Security Inspection Protocols: Impact on Container Terminal Throughput Using Simulation

Gregory A. Harris and Bernard J. Schroer
University of Alabama in Huntsville
Huntsville, AL USA

Dietmar P.F. Moeller
University of Hamburg
Hamburg, Germany

ABSTRACT

This paper presents a simulation model to determine the impact of various container inspection protocols on the operation of a seaport container terminal. The inspection protocols are A) no inspection, B) container sampling with unloading and inspection coupled and C) inspection after unloading or decoupling inspection from unloading. Any sampling plan using Protocol B had an impact on entity throughput. Decoupling the inspection from unloading in Protocol C did not impact entity throughput. In fact, entity throughput for Protocol C was similar to no container inspection for Protocol A. Included in this paper are the development of the simulation model, the experiment to evaluate the impact of inspection protocols on port operations, an analysis of the simulation results and conclusions.

KEYWORDS

Security, container, terminal, protocols, seaport, simulation

1. INTRODUCTION

Increased security is having a significant impact on the operations of seaports resulting in longer times that ships, trains and trucks are at container terminals. Ports are wrestling with various inspection procedures and installing equipment to minimize the container inspection times.

Simulation offers an excellent approach to evaluate the impact of port activities such as container inspections on terminal operations (Harris et al. 2008). This paper presents a simulation model to determine the impact of various inspection protocols on the operation of a container terminal at a seaport. The container terminal is located at the Alabama State Docks in Mobile, Alabama.

2. RELATED WORK

An approach utilizing proposed a modeling and simulation and data fusion integration for analyzing different security aspects to improve container selection based on risk evaluations for container was proposed by Bocca, et.al. (2005).

Lewis, et.al. (2002) proposed an approach for understanding the balance between the number of containers to undergo security inspection and the cost for departure delays of outbound vessels and the port cost measured by the number of containers moved.

Honsi, et.al. (2005) focused on the training validation of an emergency response plan with traffic flow integration in the event of a disaster.

A simulation model to investigate the effects of enhanced security measures on traffic flow in and around selected port gates was developed by Chatterjee (2006). Estimates were made on length of truck queues, delays and route alternatives. Kerr (2006) developed a framework for managing freight data electronically via the Internet.

Koch (2007) created PortSim, a port security simulation and visualization tool. The tool allows a user to investigate special of parameters to determine the impact of those parameters on port operations and costs. An analysis tool for safety and security developed Berkowitz and Bragdon (2006) that depicts the air-land-seaport access and potential vulnerabilities in a virtual real-time format. This tool allows for the development of surface and underwater scenes in order to evaluate incident response training and transportation security systems.

Sandia National Laboratory (2008) developed several simulation models to assist ports in the conduct of cost/benefit tradeoffs of various security measures, such as increased inspections, or more scanners, on the movement of containers through a port.

The model described in this paper is a detailed simulation of the operation of an intermodal container center. The operations of the container center are modeled at the activity level. The various inspection protocols are then overlaid on the terminal operations. The simulation model was developed in ProcessModel, a discrete event simulation package.

3. CONTAINER TERMINAL

Figure 1 is an artists rendering of the intermodal container terminal expansion at the Alabama State Docks in Mobile, AL. Figure 2 presents a flow chart of the container ship, train and truck unloading and loading operations with no container inspection.

Figure 1. Sketch of intermodal container terminal expansion at Alabama State Docks

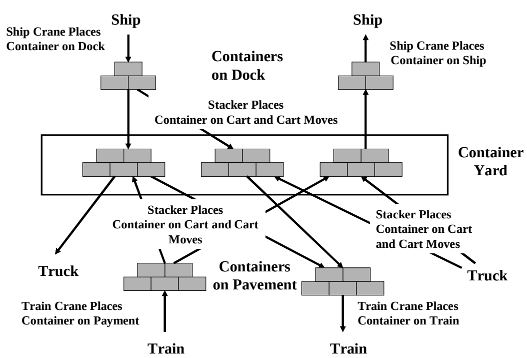

Figure 2. Overview of container traffic

4. SIMULATION MODEL

The ProcessModel (1999) in this study was initially written by Schroer et.al. (2008b) to determine the impact of various inspection sampling plans on the container throughput at the Alabama State Docks. The initial model was constructed following the conceptual framework developed by Schroer, et.al. (2008a). This conceptual framework consists of a number of submodels that run independently of one another. Each model has unique data input and entities defined by specific attributes. Data are shared between the submodels utilizing global variables. The content of the global variables can be altered within any submodel with the new values immediately shared with any other submodel. These global variables can also be used in logic statements to control the movement and routing of entities, branching logic, and updating of entity attributes.

The ProcessModel has the following submodels:

- Ship unloading and loading of containers (entity = ship)
- Train unloading and loading of containers (entity = train)
- Truck unloading and loading of containers (entity = truck, empty truck and empty truck with container)
- Movement of containers from ship dock to container yard (entity = move order1)
- Movement of containers from container yard to ship dock (entity = move order2)
- Movement of containers from train pavement to container yard (entity = move order3)
- Movement of containers from container yard to train pavement (entity = move order4)

The terminal is modeled using the following resources: ship berths, ship cranes, train slots, train cranes, truck slots, stackers, and carts. The model has thirteen entity attributes, twenty global variables, seventy activity blocks and ten entity blocks.

5. EXPERIMENTAL DESIGN

The experiment was designed to evaluate three unloading/inspecting protocols. Protocol A involves no container inspection and is used as the baseline run. The modeling logic used for ships (and similar logic is used for trains) is:

Unload
- Ship crane unloads container onto dock
- Process repeated until all containers unloaded onto dock
- After container on dock stacker loads container onto cart

Load
- Cart moves container to container year
- Stacker unloads container
- Process repeated until all containers are moved

Protocol B consists of unloading a container and then immediately inspecting the container before another container is unloaded. This protocol forbids the unloading of another container until prior container inspected. The ship logic is:

Unload
- Security inspector checks ship’s paperwork

- Ship crane unloads container onto dock
- Security inspector inspects container
- Ship crane then unloads another container onto dock (unloading is not continued until previous container has been inspected)
- Process repeated until all containers unloaded

Load
- After container inspected stacker loads container onto cart
- Cart moves container to container yard
- Stacker unloads container
- Process repeated until all containers are moved

Protocol C involves the inspection of containers independently of unloading a ship. Unloading containers continues unabated with inspections performed prior to moving the container to the container yard. The ship logic is:

Unload
- Security inspector checks ship's paperwork
- Ship crane unloads container onto dock (continue unloading even though containers not inspected)
- Process repeated until all containers unloaded

Load
- After container on dock security inspector inspects container
- Stacker places container on cart
- Cart moves container to container yard
- Stacker unloads container
- Process repeated until all containers are moved

Table 1 presents the experimental design. Protocol A is the Baseline Run with no container inspection. Three runs are using with Protocol B. An inspection rate of 100% is used in Run2, 80% in Run3 and 60% in Run4. In Protocol C the inspection is decoupled from container unloading and all containers are inspected independently of unloading from ship.

Table 1. Experimental design

Run	Description
Run1	Protocol A - no container inspection (Baseline Run)
Run2	Protocol B - 100% inspection of incoming containers
Run3	Protocol B - 80% inspection of incoming containers
Run4	Protocol B - 60% inspection of incoming containers
Run5	Protocol C – Container inspection independent of unloading

6. PROTOCOL A – NO CONTAINER INSPECTION

The input data for Run1 are shown in Table 2. In addition, the input data consisted of:

- Two ship berths for unloading and loading containers
- Two train terminals for unloading and loading containers
- Twenty truck slots (Maximum number of trucks in terminal at one time)
- Two ship cranes for unloading and loading containers from planes
- Two train cranes for unloading and loading containers from trains
- Twelve stackers for unloading and loading containers from trucks and onto and off carts
- Twenty carts for moving containers throughout the terminal
- 2 minutes to unload or load a container from plane, train, or truck
- T(15,20,25) minutes to position a ship at a terminal (T = Triangular distribution)
- T(15,20,25) minutes to position a train at a terminal
- T(4,5,6) minutes to position a truck for unloading or loading
- 2 minutes to process paperwork to load a plane, train or truck
- T(4,5,6) minutes for plane, train, or truck to exit terminal
- 2 minutes to unload and load a cart
- T(4,5,6) minutes to move a cart between a plane, train or truck and the container yard

The simulation model ran 1,440 hours, or 180 eight-hour days, which is six months. The results of the simulation for Run1 are presented in Table 3. Several interesting observations can be made with reference to Run1:

- The utilization rate is relatively high for ship berths and cranes at 68% and 67% respectively.
- The utilization rate is relatively high for train slots and train cranes at 71% and 72% respectively.
- The utilization rate is very low for tugs at 1% indicating one less tug may be possible.
- The utilization rate is very low for truck slots (the maximum number of allowed trucks in the terminal at one time is 20). It may be possible to reduce this resource freeing up space for other terminal operations.

- There are currently more stackers than required since average utilization is 34%.

Table 2. Entity parameters

Entity	Time between Arrivals (min)	
Ship	T(1320,1440,1560)	
Train	T(420,480,540)	
Empty Train	T(2080,2320,2560)	
Truck with Full Container	T(54,60,66)	
Empty Truck	T(90,120,150)	
Truck with Empty Container	T(180,240,300)	
Entity	**Containers In**	**Containers Out**
Ship	T(400,450,500)	T(200,250,300)
Train	T(90,100,110)	T(90,100,150)
Empty Train	0	T(90,100,150)
Truck with Full Container	1	81% leave with container 10% leave with no container 9% leave with empty container
Empty Truck	0	1
Truck with Empty Container	1	81% leave with container 10% leave with no container 9% leave with empty container

7. PROTOCOL B – CONTAINER SAMPLING RUNS2-4

In simulation Runs 2, 3 and 4, inspection times were modified as shown below:

- T(10,15,20) minutes for inspector to check paperwork before unloading of containers from ship or train
- T(2,3,4) minutes for inspector to check paperwork and container from truck
- 3 minutes for inspector to inspect a container from a ship, train or truck
- Five inspectors available to inspect containers

Table 3. Protocol A results

Entities through Terminal	Qty.	Time (min)	Value Added Time (min)
Ships	59	2,013	1,349
Trains	180	684	441
Empty Trains	38	424	251
Trucks	1,440	26	14
Empty Trucks	725	21	11
Truck with Empty Container	358	27	14
Resource	**Qty.**	**Utilization**	
Ship Berths	2	68%	
Ship Cranes	2	67%	
Tugs	2	1%	
Train Slots	2	71%	
Train Cranes	2	72%	
Truck Slots	20	3%	
Stackers	12	34%	
Carts	20	52%	
Containers	**Unloaded**	**Loaded**	**At Terminal**
Ships	23,746	15,309	224
Trains	18,185	23,407	564
Trucks	1,440	2,163	1,696
Empty Containers	358	156	202
Total	43,729	41,035	2,686

Tables 4-6 present the results for Runs 2, 3 and 4 with the associated varying percentages of container inspections. The simulation models ran for 1,440 hours, or 180 eight-hour days.

Ships processed through the terminal dropped from 59 with no container inspection in Run1 to 53 with 100% inspection in Run2, a 10% reduction in throughput. Likewise, the time a ship stayed in the terminal showed significant an increase from 2,013 minutes with no container inspection to 7,258 minutes with 100% inspection, an increase of 260%. Trains processed through the terminal dropped from 180 with no container inspection to 166 with 100% inspection, a 7% reduction. The time a train spent at the terminal increased from 684 minutes with no container inspection to 3,952 with 100% inspection, an increase of 477%. The truck throughput remained relative constant (1,440 to 1,441); however, the time a truck was in the terminal increased 96% from 26 minutes with no container inspection to 51 minutes with 100% inspection. The containers processed through the terminal reduced from 43,729 with no container inspection in Run1 to 39,866 with 100% inspection, an 8% decrease in throughput.

Table 4. Protocol B results: entity time at terminal

Entity	Run2		Run3	
Sampling	100%		80%	
	Qty	Time (min)	Qty	Time (min)
Ships	53	7,258	57	3,677
Trains	166	3,952	176	1,966
Empty Trains	37	593	37	532
Trucks	1,441	51	1,439	34
Empty Trucks	724	42	720	23
Trucks With Empty Containers	358	45	361	28
	Run4			
Sampling	60%			
	Qty	Time (min)		
Ships	59	2,778		
Trains	179	987		
Empty Trains	38	502		
Trucks	1,439	32		
Empty Trucks	717	21		
Trucks with Empty Containers	360	27		

Ships processed through the terminal increased 11% from 53 with 100% inspection in Run2 to 59 with 60% inspection in Run4. Note that the ships processed through the terminal in the baseline run were 59 with no inspection. The time for a ship at the terminal decreased 61% from 7,258 minutes for 100% inspection to 2,778 minutes with 60% inspection. It is interesting to note that the time a ship was in the terminal with no inspection was 2,013 minutes.

The results for trains were similar to that for ships. Trains through the terminal increased 7% from 166 with 100% inspection in Run2 to 179 with 60% inspection in Run4. The trains processed through the terminal were 180 with no inspection in Run1. The time a train spent at the terminal decreased from 3,952 minutes with 100% inspection to 987 minutes with 60% inspection, a 75% reduction. With no inspection in the baseline run the time a train spent in the terminal was 684 minutes.

The utilization of resources remained fairly constant during the simulation runs with varied sampling rates. The total quantity of containers unloaded increased from 39,866 in Run2 with 100% inspection to 43,843 in Run4 with 60% inspection, a 9% increase.

Table 5. Protocol B results: resource utilizations

Resources	Qty	Run2	Run3	Ruu4
Sampling		100%	80%	60%
		Util	Util	Util
Ship berths	2	99%	99%	90%
Ship Cranes	2	98%	98%	93%
Tugs	2	1%	1%	1%
Train Slots	2	99%	99%	94%
Train Cranes	2	96%	95%	91%
Truck Slots	2	6%	4%	3%
Stackers	12	32%	34%	35%
Carts	20	48%	51%	53%
Inspectors	5	56%	54%	49%

Table 6. Protocol B results: full container throughput

Entity	Run2	Run3	Run4
Sampling	100%	80%	60%
Ships			
In	21,707	23,528	24,309
Out	13,454	14,192	14,631
Yard	924	886	852
Trains			
In	16,718	17,710	18,094
Out	21,842	23,011	23,511
Yard	1,184	1,879	1,956
Trucks			
In	1,441	1,440	1,440
Out	2,176	2,169	2,147
Yard	279	526	729
Total			
In	39,866	42,678	43,843
Out	37,472	39,372	40,289
Yard	2,384	3,291	3,537

8. PROTOCOL C – CONTAINER INSPECTION AFTER UNLOADING RUN5

Table 7 shows the results for Run5 with Protocol C. The simulation models ran for 1,440 hours, or 180 eight-hour days. Surprisingly the results were identical to the baseline run with no container inspection.

Also surprising was that the security inspection did not delay the loading of containers onto ships, trains and trucks. The ProcessModel does not have the necessary detail logic to uniquely identify a container in the terminal and to assign the container for loading on a specific entity. As a result, as long as containers are in the container yard the loading continues.

Table 7. Protocol C results

Entities through Terminal	Qty.	Time (min)	Value Added Time (min)
Ships	59	2,007	1,352
Trains	180	695	450
Empty Trains	38	430	255
Trucks	1,442	33	20
Empty Trucks	719	21	12
Truck with Empty Container	360	27	14
Resource	**Qty.**	**Utilization**	
Ship Berths	2	67%	
Ship Cranes	2	67%	
Tugs	2	1%	
Train Slots	2	72%	
Train Cranes	2	72%	
Truck Slots	20	3%	
Stackers	12	34%	
Carts	20	52%	
Inspectors	5		
Containers	**Unloaded**	**Loaded**	**At Terminal**
Ships	23,669	14,786	674

9. ANALYSIS

Figures 3 through 5 presents bar chart graphs of the time ships, trains and trucks respectively spent in the terminal. The total time entities spent in the terminal for Protocol B were considerably greater than for Protocol A Run1. However, with Protocol C, the time in the terminal and the quantity of containers unloaded were almost identical to the baseline run conditions.

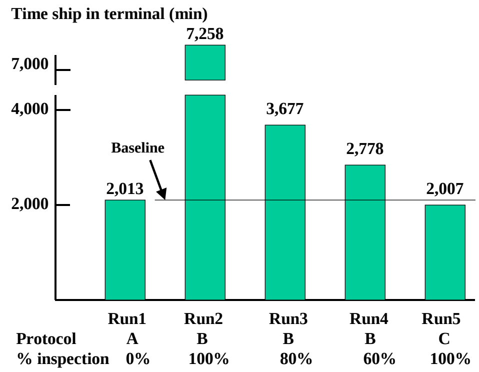

Figure 3. Time ship was in terminal

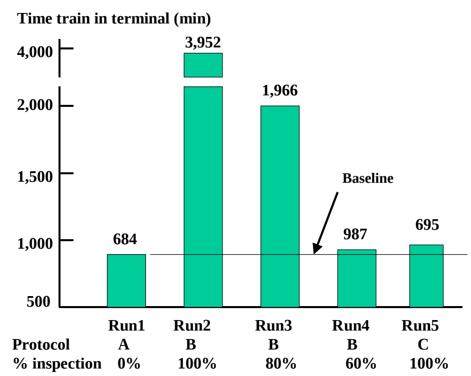

Figure 4. Time train was in terminal

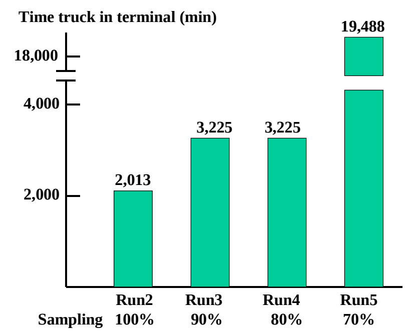

Figure 5. Time truck in terminal

Figure 6 presents the quantity of containers unloaded at the terminal for Runs1-5.

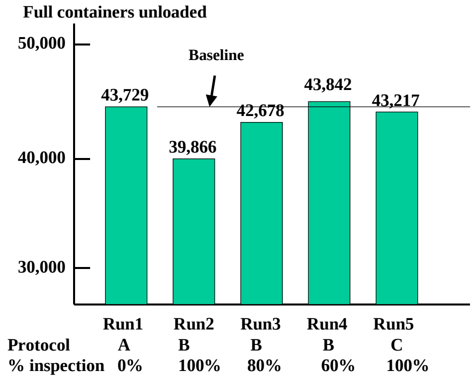

Figure 6. Containers unloaded

10. CONCLUSIONS

In summary the following conclusions are made:

- Any inspection plan for containers that includes inspection as a part of the unloading operation, such as that described in ProtocolB, increased the times for entities at the terminal. For example, 100% inspection of all incoming containers increased the time a ship was at the terminal by 260%, a train by 477% and a truck by 96% (Run2 versus no inspection for Run1). A 60% sampling plan of incoming containers increased the time a ship was at the terminal by 38%, a train by 44% and a truck by 20% (Run4 versus no inspection for Run1).

- Decoupling the container inspection from the unloading of the container minimized the impact of the inspection. The inspection protocol C for Run5 resulted in entity times identical to the Baseline Run1 with no inspection. The time a ship was at the terminal was 2,007 minutes for Run5 as compared to 2,013 minutes for Run1. The time a train was at the terminal was 695 minutes as compared to 684 for Run1. The time a truck was at the terminal was 33 minutes as compared to 26 minutes for Run1. It can be assumed that the decoupled inspection process might require resources similar to the in-process inspections described in Protocol B.

- The ProcessModel that was previously developed to simulate a container intermodal center was easily and rapidly modified to include the container inspection logic.

In conclusion container inspection protocols are critical in minimizing delays at the container terminal. It is obvious that any sampling protocol must be decoupled as much as possible from the actual unloading of containers.

Using simulation, it is a rather simple to evaluate the impact of various sampling protocols on the overall terminal operations. New inspection equipment is constantly being introduced that improve the inspection process and at the same time reduces inspection times. Again simulation can be readily applied to evaluate this new equipment and times.

Additional research should be undertaken where specific containers can be tracked for measuring the velocity of freight through the terminal and the resources required under various inspection protocols.

11. ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Department of Transportation, Federal Transit Administration, Project No. AL-26-7262-00.

12. REFERENCES

Berkowitz, C. and C. Bragdon, 2006: "Advanced Simulation Technology Applied to Port Safety and Security," *Proceedings 9th International Conference on Applications of Advanced Technology in Transportation*, Chicago, IL.

Bocca, E., S. Viazzo, F. Longo and G. Mirabelli, 2005: "Developing Data Fusion Systems Devoted to Security Control in Port Facilities," *Proceedings of 2005 Winter Simulation Conference.*

Chatterjee, A. 2006: *Assessing the Impact of Port Security Measures on Traffic Operations,* Final Report, University of Tennessee, Knoxville, TN.

Honsi, Y., S. Ramasamy, J. Selter, and P. Anderson, 2005: *Vulnerability Assessment and Emergency Evacuation Plan Simulation Training and Validation of the Port of Jacksonville,* Final Report, Center for Advanced Transportation Systems Simulation, University of Central Florida, Orlando, FL.

Kerr, P. 2006: *Development of an Interactive Freight Mobility and Security Database Structure for Research and Freight Modeling Applications*, Final Report, Center for Advanced

Transportation Systems Simulation, University of Central Florida, Orlando, FL.

Koch, D. 2007: "PortSim-A Port Security Simulation and Visualization Tool," *Proceedings 41st Annual IEEE International Cranahan Conference on Security Technology*, pages 109-116.

Lewis, B., A. Erera and C. White, 2002: *Optimization Approaches for Efficient Container Security Operations at Transshipment Seaports*, Research Report, Georgia Institute of Technology.

ProcessModel, 1999: *Users Manual*, ProcessModel Corp., Provo, UT.

Sandia National Laboratory, 2008: *Port Simulation Model*, Description on the Sandia web site www.sandia.gov, Albuquerque, NM.

Schroer, B., G. Harris, and D. Moeller, 2008a: *Conceptual Framework for Simulation Seaport Terminals*, UAH Research Report, University of Alabama in Huntsville, Huntsville, AL.

Schroer, B. G. Harris, W. Killingsworth and D. Moeller, 2008b: *Simulation of an Intermodal Center Served by Air, Train and Truck*, UAH Research Report, University of Alabama in Huntsville, Huntsville, AL.

13. BIOS

Gregory Harris is Director of the Alabama Technology Network at UAH. He is a certified NIST lean manufacturing trainer. Gregory holds faculty positions in the College of Engineering and the College of Business at UAH and teaches Production & Inventory Control and Supply Chain Management. Harris has a Ph.D. in Industrial and Systems Engineering from UAH and is a registered Professional Engineer.

Bernard Schroer is Principal Research Engineer at the University of Alabama in Huntsville (UAH). He is a Fellow of IIE, a Fellow of the SME and a member of SCS. He has a Ph.D. in Industrial Engineering from Oklahoma State University and is a registered Professional Engineer.

Dietmar Moeller is Professor of Computer Science and Computer Engineering at the Mathematics, Computer Science and Science Faculty of the University of Hamburg, Germany. He also serves as Chair of Computer Engineering. Moeller has a Dr.-Ing. (Ph.D.) in electrical engineering and control theory from the University of Bremen.