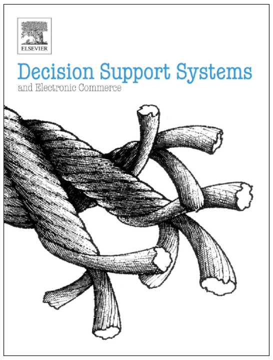


A decision support system for integrated container handling in a transshipment hub

Pasquale Legato, Rina Mary Mazza

PII: S0167-9236(18)30033-2
DOI: https://doi.org/10.1016/j.dss.2018.02.004
Reference: DECSUP 12930

To appear in: *Decision Support Systems*

Received date: 22 May 2017
Revised date: 24 November 2017
Accepted date: 9 February 2018

Please cite this article as: Pasquale Legato, Rina Mary Mazza , A decision support system for integrated container handling in a transshipment hub. The address for the corresponding author was captured as affiliation for all authors. Please check if appropriate. Decsup(2017), https://doi.org/10.1016/j.dss.2018.02.004

A decision support system for integrated container handling in a transshipment hub

Pasquale Legato, Rina Mary Mazza

Department of Informatics, Modeling, Electronics and System Engineering (DIMES)
University of Calabria
Via P. Bucci 42C
Rende (CS), 87036, ITALY
legato@dimes.unical.it, rmazza@dimes.unical.it

Abstract

The productivity of a maritime container terminal can be improved through a model-driven decision support system (DSS) focused on a better integration among container handling operations occurring across the quay, transfer and yard areas. Integration is pursued to minimize the blocking, locking and other queuing phenomena which are unavoidable, especially when human-operated equipment is shared in a real environment subjected to several random events and activities. An integrated queuing network is proposed in this paper as the natural modeling paradigm for a DSS aimed to highlight and quantify the blocking, locking and other queuing phenomena experienced in real practice. After an in-depth discussion of the limitations of solving the queuing network model analytically, discrete-event simulation is adopted as solution method. Numerical examples referred to a case study for a real transshipment hub return reliable estimates for the above queuing phenomena. They illustrate how the queuing-based DSS may effectively support the operations manager in determining the proper operational policies and equipment management with respect to a proficient integration of container handling operations.

Keywords: Model-driven decision support systems, port operations, integration, blocking and locking, queuing model, simulation

1. Introduction

In 2013 the 10 largest container ports in the world accounted for 39.7% (i.e. 204m TEUs) of the total volumes handled by the world’s 100 leading container ports [10]. These mega-ports are responding to the ongoing vessel upsizing pursued by the container industry which is also embracing new alliance strategies in the attempt to return to sustainable profits by cutting down on operating costs. As a result, the container terminals working within these mega-ports are expected to comply with the corresponding need of productivity surge by adopting more flexible and effective infrastructures, equipment, policies and practices. Supposing that any further investment worth billions of dollars in new technology is currently to be ruled out, the most probable options for productivity raise in container terminals will likely rely on the integrated management of the internal logistic processes. Hence, the central role played by a proper decision support system (DSS) for operations integration and competitiveness becomes evident.

In this paper we propose a model-driven DSS [30] for a real maritime container terminal devoted to pure transshipment. The terminal features human-operated equipment: rail mounted gantry cranes on the quay and a fleet of self-lifting shuttle vehicles called straddle carriers (SCs), where the latter provide for both container handling and transfer. The DSS is based on a specialized queuing-based modeling paradigm and it accounts for system infrastructure and human behavior under various sources of randomness. In particular, with respect to resource assignment and operational policies, the DSS is used to support decisions concerning *i*) the number of SCs to be assigned to each quay crane (QC) involved in container discharge/loading (D/L) operations, *ii*) the container deployment policy to be followed when stacking containers in the storage blocks on the yard and *iii*) the “best” path to be taken by the SC drivers according to the vehicle traffic they encounter along horizontal and vertical corridors and intersection points of the yard. Observe that after rubber-tired gantry cranes, SCs are the second most used material handling equipment (20.2%) in the storage yard [38]. So the core system of the DSS can be generalized and used in other container terminals as well.

For a preliminary understanding of how a DSS can aid operations integration, let us introduce the blocking, starvation and locking problems arising across the quay and yard areas. Under limited container storage capacity at the bottom of each QC, crane (service) blocking occurs during container discharge whenever the storage capacity is saturated due to an inadequate rate of container retrieval by the dedicated SCs. Vice versa, crane starvation occurs during container loading whenever a QC finds no containers in its buffer area due to an inadequate arrival rate of SCs with containers. In the yard area, row locking is necessary because of the limited space between adjacent yard rows: SCs cannot simultaneously operate in adjacent yard rows. This stated, on the quay, operations integration minimizes the blocking and starvation phenomena suffered by QCs. In the transfer area, integration is meant to minimize vehicle interaction due to congestion along the transfer paths and at road intersections. To finish, operations integration on the yard allows to minimize waiting and row locking phenomena affecting container handling operations performed by the SCs.

The rest of the paper is organized as follows. In Section 2 the literature on the use of DSSs in container terminals is reviewed. In Section 3 the core model of the DSS is proposed as a queuing network model which discloses the blocking, locking and congestion phenomena. Discrete-event simulation is adopted to overcome the limitations underlying the analytical solution. A description of the DSS interface follows. A case study is presented in Section 4. Conclusions are drawn in Section 5.

2. Literature Review

Decision making in container terminals has received important support from the scientific community over the last years. In particular, considerable research has been devoted to supporting the allocation and scheduling of container transfer and handling equipment as separate decision problems [6, 7]. With respect to this field, the development of DSSs can be grouped into four different areas.

Containers are the main focus of the first area. In [35] Shen and Khoong propose a DSS for the distribution of empty containers to the ports that need them. The DSS is based on large-scale network optimization models for optimizing the flow of empty containers over a multi-period planning horizon and it allows to account for changes in the supply and demand of empty containers. Bandeira et al. [1] propose a DSS for the integrated distribution and allocation of empty and full containers along the supply chain with the aim of minimizing the cost (or time) in global transportation. The core of the system is given by a network model which is solved according to two stages: a static stage for container allocation and movement, a dynamic stage for supply/demand update in time. Dereli and Daş [13] design a DSS for the container loading problem whose aim is to support packing items into a container without overlapping, while maximizing the utilization rate of the container. A swarm intelligence algorithm is the main component of the system which is also provided with a graphical interface and a simulation module for 3D visualization. In [17], Fazi et al. use a DSS to allocate import containers to a heterogeneous fleet composed by barges and trucks. The problem is modeled as a classic heterogeneous fleet vehicle routing problem with the objective of minimizing the transportation costs of containers from the seaside to the hinterland.

Stowage planning is the subject of the second area. The system focused by Liu et al. [23] supports decision-making with respect to demand forecasting, container stowage planning and shipping line optimization. These three modules are based on the use of some classical approaches (i.e. exponential smoothing, regression analysis and neural networks), an LP formulation and a genetic algorithm integrated with sequence alignment, respectively. In [12] Delgado et al. propose a DSS to generate a container vessel stowage plan from a master plan with respect to container loading to be carried out in a single port. An LP model and a greedy heuristic are combined in a 2-phase placement heuristic for stowage plans generation.

Decision problems pertaining to equipment are in the third area. Moghadam et al. [25] propose a DSS for selecting the yard equipment among straddle carriers, rubber-tyred gantry cranes and rail-mounted gantry cranes with the aim of optimizing truck turnaround times. This is obtained by combining a multiple attribute decision making method together a fuzzy analytical hierarchy process. Ngai et al. [28] design and develop an intelligent context-aware DSS to support operational decision-making in a container terminal. Their system employs ubiquitous computing technology to track the geographical locations and status of trucks, QCs and rubber-tyred gantry cranes for performing accurate real-time monitoring. Ding et al. [14] present a DSS that is designed to improve QC operations by combining tandem lifting and dual cycling. The system embeds a two-stage mathematical model and a two-level heuristic for container sequencing on a single vessel bay with the objective of minimizing the makespan.

Coordinating and analyzing the performance of container transfer and handling equipment in a, more or less, integrated manner is the object of the last area. In [26] Murty et al. discuss the mathematical models and algorithms used in designing a DSS to support the daily operations in a container terminal. The discussion covers storage space assignment, dispatching policy at the terminal gate and the berth, the routing of trucks in the storage yard, optimal deployment of yard cranes among the blocks, and the optimal allocation of internal trucks to QCs, and hiring internal trucks over the day. Salido et al. [32] develop a DSS with the aim of minimizing both the waiting times of vessels and the amount of relocations of containers on the storage yard. The system integrates a heuristically-guided planner for generating rehandling-free intra-block remarshalling plan for container yards and a meta-heuristic approach for solving the berth assignment and quay crane assignment problems. In [37] Ursavas provides a DSS that simultaneously solves the berth allocation, QC allocation and QC scheduling problems. It is built around a bi-objective optimization model that minimizes the labor costs and the service time of a vessel, while bearing the flexibility of adjusting the balance within these conflicting objectives.

Although contributions by [26, 32, 37] refer to operations integration across different terminal areas, none of them focus on the specific detection and evaluation of i) blocking and starvation phenomena arising at the bottom of the QCs, ii) vehicle interaction along the transfer paths and waiting at road intersections and iii) waiting and locking phenomena triggered by shuttle vehicles that share, in mutual exclusion, stacking/retrieval locations within the yard area. Our model-driven DSS aims to contribute to this gap within the literature, since the occurrence of any of the above events affects the vessel turnaround time. In particular, our queuing-based DSS allows to capture the realistic propagation of delays within the overall container discharge/loading process. Moreover, the integrated handling and transfer of every individual container is explicitly modeled, while accounting for system infrastructure, traffic, security issues and human behavior of the man-operated SCs on a Manhattan like yard storage area.

3. The Model-driven DSS

The objective of the model-driven DSS proposed in the following is to effectively support the operations manager in determining the proper operational policies and equipment management with respect to a proficient integration of container handling operations. The DSS has been designed so even non-technical users can manipulate model parameters to examine the sensitivity of outputs or conduct a more ad hoc "what-if" analysis. The aim of the DSS is to detect and quantify blocking, locking and other queuing phenomena in an SC-based container terminal. The system runs under Microsoft Windows 8 on a pc equipped with an Intel core I7 3.5GHz quad-core processor and 16G of RAM.

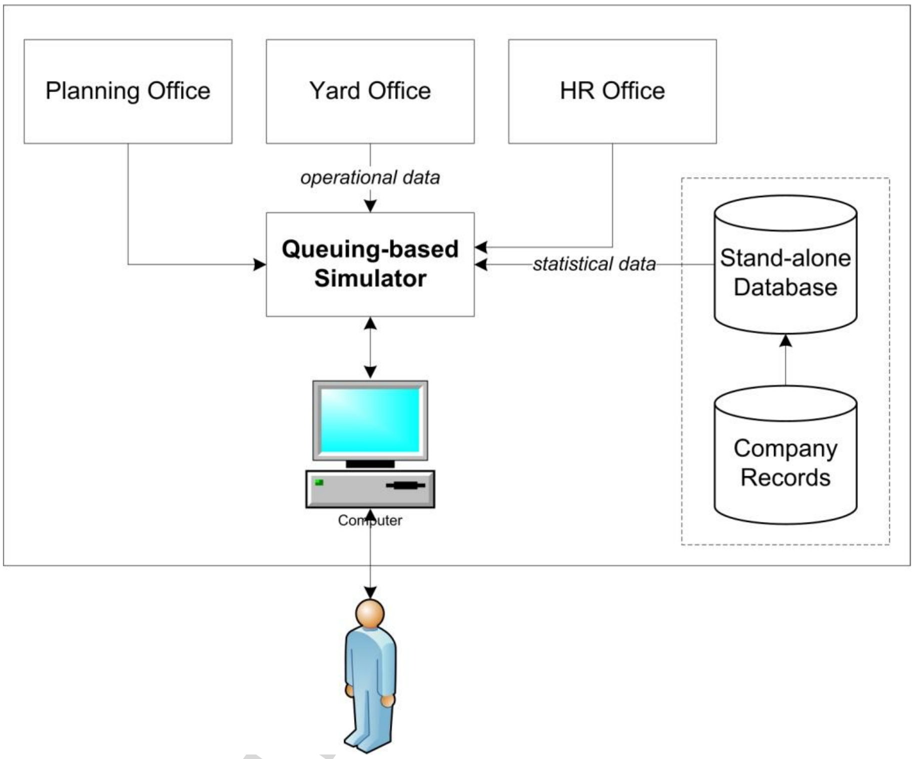


Figure 1 – Architecture of the model-driven DSS

Inspired by the classical design proposed by Sprague [36], the DSS consists in three major components: the model system, the data system and the user interface (see Figure 1). The model system is the dominant component in the architecture. Given the complexity of SC-based container terminals, an integrated model that can handle a realistic set of interrelated decisions is used to provide the functionality for the DSS. The data system manages the information flow from both company departments (i.e. planning office, yard office and HR office) and company databases. Most successful DSSs have found it necessary to create a DSS database which is logically separate from other operational databases [36]. This option is also portrayed in Figure 1 where a stand-alone database is required when, for instance, the main information system of a container terminal is designed around two separate databases that cannot be synchronized. Finally, the user interface is the component from which much of the power, flexibility and usability characteristics of the DSS are derived. A “good” interface holds a fair trade-off in terms of action language, presentation language and knowledge base.

More detailed explanations including how the conceptual model of the DSS was conceived, solved and presented for simple interaction with the user.

3.1. The Queuing Network Model

The major logistic operations in a container terminal devoted to pure transshipment are carried out through the interaction of the quay, transfer and yard areas. In an SC-based transshipment terminal, QCs are assigned, deployed and scheduled along the quay in order for container discharge/loading operations to take place. During these operations, SCs provide for both container transfer from the quay to the yard and vice versa, along with container stacking and retrieval on the terminal yard. So the quay, transfer and yard areas are logically connected to form an integrated system that can be modeled by the closed-type queuing network in Figure 2.

In this model, the fixed-size fleet of SCs represents the fixed population of customers circulating within the model which can easily be seen as formed by three subsystems: quay, transfer and yard. Containers, in turn, are passive resources that need to be seized/released by SCs from/to vessels berthed along the quay or from/to storage rows within the yard. Arriving and departing vessels determine the volume of containers to be handled by the fleet of SC-customers whose optimal size could be determined by a “what-if” analysis supported by the queuing network model at hand.

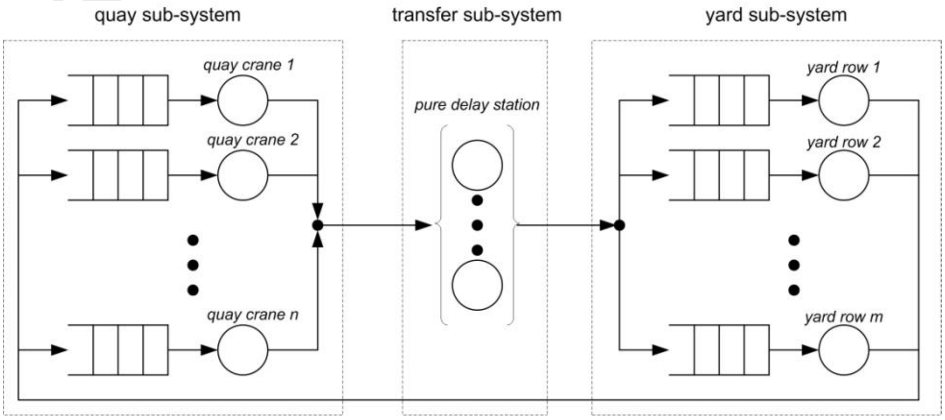


Figure 2 - The queuing network model of the integrated quay-transfer-yard system

Some non-standard features in each of the terminal sub-systems prevent us from obtaining an analytical solution (to be used within the DSS) for the model in Figure 2 by resorting to the celebrated mean value analysis algorithm [31]. The major of the non-standard features are now described, under the assumption that the time horizon of the model ranges within a matter of days.

To begin with, during container discharge/loading operations in the quay sub-system, gantry cranes act like on-off sources according to their busy-idle periods. Crane busy-idle time periods are triggered by the arrival/departure of different vessels at a D/L point where container discharge/loading operations are performed. The duration of these discharge/loading operations (i.e. the time it takes a quay crane to move a container from on board the vessel to the quay ground or vice versa) is expected to be rather deterministic, but it actually varies due to service interruptions during normal operations. The number and time durations of these interruptions can prevent using an exponential distribution to model the crane service time and, thus, prevent solving the queuing network by applying analytical methods.

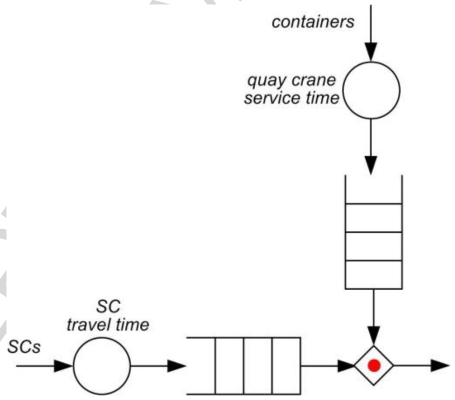


Figure 3 - SC and quay crane synchronization in container discharge

An additional problem to applying this kind of solution method is featured by the SC-container synchronization phenomenon occurring at the bottom of the crane. As illustrated in Figure 3 for the discharge option, an SC first joins a container unloaded by a QC in the synchronization point marked with a dot and, after the pick up operation, the SC leaves for the yard for container stacking. The container set down operation occurring at the end of the SC travel, in a target stacking position within a specific yard row, corresponds to the fork operation. Fork and join mechanisms are always present in container loading and discharge operations, respectively, and, in either case, they inhibit the use of exact solution methods of the queuing network model.

As for the transfer sub-system, in our queuing network the travel service is modeled as a pure delay station with a finite number of parallel paired servers. These paired servers, one per SC involved in transfer operations, are used to map the vehicle's different status during the two legs of its round trip and each is characterized by a different service time (unloaded vehicles are usually faster). Vehicles can be considered independent at low traffic levels; otherwise, as the number of vehicles increases, congestion-based dependency among different pairs of servers arises and should be properly captured by an increasing service transfer time. So, both the number of vehicles simultaneously circulating in the transfer area and the status of the vehicle determine the individual service time of the corresponding pair of servers in Figure 4. An alternative approach to modeling the transfer sub-system could be based on the idea of using the so-called flow-equivalent server station [8] to represent the entire transfer sub-system and evaluate its throughput. Specifically, vehicle interaction could be captured through the state-dependent flow-equivalent server rate and the number of vehicles simultaneously circulating in the transfer area could be captured by the number of customers in the flow-equivalent station.

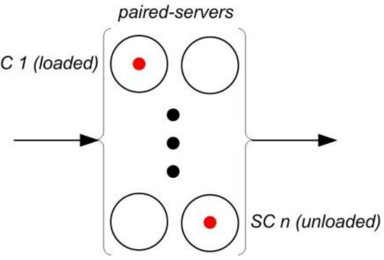


Figure 4 - The transfer subsystem modeled by parallel paired servers

Unfortunately, the analytical evaluation of the flow-equivalent station requires the exponential assumption on the "state-dependent service rate". Additionally, it is worth remarking that using the first-moment of the real distribution does not return satisfactory results when evaluating performance metrics such as waiting times and probabilities.

Finally, in the yard sub-system, handling and transfer operations from/to a given yard row require accessing that row in compliance with the security measures that depend on the yard organization and technology. An SC is granted access to a row if no other SC is already performing handling operations in that row or in the adjacent rows. As shown by the simple example portrayed in Figure 5, service initiation in the central row (i.e. SC entering the central row) issues a busy condition for that row, but also a so-called locking condition on the two adjacent rows. This yields dependency relations among network stations, due to the phenomenon of service prevention at a given station when service starts at the adjacent station. The assumption of independency among stations with the further "load balance" property is at the basis of the analytical tractability of any queuing network [3].

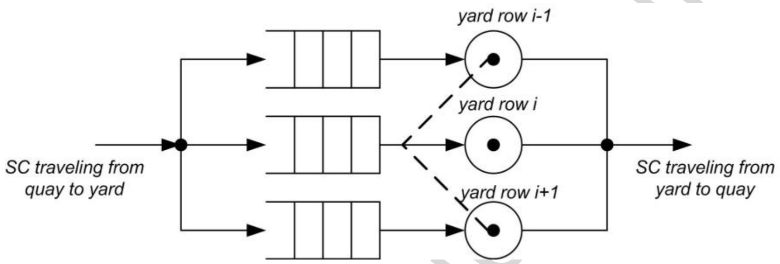


Figure 5 - Row locking on the yard

Here, unfortunately, the dependency among different queuing stations due to the mutual locking phenomenon prevents the network at hand from being covered by the celebrated BCMP theorem [3] and, therefore, solved by the exact Mean Value Analysis algorithm. To the best of our knowledge, we are the first in pointing out this modeling aspect within the context of maritime container terminals. Only relaxing the locking condition or assuming that the workload imposed to any given yard block is so low that the locking condition is unlikely to occur can provide an analytical quantitative evaluation of the queuing network at hand.

To the best of our knowledge, to date no analytical approximations have been proposed to manage the above model features that are not covered by the exact solution. Hence, our analysis could also be worth to motivate new research efforts on the approximate analytical solution of non-standard queuing networks. Here we solve the queuing network at hand by discrete-event simulation [5, 18, 24]. For an exhaustive survey of the available research literature on the application of simulation models in port development the reader to [15].

3.2. Solution by Discrete-Event Simulation

Discrete-event simulation (hereafter called simulation) is used to solve the queuing network model in Figure 2 which provides the primary functionality of the proposed model-driven DSS.

In order to simulate the queuing network of our DSS, we necessarily had to refer to a specific *worldview* [29]. Alike many researchers, we agree that worldviews are hard to compare since they describe system behavior at different levels. It is difficult to assess whether or not there is an execution rate advantage to our choice, but since we are pursuing a detailed representation of events and activities at an inner level of process modeling, we resorted to the basic *event-scheduling worldview*. By using this approach, we represented all the events that are of importance to the characteristics of the queuing network and to the overall objective of the DSS. Specifically, we defined all the system events in Table 1 and mapped every single one of them to the sub-system of the queuing network model in Figure 2 in which the events occur, being Q the quay sub-system, T the transfer sub-system and Y the yard sub-system. We then listed the other event(s) scheduled by the simulation when a specific event is executed, as well as the resources seized or released in order to carry out the corresponding action(s). For the sake of clarity, let us consider the first event *containerdischarge*. This event takes place in the quay sub-system and it requires seizing a quay crane to perform container discharge. The end of the container discharge activity schedules a *containerinbuffer* event that takes place in the same subsystem. The latter event releases the quay crane unit previously assigned and seizes a buffer space where the container is set down after being discharged. This logic goes on for the entire list of the events and by executing in order the sequence of event-action couples in Table 1, an instance of the overall container discharge process occurring in the integrated quay-transfer-yard system represented in Figure 2 is actually simulated.

Event	**Sub-system of occurrence**	**Actions**	**Resources**	
			Seize	**Release**
containerdischarge	Q	schedules *containerinbuffer*	QC	-
containerinbuffer	Q	schedules *containerhandling*	buffer space	QC
containerhandling	Q	schedules *SCcrossquay*	SC	-
SCcrossquay	T	schedules *SCsegmenttravel* or schedules *containerhandling*	- -	buffer space SC
SCbeforeintersection	T	schedules *SCafterintersection*	intersection	Road
SCafterintersection	T	schedules *SCsegmenttravel*	road	Intersection

SCsegmenttravel	T	schedules *SCatintersection* or schedules *SCarrivalatrow* or schedules *SCcrossquay*	road	Intersection
SCarrivalatrow	Y	schedules *SCdeparturefromrow*	row	road
SCdeparturefromrow	Y	schedules *SCsegmenttravel*	road	row

Table 1 - Discrete events of the simulation model

Every event in Table 1 marks the beginning or the end of a given model activity and must be counted only once. We always refer to an event as the beginning (end) of a specific activity, but, for the sake of shorter notation, we omit any “begin” (“end”) prefix (suffix) from the event name.

The time at which an event occurs is used to advance simulation time, update system state, change entity attributes and update event counters, cumulative measures related to container handling and transfer activities and, hence, statistical estimates that will be used to return statistical profiles, as well as both point and interval estimates at the end of the simulation. In the following we list the performance measures estimated by the output analysis of one or more simulation runs and specify which events are used in their point or interval estimation. It is important to underline that all event times and numbers are also recorded on dedicated Excel files which allow the user to fit both theoretical and empirical distributions and/or estimate mean, variance, skewness, probabilities, quantiles, max, min and/or any other customized summary statistics.

- The container transfer time: it is the time it takes an SC to transfer a container from the bottom of a quay crane to a yard row or vice versa. It is computed as the difference between *SCarrivalatrow.time* and *SCcrossquay.time*, where *.time* is an attribute marking the instant at which the specific even occurs.
- The proportion of an SC waiting at an intersection: it measures the chance of an SC queuing at an intersection of the yard road grid during its travel between the yard and quay sub-systems. The related counters are updated by a call to the *SCbeforeintersection* event.
- The SC waiting time at an intersection: it is the time an SC queues at an intersection of the yard road grid during its travel between the yard and quay sub-systems. It is computed as the difference between *SCbeforeintersection.time* and *SCafterintersection.time*.

- The proportion of an SC waiting in front of a yard row: it measures the chance of an SC queuing in front of a yard row because the row is either busy or locked. The related counters are updated by a call to the *SCarrivalatrow* event.
- The SC waiting time in front of a yard row: it is the time an SC queues in front of a yard row because the row is either busy or locked. The related counters are updated by a call to the *SCarrivalatrow* event.
- The proportion of an SC waiting in front of a QC: it measures the chance of an SC queuing in front of a quay crane because at least another SC is already picking up or setting down a container in the buffer area of that crane. The related counters are updated by a call to the *containerhandling* event.
- The SC waiting time in front of a quay crane: it is the time an SC queues in front of a quay crane. It is computed as the difference between *containerhandling.time* and *containerinbuffer.time*.
- The proportion of QC blocking: it measures the chance of a quay crane getting blocked when discharging a container because there is no available space for setting the container down in its buffer space. The related counters are updated by a call to the *containerinbuffer* event.
- The QC blocking time: it is the time duration a quay crane remains blocked. It is computed as the difference between *containerdischarge.time* and *containerinbuffer.time*.

Thanks to the law of large numbers the proportions listed above become reliable point estimates for probabilities as the related sample size goes large enough within a simulation run; however, using the binomial model we may also return interval estimates by following the recommendations in [4].

With respect to all the remaining output measures of waiting times and blocking times listed above, model simulation returns again both point and interval estimates of the true waiting times and blocking times in the real system. In this respect, to keep the unavoidable estimation error under control and low enough, we adopted a method that allows us to determine the appropriate number of simulation runs (replications) for whatever simulation experiment of interest for the DSS user. Specifically, according to Nakayama [27], we defined in advance a maximum acceptable relative error value $\epsilon > 0$

(e.g. value $\epsilon = 0.1$), where the estimator returned by the simulation should be within $100\epsilon\%$ of the correct value with high probability $100(1 - \delta)\%$ (e.g. 90%). To this purpose, we adopted a two-stage procedure: in stage 1, both the sample mean ($\bar{\mu}$) and sample variance (S^2) of the real performance measure (μ) are estimated from n_0 ($n_0 \geq 10$) pilot runs; in stage 2, $N_r(\epsilon)$ additional independent production runs are generated

$$N_r(\epsilon) = \frac{\left[z_{1-\delta/2}^2 \cdot S^2(n_0)\right]}{\bar{\mu}^2 \cdot \epsilon^2}$$

and the new sample mean and sample variance obtained from these runs are used to obtain an estimate within $100\epsilon\%$ of the correct value with high probability $100(1 - \delta)\%$. We also considered variance reduction techniques with the purpose of using less simulation effort [22].

3.3. The DSS Interface

Microsoft Visual Basic was used to develop the user-friendly interface between the system and the operations manager. The system GUI, which is illustrated in Figure 6, is practically divided into three major sections for defining the terminal areas and the equipment used in these areas. Two other minor sections are dedicated to reading and writing technical data with respect to model simulation and on-line results, respectively.

In the first section, the berth is defined by one long segment whose overall length can be expressed by specifying a given number of bollards (i.e. short posts along the berth round which vessel ropes are fastened). Since the berth is equipped with quay cranes that perform container discharge/loading operations between berthed vessels and the quay area, bollards are also used to identify the actual physical location of each single quay crane.

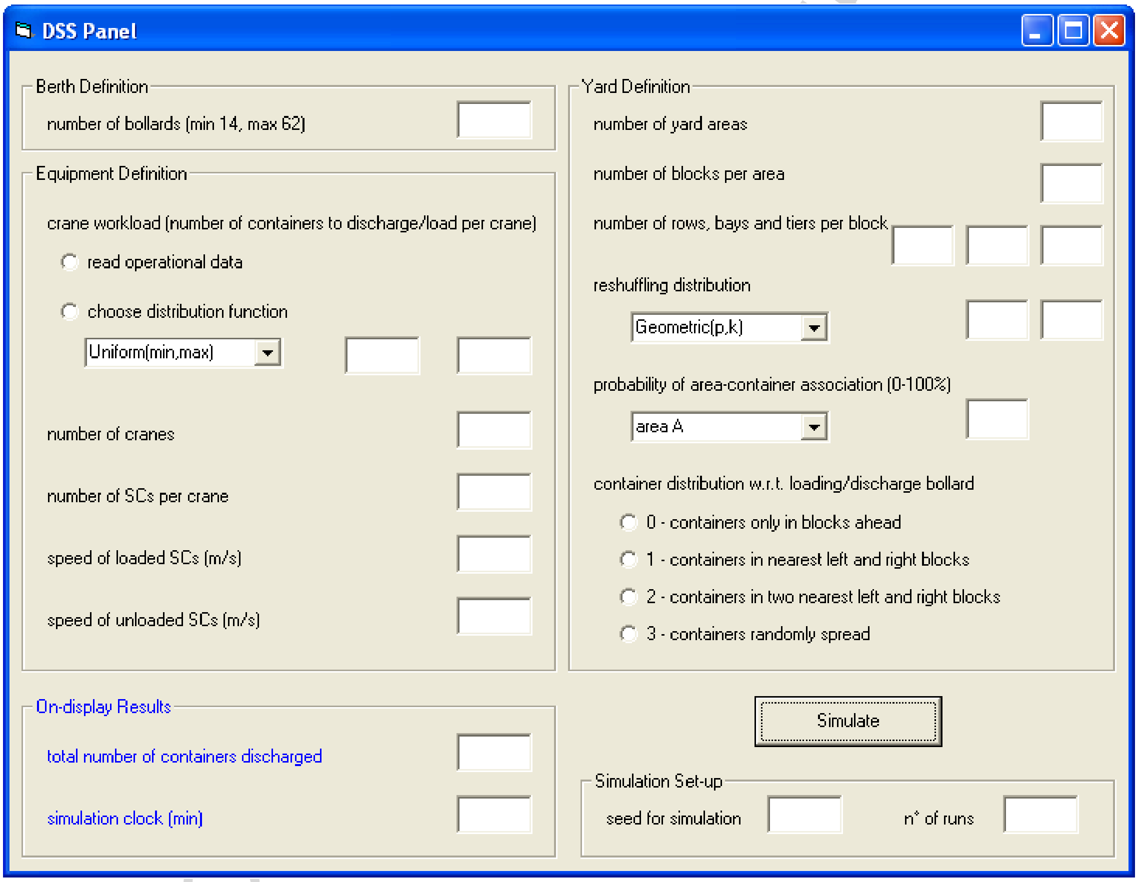
The yard definition section accounts for both yard organization and, with respect to the horizontal and vertical distance from the discharge/loading points, container stacking and retrieval policies. To characterize the physical layout of the yard, the user must define the number of areas in which the yard is divided (e.g. if three, areas are labeled with the letters A, B and C) and the number of blocks in each area. The yard definition is completed by specifying block composition in terms of number of rows, bays and tiers, as well as the distribution for container reshuffling. Reshuffles are unfruitful container movements to be performed during container retrieval on the yard when the target container is located on a ground slot or intermediate position underneath other containers [21]. A discrete empirical distribution is used: its values depend on the number of tiers featured by a block and the average occupation level of the yard. As for yard organization, the specific container stacking and retrieval policy adopted by the terminal can be defined by setting two factors: the horizontal distance and the vertical distance standing between the stacking/retrieval point on the yard and the D/L point on the quay. After the number of yard areas has been inserted, the vertical distance factor is defined by specifying the area-container association probabilities: the first area (i.e. area A) is the closest to the quay, the last is the farthest away. Instead, the horizontal distance between the yard block and the D/L point is represented by choosing one of the four options of the so-called container spreading policy:

- policy 0 - containers are transferred to/from the yard block(s) right in front of the D/L point;
- policy 1 - containers can also be transferred to/from the yard block(s) immediately to the left and to the right of the block(s) in front of the D/L point;
- policy 2 - containers can also be transferred to/from the two nearest yard blocks to the left and to the right of the block(s) in front of the D/L point;
- policy 3 - containers can be randomly transferred to/from any yard block.

In the equipment definition section, besides defining the number of quay cranes and the number of SCs to be assigned per quay crane, the user has two different alternatives to define the crane workload, i.e. the number of containers to discharge/load per quay crane. This number can either be read via Excel from a discharge/loading plan usually provided by the planning office (see Figure 1) or be chosen from a drop-down menu containing several discrete distribution functions. As for the additional vehicle information required, the user can insert two alternative nominal-free SC speeds: one for loaded SCs, the other for unloaded SCs (see Figure 4). The vehicle-driver system is modeled individually and the driver’s behavior is based on a (discrete) vehicle-following logic [19]. During actual container transportation, drivers move according to an *en route* rationale, meaning that they are provided with travel-related information after they start their trip and in real-time [9]. A driver tries to minimize the total travel time by choosing the less congested segments during his/her travel. As for the actual transportation speed of an SC, it may vary between the point of container discharge/loading along the quay and the specific row of a given block located in one of the yard areas in which the container is planned to be stored/retrieved. It is computed according to Greenshields' model [20]

$$v = v_f - \left(\frac{v_f}{k_j}\right)k$$

where v is the mean speed at density k, v_f is the free speed and k_j is the jam density.



choose distribution function: Uniform(min,max)

reshuffling distribution: Geometric(p,k)

probability of area-container association (0-100%): area A

Figure 6 – GUI panel of the DSS

The user completes the input stage by providing first the simulation seed to generate a sequence of random numbers that are used to generate both event occurrence times (e.g. for resource seize and release) and random deviates for related activities (e.g. handling operations) and then number of simulation runs to be executed for the defined simulation experiment (see section 3.2).

On the output side, the on-display results returned by the simulator are intended to give some quick information related to the end of the simulation run(s): the (average) number of containers discharged/loaded and the (average) time required to complete all container D/L operations.

4. Numerical Experiments

In this section we present two sets of numerical experiments. The first set is devoted to verification and validation issues focused on mimicking container handling on the quay and yard with an adequate degree of credibility and accuracy. After describing the scenario setting for a real SC-based transshipment terminal, the second set of numerical experiments is carried out for illustrative purposes: within the specific case study, what-if experiments are performed to determine the extent to which the operations manager may benefit of a distributed rather than concentrated container storage policy on the yard.

4.1. Verification and Validation

The design and implementation of the simulation model in the model-driven DSS has been carried out in compliance with all the conventional steps used to guide a thorough and sound simulation study [2]. Due to the importance of having confidence in the “predictive” capability of the model, in the following we account for the verification and validation steps of the study.

Model verification is often defined as “ensuring that the computer program of the computerized model and its implementation are correct”; model validation is usually defined as “substantiation that a computerized model with its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” [34]. These are the two definitions adopted here.

4.1.1. Verification

Three classes of verification techniques have been used in order to verify if the input parameters and logical structure of the queuing network are correctly represented by the simulation model. To serve this purpose, common-sense techniques, thorough documentation and traces have been applied on a set of *ad hoc* instances in which container discharge scenarios have been defined according to the settings reported in Table 2. These settings cover from the physical features of the system to the equipment number and characteristics in each area of the system. For the sake of convenience, we verify a sample scenario focused on discharge operations and show the central value of the interval estimates for immediate comparison.

Feature	Number	Characteristics
berth	20 bollards	24 meters per bollard
QCs	2 rail-mounted QCs	max discharge rate 28 containers/h
workload	2074 containers	1030 QC1, 1044 QC2
SCs	4 per QC	max 4 m/s loaded, 5 m/s unloaded
yard	4 areas	32 rows, 16 bays and 3 tiers per block
reshuffles	no reshuffles	empirical(p_1=1,v_1=0; p_2=0,v_2=1; p_3=0,v_3=2)
containers	area destination	100% area A vs 100% area D
containers	block destination	policy 3

Table 2 - System settings for verification experiments

This scenario is one of the dynamic tests (i.e. investigation of input-output relations) designed to verify the different impact of discharging only 1-TEU containers to area A (i.e. the closest to the quay), rather than 1-TEU containers to area D (i.e. the farthest from the quay). In these experiments, one may expect significant differences in container transfer time, as well as a different number of intersections being crossed according to container destination.

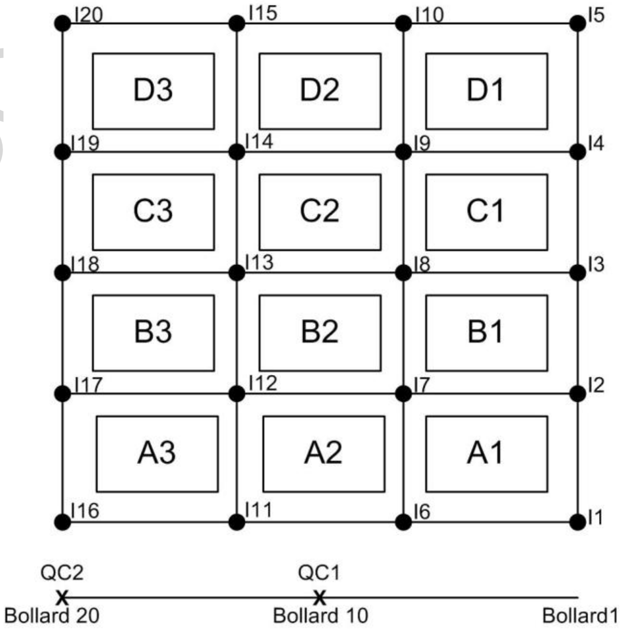


Figure 7 - System layout for scenario based on different container types

A representation of the overall berth-quay-yard segment is given in Figure 7 to favor a better understanding of the location of blocks, road segments and intersections with respect to the given berthing positions of quay crane 1 (QC1 at bollard 10) and quay crane 2 (QC1 at bollard 20).

The results of this set of simulation experiments over multiple runs are illustrated in Tables 3 and 4 with container destination being areas A and D, respectively. Besides the overall time (CT) to complete the discharge of 2074 containers, the two tables report on probabilities and times referred to both SCs and QCs. The waiting probability (WP) of SCs at a given intersection is provided along with the corresponding average waiting time (WT in seconds). The blocking probability (BP) of QCs is provided along with the corresponding average blocking time (BT in seconds).

SCs	CT	SCS at Intersections			QCs on Berth		
		n°	WP	WT	n°	BP	BT
1	4110	6	0.016	2.31	1	0.83	53.38
		7	0.007	1.83			
		11	0.014	1.80	2	0.88	150.2
		12	0.004	0.99			
2	2096	6	0.044	2.58	1	0.32	61.54
		7	0.030	1.92			
		11	0.054	2.26	2	-	-
		12	0.018	2.09			
3	1754	6	0.045	2.46	1	-	-
		7	0.015	1.94			
		11	0.060	2.36	2	-	-
		12	0.017	2.30			
4	1741	6	0.048	2.27	1	-	-
		7	0.028	1.65			
		11	0.066	2.36	2	-	-
		12	0.017	1.84			

Table 3 - Results for scenario A=100% based on different container types

As one may observe, the greater the number of SCs, the greater the probability of waiting at an intersection. In the case in which 100% of the containers are discharged to area A, SCs are expected to wait only at the intersections located along the lower levels of the roadway grid (i.e. intersections 6, 7, 11, 12). On the other hand, when 100% of the containers are discharged to area D, SCs are expected to wait at intersections located along every level of the roadway grid (from intersection n°6 to n°20). The distance to travel and the time required by a given number of SCs during container discharge also affect the blocking probability and the corresponding blocking time suffered by the quay cranes. In case A=100%, QCs are expected to avoid blocking when 3 or more SCs are assigned to each QC for container discharge operations to area A. This number of SCs rises to 4 when the containers to be discharged require transfer for storage in area D due to the greater time required to cycle between the quay and yard areas.

SCs	**CT**	**SCs at Intersections**									**QCs on Berth**		
		n°	**WP**	**WT**	**n°**	**WP**	**WT**	**n°**	**WP**	**WT**	**n°**	**BP**	**IT**
1	6481	**6**	0.003	2.89	**11**	0.010	1.63	**16**	-	-	**1**	0.987	209.8
		7	0.004	2.23	**12**	0.007	2.21	**17**	-	-			
		8	0.003	2.37	**13**	0.005	2.91	**18**	-	-	**2**	0.992	272.1
		9	0.013	2.02	**14**	0.002	2.87	**19**	-	-			
		10	0.006	2.15	**15**	0.010	2.48	**20**	-	-			
2	3246	**6**	0.025	2.39	**11**	0.031	2.11	**16**	-	-	**1**	0.663	82.16
		7	0.014	2.44	**12**	0.014	1.96	**17**	0.002	0.97			
		8	0.021	2.21	**13**	0.019	1.93	**18**	0.005	2.40	**2**	0.716	119.2
		9	0.025	2.69	**14**	0.021	2.34	**19**	0.002	2.85			
		10	0.014	1.99	**15**	0.010	1.84	**20**	-	-			
3	2178	**6**	0.040	2.34	**11**	0.053	2.20	**16**	0.009	2.01	**1**	0.115	31.97
		7	0.041	2.43	**12**	0.043	2.05	**17**	-	-			
		8	0.033	2.16	**13**	0.034	2.18	**18**	0.006	2.17	**2**	0.420	56.14
		9	0.041	2.17	**14**	0.031	2.08	**19**	0.003	3.89			
		10	0.027	1.98	**15**	0.020	2.02	**20**	-	-			
4	1723	**6**	0.048	2.40	**11**	0.077	2.50	**16**	0.003	4.00	**1**	-	-
		7	0.046	2.07	**12**	0.052	2.34	**17**	0.011	2.31			
		8	0.044	2.33	**13**	0.045	2.35	**18**	0.005	1.02	**2**	-	-
		9	0.050	2.48	**14**	0.035	2.35	**19**	0.008	2.49			
		10	0.021	1.97	**15**	0.010	1.43	**20**	0.005	0.90			

Table 4 - Results for scenario D=100% based on different container types

4.1.2. Validation

Three classes of validation techniques have been used to validate the sub-models and the overall DSS model [33]. Both subjective and objective tests have been used to compare the model and its behavior to the real system and its behavior. The former usually require “educated” people to make judgments about the model and its output; the latter always require data on the behavior of both the system and the model. In this specific case we have considered:

- comparison to other models (objective test) – various results (e.g. outputs) of the simulation model being validated are compared to results of other valid models;
- face validity (subjective test) – individuals knowledgeable about the system are asked whether the model and/or its behavior are reasonable;
- historical data validation (objective test) – if historical data exists, part of the data is used to build the model and the remaining data is used to determine whether the model behaves as the system does.

Provided that the entire model-driven DSS was developed in close cooperation with the senior management and the technical staff of a real container terminal, here we present the tests based on the comparison to other models and those based on historical data validation rather than face validity.

The comparison to other models was feasible on the basis of two decades of simulation experience with the terminal company. In this particular case, we validated the quay sub-system by considering the completion time of the quay cranes (i.e. the time required to complete container discharge/load operations) and performing a three-way comparison of the results returned by the DSS simulator, by a pre-existing valid model (D/L simulator) [5] and by the terminal’s operations manager (Terminal). The system settings for this specific validation experiment (n°1) are reported in Table 5.

Feature	Parameter	Characteristics
berth	20 bollards	24 meters per bollard
QCs	3 rail-mounted QCs	max discharge rate 28 containers/h
workload	891 containers	258 QC1, 366 QC2, 267 QC3
SCs	4 per QC	max 4 m/s loaded, 5 m/s unloaded
Yard	4 areas	32 rows,16 bays and 3 tiers per block
reshuffles	no reshuffles	empirical(p_1=1,v_1=0; p_2=0,v_2=1; p_3=0,v_3=2)
containers	area destination	60% area A, 20% area B, 15% area C, 5% area D
containers	block destination	policy 0

Table 5 - System settings for validation experiment n°1

As one may see, three quay cranes are assigned to a vessel to perform both discharge and loading activities for 891 containers. The containers are assigned to different areas, while the blocks are all in front of the D/L points (policy 0). No container reshuffling on the yard is required because of previously performed housekeeping operations [11]. Observing the 90% confidence intervals in Table 6 generated by the DSS simulator, the behavior of the model clearly mirrors the real behavior of the quay sub-system.

Source	**Quay crane completion time (h)**		
	QC1	**QC2**	**QC3**
Terminal	9.25	15.17	10.25
D/L simulator	[9.13–9.32]	[15.12–15.39]	[10.24–10.47]
DSS simulator	[9.23–9.36]	[15.25–15.42]	[10.19–10.36]

Table 6 – Validation of the quay sub-system

For the purpose of validating the transfer sub-system, we used historical data pertaining to the time required to transfer discharged containers from a vessel to the yard. The system settings for this specific validation experiment (n°2) are reported in Table 7.

Feature	**Parameter**	**Characteristics**
berth	20 bollards	24 meters per bollard
QCs	5 rail-mounted QCs	max discharge rate 28 containers/h
workload	420 containers	119 QC1, 32 QC2, 164 QC3, 32 QC4, 73 QC5
SCs	4 per QC	max 4 m/s loaded, 5 m/s unloaded
yard	4 areas	32 rows,16 bays and 3 tiers per block
reshuffles	no reshuffles	empirical(p_1=1,v_1=0; p_2=0,v_2=1; p_3=0,v_3=2)
containers	area destination	27% area A, 49% area B, 5% area C, 19% area D
containers	block destination	policy 0

Table 7 - System settings for validation experiment n°2

In this case, five quay cranes are assigned to a vessel to discharge 420 containers. The containers are to be stacked in different areas, while the blocks are all in front of the D/L points (policy 0). No container reshuffling on the yard is required for container discharge. The ExpertFit distribution-fitting software [16] was used to specify, rank and then choose the “best” probability distribution for the statistical data recorded for the real container round-trip transfer times (Terminal), as well as the data returned by the simulator (DSS simulator). Table 8 shows this comparison according to which the best candidate distribution (under the Kolmogorov-Smirnov statistical test) is actually a Beta in both cases, bearing a similar mean (in seconds) and standard deviation for the transfer times.

Source	Mean (s)	Standard Deviation	Model	Score	Goodness-of-fit Test
Terminal	216.4	52.36	Beta	100%	K-S
DSS simulator	213.4	50.07	Beta	98.9%	K-S

Table 8 – Validation of the transfer sub-system

The results from these and other verification and validation experiments make us confident of the correctness and credibility of the simulation model in our DSS.

4.2. Case Study

The case study presented herein is based on our study related to the model-driven DSS designed and implemented in cooperation with the terminal company that manages the transshipment hub located at the port of Gioia Tauro in southern Italy.

4.2.1. Scenario Definition

At the container terminal in Gioia Tauro the berth is divided into 24-meter spaced bollards and it is equipped with rail-mounted gantry cranes. The yard runs parallel to the berth and starts immediately after the quay area. Thus, both storage and transportation services occur on the yard.

For storage purposes, the yard is divided into four areas known as A, B, C and D. With very rare exceptions to the rule, 20-foot (a.k.a. 1 TEU - twenty equivalent units) containers are stored in A, 40-foot (a.k.a. 2 TEUs) containers are stored in B, refrigerated containers (a.k.a. refers) are stored in C and empty containers are stored in D. Whatever be the area, containers are arranged perpendicularly to the quay, mainly due to the SC-based yard technology available at the terminal. Each area features a given number of container storage blocks. A block, in turn, is organized in 32 rows (or lanes), 16 bays and 3 tiers of height.

Figure 8 shows an example of how the terminal's stacking area is organized in blocks. The dotted lines are examples of paths (from quay position Q_i to yard position Y_i, i=1..4) a man-guided SC may follow during container transfer between source and destination points. These movements are not always the same since at intersections (see, for example, intersection 1 along path Q_4Y_4) the driver decides in real time which segment to take according to an *en route* logic. So, his/her decision depends on both the destination point and the traffic encountered along the alternative directions in that very moment. As a result, SC driver movements change from trip to trip and the overall path followed by the driver is built accordingly.

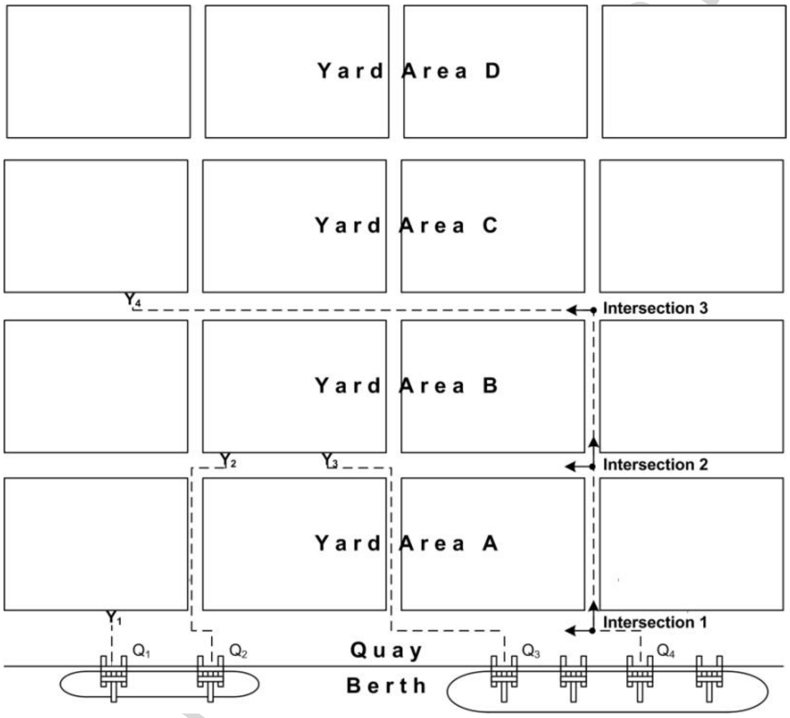


Figure 8 – Terminal layout and SC driver behavior

In this roadway grid network where corridors form squares centered on the container blocks, two SCs, one in each direction, can travel along the two-lane corridor where no overtaking occurs (i.e. no lane changing is implemented). The free speed of the SC is 20km/h when unloaded and 15km/h when loaded. At intersections, the driver queues before the intersection point until it becomes available for vehicle crossing (i.e. no gap acceptance). Once the SC reaches and accesses the target row for container storage/retrieval, the time required by the SC to perform container handling inside the row can be estimated by the sum of two terms. The first term accounts for the distance that the SC must travel within the row to reach the target bay and, thus, it varies from 1 to 16. The second term depends on the type of operation to be performed: during container stacking, the container is likely to be set down on top of the container stack, whereas during container retrieval the target position may be a ground or an intermediate slot. This may require 1 or 2 container reshuffles if other containers are stacked above the container of interest.

4.2.2. What-if Experiments

To fully understand how planners can use the DSS in real-life operations management, let us suppose that a considerable amount of 40-foot containers has been scheduled for discharge in eight different D/L points. The containers are bound to be transferred towards the same block in the so-called area B of the yard. Bearing in mind the specific layout of the yard, the operations manager seeks support in deciding which of the following two alternative container spreading policies is more performing in terms of quay-yard cycling time. The quay-yard cycling time is the time it takes an SC to transfer a container to its stacking position on the yard and return to the quay. Thus, it includes the two-way transfer time, the waiting time in front of the yard row and the handling time within the row. Since the handling time within the row of a specific container does not change whatever be the row, it can be omitted in the comparison of the two different policies. If all the containers are transferred to the same yard block in area B right in front of the D/L points (i.e. policy 0 in the yard definition panel in Figure 6), the average transfer time should stay “small” because of the limited distance to cover from the quay to the yard and vice versa. However, the overall quay-yard cycling time is likely to grow due to both the waiting times and locking times caused by the large number of SCs concentrated in the same block. On the other hand, containers can also be transferred to the blocks immediately to the left and to the right of the block in area B right in front of the D/L points (i.e. policy 1 in the yard definition panel in Figure 6). In this option, the distance to cover should grow larger, but this disadvantage should be compensated by smaller waiting times and locking times. As a matter of fact, when spreading the containers over a greater number of yard rows, the SCs should benefit of an earlier access to the rows for container stacking.

Table 9 summarizes the settings of this particular scenario in terms of resources and policies involved, as well as statistical distributions adopted in process modeling. The SC free speed is taken from the vehicle's fact sheet.

Feature	Number	Characteristics
berth	20 bollards	24 meters per bollard
QCs	8 rail-mounted QCs	16-order Erlang(avg=2min) with a 94% score, not rejected by K-S test
workload	40-foot containers	Uniform(min=1000,max=1200) with a 90% score, not rejected by Chi-square test
SCs	4 per QC	max 4 m/s loaded, 5 m/s unloaded
yard	4 areas	32 rows,16 bays and 3 tiers per block
reshuffles	no reshuffles	empirical(p_1=1,v_1=0; p_2=0,v_2=1; p_3=0,v_3=2)
containers	area destination	100% area B
containers	block destination	policy 0 vs policy 1

Table 9 - Scenario settings for alternative container spreading policies

Observe that, in a similar scenario, the weight of the waiting times and locking times of the SCs on the yard, as well as the blocking times of the quay cranes should be considered with care by the operations manager when evaluating which container storage policy carries a better payoff with respect to quay-yard cycling time. In particular, as shown in Table 10, if one was to relax, for instance, the locking condition on adjacent yard rows, this assumption would return a poor estimate of the probability of waiting in front of a yard row. In real-life conditions, this probability would be equal to 0.536 where 0.193 is due to a busy yard row and 0.343 is due to a locked yard row. In other words, the relaxed model would assume this probability of waiting to be approximately one-third of its actual value. Observe that, in a similar scenario, the weight of the waiting times and locking times of the SCs in front of the yard rows, as well as the blocking times of the quay cranes should be considered with great care by the operations manager when evaluating which container storage policy carries a better payoff with respect to quay-yard cycling time. Table 10 shows by comparison the difference between estimating by simulation the probability of waiting in front of a yard row with (real case) or without (simplified case) considering the locking condition caused by SCs in adjacent rows. Under real operational conditions, the probability of an SC waiting in front of a row is equal to 0.536, where 0.193 is due to a busy yard row and 0.343 is due to a locked yard row. Using a model which does not account for the locking condition would return a very poor estimate (0.193). In other words, the relaxed model would assume this probability of waiting to be approximately one-third of its actual value.

Performance	Real Case	Simplified Case
P(SC waits in front of a row)	0.536	0.193
P(busy row|SC waits)	0.193	0.193
P(locked row|SC waits)	0.343	0.0

Table 10 - Detection of the queuing phenomena on the yard

This stated, in Table 11 we can appreciate the contribution of both the (average) transfer time and the (average) waiting time in front of a yard row, whether it be busy or locked, when estimating the yard-quay cycling time for policies 0 and 1. Observe that the transfer times include the waiting times at the road intersections crossed (see Table 12) to reach the destination block in area B for every single container. In this particular scenario, policy 1 seems to be more competitive (i.e. 253s versus 289s). Here the disadvantage of greater transfer times is more than offset by the advantage of having a smaller waiting time in front of a row by 40%. In addition, policy 1 outperforms policy 0 in terms of the probability of finding more than one vehicle already waiting in front of a row (see Table 13): the smaller the number of vehicles, the less the waiting time.

Performance	Policy 0	Policy 1
average quay-yard cycling time	288.8	253.1
average transfer time	93.65	95.44
average return transfer time	88.76	93.44
average waiting time in front of a row	106.4	64.22

Table 11 - Comparison of average quay-yard cycling time (s)

Policy 0			Policy 1		
Int. n°	**WP**	**WT**	**Int n°**	**WP**	**WT**
1	0.009	1.71	**1**	0.051	2.31
2	0.000	0.00	**2**	0.021	2.43
3	-	-	**3**	0.015	0.84
4	-	-	**4**	-	-
5	-	-	**5**	-	-
6	0.341	3.15	**6**	0.354	4.60
7	0.369	2.98	**7**	0.305	2.66
8	0.166	2.30	**8**	0.160	2.31
9	-	-	**9**	-	-

10	-	-	**10**	-	-
11	0.158	2.47	**11**	0.183	2.67
12	-	-	**12**	0.209	2.49
13	-	-	**13**	0.117	2.34

Table 12 - Comparison of waiting probabilities and average times (s) at road intersections (Int)

Performance	Policy 0	Policy 1
P(finding n=0 SCs already waiting)	0.749	0.913
P(finding n=1 SC already waiting)	0.175	0.087
P(finding n=2 SCs already waiting)	0.062	-
P(finding n=3 SCs already waiting)	0.015	-

Table 13 - Comparison of n° of SCs found already waiting in front of a yard row

For the sake of completeness, in addition to the average values reported throughout the tables above, Table 14 indicates the related distribution type and the goodness-of-fit test (either Kolmogorov-Smirnov or Anderson-Darling) adopted in this particular scenario (i.e. all containers are transferred to area B). It also includes the number of simulation replications required to keep the unavoidable estimation error under control and within $100\epsilon\%$ ($\epsilon = 0.05$) of the real value under estimation.

Output Data	Distribution	Score	Test	N° runs
waiting time at intersections	Weibull	100%	K-S	34
waiting time at rows	Gamma	98%	K-S	47
container transfer time	Gamma	95%	A-D	10

Table 14 – Distribution profiles for simulation output

As for a final evaluation, the operations manager examines the effect of these policies on the activities performed in the quay area. Quay cranes are, in fact, the most expensive equipment in a container terminal and their productivity measures the efficiency of the overall discharge-loading process. Here policy 1 outperforms policy 0 once again. In the former case, no QC is blocked during container discharge, whereas in the latter crane blocking may occur. Fortunately, it occurs according to a small probability (i.e. 0.02) and the average blocking time is estimated to be in the order of 45 seconds (that's why no fitting efforts have been carried out and reported in Table 14). Crane blocking occurs within those time intervals where the arrival rate of the group of SCs assigned to a given quay crane is smaller than the quay crane's service rate. As a result, it is preferable to manage SC cycling with the aim of reducing enough the time intervals in which no SCs are waiting in front of the cranes, rather than having larger time intervals in which cranes may be blocked because of an SC shortage. Even with respect to this issue, policy 1 is expected to perform better than policy 0, as shown in Table 15.

Performance	Policy 0	Policy 1
P(finding n=0 SCs already waiting)	0.559	0.335
P(finding n=1 SC already waiting)	0.400	0.573
P(finding n=2 SCs already waiting)	0.040	0.092
P(finding n=3 SCs already waiting)	0.001	0.001

Table 15 - Comparison of n° of SCs found already waiting at the feet of the assigned crane

5. Conclusions

Decisions on the integrated management of container handling operations in a maritime container terminal stem from a real environment where events and logistic activities occur at random. So, the terminal manager asks for a proper supporting model to quantify the effects of randomness upon an efficient integrated management. We have shown that the quay crane blocking and starvation phenomena during container discharge/loading, as well as the vehicle interaction during internal container transfer and, finally, the yard row locking policy in the storage area can be captured by a queuing network model. This has been put at the basis of the development of a (queuing) model-driven DSS. Once that the difficulty of pursuing any practical analytical solution of the driving queuing network has been discussed, solution efforts have been steered towards the use of discrete-event simulation.

The proposed DSS has been conceived to support operations management at a generic container terminal of pure transshipment equipped with self-lifting shuttle vehicles. DSS capability in representing with great detail the trips of the self-lifting shuttle vehicles devoted to container handling and transfer between the quay and yard has been successfully validated and put at work in the transshipment hub located at the port of Gioia Tauro in Italy.

The major related numerical example presented has allowed to quantitatively evaluate the extent to which the terminal manager may benefit of a distributed rather than concentrated container storage policy on the yard, under heavy container discharge operations concentrated within a limited segment of the quay. Supported by our system, the distributed policy adopted at the container terminal in Gioia Tauro has shown to reduce the vehicle waiting phenomenon due to row locking on the yard and it guarantees a faster return of the vehicles to the cranes on the quay. To appreciate this, one may consider that the cost per hour of a vehicle is computed by considering the cost of consumables (i.e. fuel and tires), personnel and maintenance. At the container terminal in Gioia Tauro it is estimated to be €58.10/h per vehicle. Overall, the integrated management supported by our system reduces the crane blocking phenomenon, improves the quay crane productivity and, thus, reduces the vessel turnaround time which is especially appreciated by the shipping liners in terms of quality of service.

Acknowledgments

The authors would like to thank Carmine Crudo, former General Manager of Medcenter Container Terminal SpA, and all the technical staff for a number of valuable information, suggestions and data retrieval since the beginning of our work.

References

[1.] Bandeira, D. L., Becker, J. L. & Borenstein, D. (2009). A DSS for integrated distribution of empty and full containers. *Decision Support Systems*, *47*, 383–397. DOI: 10.1016/j.dss.2009.04.003.

[2.] Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2000). *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

[3.] Baskett, F., Chandy, K. M., Muntz, R. R., & Palacios, F. (1975). Open, closed, and mixed networks of queues with different classes of customers. *ACM Journal*, *22*(*2*), 248–260. DOI: 10.1145/321879.321887.

[4.] Brown L. D., Cai T. T., & DasGupta, A. (2001) Interval estimation for a binomial proportion. *Statatistical Science*, 16, 101–133. DOI: 10.1214/ss/1009213286.

[5.] Canonaco, P., Legato, P., Mazza, R. M., & Musmanno, R. (2008). A queuing network model for the management of berth crane operations. *Computers & Operations Research*, 35, 2432 – 2446. DOI: 10.1016/j.cor.2006.12.001.

[6.] Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014a). Storage yard operations in container terminals: literature overview, trends, and research directions. *European Journal of Operational Research*, 235, 412–430. DOI: 10.1016/j.ejor.2013.10.054.

[7.] Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014b). Transport operations in container terminals: literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, 236(1), 1–13. DOI: 10.1016/j.ejor.2013.11.023.

[8.] Chandy, K. M., Herzog, U. & Woo, L. (1975). Parametric analysis of queueing network models. *IBM Journal of Research & Development*, 1, 36–42.

[9.] Chowdhury, M.A. & Sadek, A.W. (2003). *Fundamentals of Intelligent Transportation Systems Planning*. Boston, Massachusetts: Artech House.

[10.] Containerisation International and Lloyd’s List, 2013. One hundred ports - The world’s busiest container terminals.

[11.] Cordeau, J.-F., Legato, P., Mazza, R. M., & Trunfio, R. (2015). Simulation-based optimization for housekeeping in a container transshipment terminal. *Computers & Operations Research*, 53, 81–95. DOI: 10.1016/j.cor.2014.08.001.

[12.] Delgado, A., Jensen, R. M., & Guilbert, N. (2012). A placement heuristic for a commercial decision support system for container vessel stowage. In *Proceedings of the 38th Latin America Conference on Informatics*, 1–9.

[13.] Dereli, T. & Daş, G. S. (2010). Development of a decision support system for solving container loading problems. *Transport*, *25*(*2*), 138–147. DOI: 10.3846/transport.2010.17.

[14.] Ding, Y., Wei, X.–J., Yang, Y., & Gu, T.-Y. (2017). Decision support based automatic container sequencing system using heuristic rules. *Cluster Computing*, *20*, 239–252. DOI: 10.1007/s10586-016-0678-2.

[15.] Dragović, B., Tzannatos, E., & Park, N. K. (2017). Simulation modelling in ports and container terminals: literature overview and analysis by research field, application area and tool. Flexible Services and Manufacturing Journal, 29, 4–34. DOI: 10.1007/s10696-016-9239-5.

[16.] ExpertFit Version 6.00 – Univesity Edition Copyright © 1995-2004 by Averill M. Law.

[17.] Fazi, S., Fransoo, J. C., & Van Woensel, T. (2015). A decision support system tool for the transportation by barge of import containers: a case study. *Decision Support Systems*, *79*, 33–45. DOI: 10.1016/j.dss.2015.08.001.

[18.] Fleming, M., Huynh, N., & Xie, Y. (2013) Agent-based simulation tool for evaluating pooled queue performance at marine container terminals. *Transportation Research Record*, 2330, 103–112. DOI: 10.3141/2330-14.

[19.] Gazis, D. C., Herman, R. & Potta, R. E. (1959). Car-following theory of steady state traffic flow. *Operations Research*, *7*, 499–505. DOI: 10.1287/opre.7.4.499.

[20.] Greenshields, B. D. (1935). A study of highway capacity. In *Proceedings Highway Research Record Volume 14* (pp. 448-477). Washington D.C.

[21.] Legato, P. & Mazza, R. M. (2013). Managing container reshuffling in vessel loading by simulation. In R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, & M.E. Kuhl (Eds.), *Proceedings of Proceedings of the 2013 Winter Simulation Conference* (pp. 3450–3461). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[22.] Legato P., Mazza, R. M., & Gullì, D. (2014). Integrating tactical and operational berth allocation decisions via simulation–optimization. *Computers & Industrial Engineering*, 78, 84–94. DOI: 10.1016/j.cie.2014.10.003.

[23.] Liu, Y., Zhou, C., Guo, D., Wang, K., Pang, W. & Zhai, Y. (2010). A decision support system using soft computing for modern international container transportation services. *Applied Soft Computing*, *10*, 1087–1095. DOI: 10.1016/j.asoc.2009.06.015.

[24.] Mavrakis, D., & Kontinakis, N. (2008). A queueing model of maritime traffic in Bosporus straits. *Simulation Modelling Practice and Theory*, 16(3), 315–328. DOI: 10.1016/j.simpat.2007.11.013.

[25.] Moghadam, M. K., Jahromi, A. R. M., & Nooramin, A. S. (2011). A fuzzy AHP decision support system for selecting yard cranes in marine container terminals. *WMU Journal of Maritime Affairs*, *10*, 227–240. DOI: 10.1007/s13437-011-0007-9.

[26.] Murty, K. G., Liub, J., Wan, Y.-W., & Linn, R. (2005). A decision support system for operations in a container terminal. *Decision Support Systems*, *39*, 309– 332. DOI: 10.1016/j.dss.2003.11.002.

[27.] Nakayama, M. (2006). Output analysis for simulation. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, & R. M. Fujimoto (Eds.), *Proceedings of the 2006 Winter Simulation Conference*, (pg. 36–46). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[28.] Ngai, E. W. T., Li, C.-L., Cheng, T. C. E., Lun, Y. H. V., Lai, K.-H., Cao, J. & Lee, M. C. M. (2011). Design and development of an intelligent context aware decision support system for real-time monitoring of container terminal operations. *International Journal of Production Research*, *49*(*12*), 3501–3526. DOI: 10.1080/00207541003801291.

[29.] Pidd, M. (2004). Simulation worldviews – So what? In R.G. Ingalls, M.D. Rossetti, J.S. Smith, & B.A. Peters (Eds.), *Proceedings of the 2004 Winter Simulation Conference* (pp. 288–292). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[30.] Power, D. J. & Sharda, R. (2007). Model-driven decision support systems: concepts and research directions. *Decision Support Systems*, *43*, 1044–1061. DOI: 10.1016/j.dss.2005.05.030.

[31.] Reiser, M. & Lavenberg, S.S. (1980). Mean value analysis of closed multichain queueing networks. *ACM Journal*, *27*(*2*), 313–322. DOI: 10.1145/322186.322195.

[32.] Salido, M. A., Rodriguez-Molins, M., & Barber, F. (2012). A decision support system for managing combinatorial problems in container terminals. *Knowledge-Based Systems*, *29*, 63–74. DOI: 10.1016/j.knosys.2011.06.021.

[33.] Sargent, R.G. (2010). Verification and validation of simulation models. In B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, & E. Yücesan (Eds.), *Proceedings of the 2010 Winter Simulation Conference*, (pg. 166–183). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[34.] Schlesinger, S. (1979). Terminology for model credibility. *Simulation*, 32(3), 103-104. DOI: 10.1177/003754977903200304.

[35.] Shen, W.S. & Khoong, C.M. (1995). A DSS for empty container distribution planning. *Decision Support Systems*, *15*, 75–82. DOI: 10.1016/0167-9236(94)00037-S.

[36.] Sprague, R. H. (1980). A framework for the development of decision support systems. *MIS Quarterly*, 4(4), 1-26.

[37.] Ursavas, E. (2014). A decision support system for quayside operations in a container terminal. *Decision Support Systems*, *59*, 312–324. DOI: 10.1016/j.dss.2014.01.003.

[38.] Wiese, J., Suhl, L., & Kliewer, N. (2011). Planning container terminal layouts considering equipment types and storage block design. In Böse, J.W. (Ed.), *Handbook of Terminal Planning*, 219–245, New York: Springer.

BIOGRAPHICAL NOTE

PASQUALE LEGATO is an Associate Professor of Operations Research in the Department of Informatics, Modeling, Electronics and System Engineering (DIMES) at the University of Calabria, Rende (CS, Italy). He has been a member of the Executive Board of the University of Calabria as well as university delegate for the supervision of associations and spin-offs from the University of Calabria. He has been involved in several EEC funded research projects aimed at the technological transfer of SO procedures and frameworks in logistics. He is a member of the INFORMS Simulation Society. His research activities focus on predictive stochastic models for cyber security, queuing network models, stochastic simulation and the integration of simulation techniques with combinatorial optimization algorithms. His e-mail address is: legato@dimes.unical.it and his web-page can be found at http://wwwinfo.dimes.unical.it/legato.

RINA MARY MAZZA is the Research Manager of the Department of Informatics, Modeling, Electronics and System Engineering (DIMES) at the University of Calabria, Rende (CS, Italy). She graduated in Management Engineering and received a PhD in Operations Research from the above university. She has a seven-year working experience on knowledge management and quality assurance in research centers. She has also been a consultant for operations modeling and simulation in container terminals. Her current research interests include discrete-event simulation and optimum-seeking by simulation in complex systems. Her e-mail address is: rmazza@dimes.unical.it.

A decision support system for integrated container handling in a transshipment hub

Highlights

- Model-driven decision support system for integrated container handling
- Queuing network model for resource blocking, locking and vehicle interactions
- Solution by highly-detailed discrete-event simulation
- Quantitative results to support real-life operations management decisions