21] Ballis, A., and C. Abacoumkin. "A Container Terminal Simulation Model with Animation Capabilities." *Journal of Advanced Transportation* 30, no. 1 (1996): 37-57.
- [22] Kozan, E. "Increasing the Operational Efficiency of Container Terminals in Australia." *Journal of the Operational Research Society* 48, no. 2 (1997): 151-161.
- [23] Kozan, E. "Comparison of Analytical and Simulation Planning Models of Seaport Container Terminals." *Transportation Planning and Technology* 20 (1997): 235-248.
- [24] Merkurjev, Y., J. Tolujew, E. Blymel, L. Novitsky, E. Ginters, E. Viktorova, G. Merkurjeva, and J. Pronins. "A Modeling and Simulation Methodology for Managing the Riga Harbour Container Terminal." *Simulation* 71, no. 2 (August 1998): 84-95.
- [25] Gambardella, L. M., A. E. Rizzoli, and M. Zaffalon. "Simulation and Planning of an Intermodal Container Terminal." *Simulation* 71, no. 2 (August 1998): 107-116.
- [26] Bruzzone, A. G., and R. Signorile. "Simulation and Genetic Algorithms for Ship Planning and Shipyard Layout." *Simulation* 71, no. 2 (August 1998): 74-83.
- [27] Merkurjeva, G., Y. Merkurjev, and J. Tolujew. "Computer Simulation and Metamodeling of Logistics Processes at a Container Terminal." *Studies in Informatics and Control* 9, no. 1 (March 2000): 53-59.
- [28] Dzielinski, A., K. Amborski, P. Kowalczyk, J. Sukiennik, and A. Pawlowski. "Simulation of Operation of a Medium Sized Seaport. Case Study: Port of Gdansk." In *Simulation in Industry, Proc. 14th European Simulation Symposium*, 313-317. 2002.
- [29] Shabayek, A. A., and W. W. Yeung. "A Simulation Model for the Kwai Chung Container Terminals in Hong Kong." *European Journal of Operational Research* 140, no. 1 (July 2002): 1-11.
- [30] Kia, M., E. Shayan, and F. Ghotb. "Investigation of Port Capacity under a New Approach by Computer Simulation." *Computers and Industrial Engineering* 42 (2002): 533-540.
- [31] Yun, W. Y., and Y. S. Choi. "A Simulation Model for Container-Terminal Operation Analysis using an Object-Oriented Approach." *International Journal of Production Economics* 59 (1999): 221-230.
- [32] Noth, M., and A. Borning. "An Extensible, Modular Architecture for Simulating Urban Development, Transportation, and Environmental Impacts." *Computers, Environment and Urban Systems* 27, no. 2 (March 2003): 181-203.
- [33] Nielsen, O. A., and R. D. Frederiksen. "Rule-Based, Object-Oriented Modelling of Public Transport Systems." In *Proc. 2001 European Transport Conference*. np. 2001.
- [34] Lewellen, M., and K. Tumay. "Network Simulation of a Major Railroad." In *Proc. 1998 Winter Simulation Conference*, 1135-1138. 1998.
- [35] Ton, T. T., J. A. Black, and U. Vandebona. "An Object-Oriented Approach to Transport Modelling: Software Development to Calculate Road Traffic Noise." *Transportation* 25, no. 3 (August 1998): 307-328.
- [36] Chabrol, M., and D. Sarramia. "Object Oriented Methodology Based on UML for Urban Traffic System Modeling." *Lecture Notes in Computer Science* 1939 (2000): 425-439.
- [37] Trépanier, M., and R. Chapeau. "Linking Transit Operational Data to Road Network with a Transportation Object-Oriented GIS." *Journal of the Urban and Regional Information Systems Association* 13, no. 2 (2001): 23-27.
- [38] *PORTSIM Version 6.0 IOC Accreditation Criteria*. Military Surface Deployment and Distribution Command - Transportation Engineering Agency (SDDC-TEA), Newport News, VA, January 2004.
- [39] *PORTSIM SPOD Version 3.0 Test Plan*. Military Surface Deployment and Distribution Command - Transportation Engineering Agency (SDDC-TEA), Newport News, VA, December 2004.
- [40] Mazumdar, S. "Support for Statistical Analysis of the Configurable Port Simulation (CPortS)." Master's thesis, Department of Electrical and Computer Engineering, Old Dominion University, December 2002.
- [41] Mathew, R. "An Object-Oriented Architecture for Concurrent Processes in a Port Simulation." Master's thesis, Department of Electrical and Computer Engineering, Old Dominion University, December 2002.
- [42] Frank, P. "A Generic Object-Oriented Random Variate Model for Discrete Event Simulations: Separating the Data Model from the Process Model." Master's thesis, Department of Electrical and Computer Engineering, Old Dominion University, December 1999.
- [43] Leathrum, J. F., R. R. Mielke, S. Mazumdar, R. Mathew, Y. Manepalli, V. Pillai, and R. N. Malladi. "A Simulation Architecture to Support Intratheater Sealift Operations." *Mathematical and Computer Modelling* 39, no. 6-8 (May 2004). (Also to appear in the book *Defense Transportation: Algorithms, Models and Applications for the 21st Century*.)
- [44] Kiran, J. R. "Output Analysis of the Configurable Port Simulation (CPortS)." Master's thesis, Department of Electrical and Computer Engineering, Old Dominion University, May 2003.
- [45] DMSO. *Defense Modeling and Simulation Office (DMSO) VV&A Recommended Practices Guide*. (online at <http://vva.dmsomil>)
- [46] Mazumdar, S., R. Mathew, and J. F. Leathrum. "A Strategy for Distributing Simulations for Statistical Analysis." In *Proc. Summer Computer Simulation Conference (SCSC 2004)*, 151-156. 2004.

Author Biographies

Reejo Mathew is a Ph.D. candidate in Electrical and Computer Engineering at Old Dominion University. He received his M.S. degree in Computer Engineering from Old Dominion University in December 2002, and his B.E. degree in Electronics Engineering from the University of Mumbai, India, in July 2000. His research interests include distributed simulation engines, parallel and distributed simulation, and cargo logistics simulations.

Dr. James F. Leathrum, Jr. is an Associate Professor of Electrical and Computer Engineering at Old Dominion University. He earned his B.S. degree in Computer Engineering from Lehigh University in 1997 and his M.S. and Ph.D. degrees in Electrical Engineering from Duke University in 1989 and 1992, respectively. His primary research interests are in discrete-event simulation, distributed simulation, and cargo logistics simulations.

Saurav Mazumdar is a Ph.D. student in Electrical and Computer Engineering Department at Old Dominion University. His research interests include parallel and distributed simulations, 3-D visualization, virtual reality, and bioinformatics. His dissertation research will include determining how to optimally assimilate content from different virtual reality platforms.

Taylor Frith is currently the project engineer for the Military Surface Deployment and Distribution Command Transportation Engineering Agency's Port Simulation (PORTSIM) computer model and simulation program. He joined Old Dominion University's (ODU) PORTSIM research and development team in 1998. He received his B.S. degrees in Electrical Engineering Technology and Computer Science from ODU. He is currently a software engineer with MYMIC LLC, Portsmouth, Virginia.

Joseph Joines is a Transportation Systems Management Specialist in the Systems Integration Division of the Military Surface Deployment and Distribution Command, Transportation Engineering Agency. He is the Program Manager for the PORTSIM computer model. He has lent his maritime, transportation, and seaports expertise to a number of other modeling efforts. In addition to a B.S. degree in Business Management from the University of Maryland, he has taken numerous graduate-level maritime- and transportation-related courses and training. He has been a member of the Society of Naval Architects and Marine Engineers, Military Operations Research Society, and National Defense Transportation Association.