PLAN VALIDATION FOR CONTAINER TERMINALS

Csaba A. Boer
Yvo A. Saanen

TBA BV
Karrepad 2a
Delft, 2623AP, THE NETHERLANDS

ABSTRACT

Terminal operating systems (TOS) play a major role in today's terminal performance. A TOS is used to create operational plans in order to ensure timely handling of vessels, trucks and trains at minimal operational cost. Creating appropriate operational plans forms a challenge for the planners, as plans should take into account a variety of aspects, such as grounding decisions and equipment dispatching. In addition, several decisions have to be made within a limited time frame, couple of hours before starting the operation. Wrong decisions in the plan can have major impact on the operation and implies financial and safety consequences. Planners face these problems only when the plans are actually executed in live operation. In this article, we propose to use a fast yet accurate simulation of both the operation and the TOS in order to support planners to investigate and adapt plans before they are applied in live operations.

1   INTRODUCTION

In our long run experience of working in container logistics, terminal managers have been complaining to us about the performance of the terminal caused by inappropriate quality of plans (vessel plans, yard plans). Shift plans describe planning activities within a given time frame at container terminals. Typical planning activities are: loading and discharging a vessel, dispatching and utilization of the transportation equipments, utilization of the berth, and container allocation strategies in yard (Agerschou et al. 2004, Stahlbock and Voss 2008, Van Ham and Rijsenbrij 2012). Shift plans are usually prepared a couple of hours before the operation begins. In order to achieve a high productivity and meet contractual berthing windows at the lowest costs, it is crucial to find the optimal amount of equipment deployed. Not only the amount deployed, but also where they are deployed, and how the pick and drop containers in the yard is key to an efficient operation. In order to create an appropriate shift plan the planner has to properly investigate all these aspects and make a good decision in a limited time frame.

In order to improve the quality of the shift plan, we here suggest an approach to answer the following research question:

How can we support the planners' decision making to provide a quality shift plan within a limited time frame?

This is an important question raised by several terminal managers who want to improve the performance of their terminal.

In this paper we propose to use simulation to help planners in their decision making. Simulation has been applied successfully for years in different domains (Law and Kelton 2000). Our approach involves two simulation models, a simulation model of the container terminal (including all entities such as layout,

equipment, drivers, stacks, etc.) and a simulation model of the terminal operating system that controls the container operation. Simulation has the advantage that it can run faster than real time, so the planner can benefit from using simulation as a decision making system by running and analyzing various different scenarios (experiments) at the same time. Scenarios containing different planning decisions (e.g., different amount of equipment dispatched, alternative stowage sequence, alternative grounding locations) result in different outputs helping the planners to make appropriate decisions. Although the idea of using simulation for this purpose is not new (Gambardella et al. 1998, Ottjes et al. 2006, Zeng and Yang 2009, Sun et al. 2012), there are a couple of innovative aspects of our approach. First, we start with *actual data*, that originates directly from the TOS. Secondly, we compare the simulated TOS with the real TOS, both working against the *same* simulation model. Furthermore, the model originates from running in 'emulation mode' (see figure 1), where it is coupled to the real TOS for fine tuning purposes (Boer and Saanen 2008, Boer and Saanen 2012a). As the real TOS does not allow for fast evaluation – in previous papers we reported a maximum speed of 3x real-time – we propose another approach to evaluate a plan in a feasible (operational) timeframe.
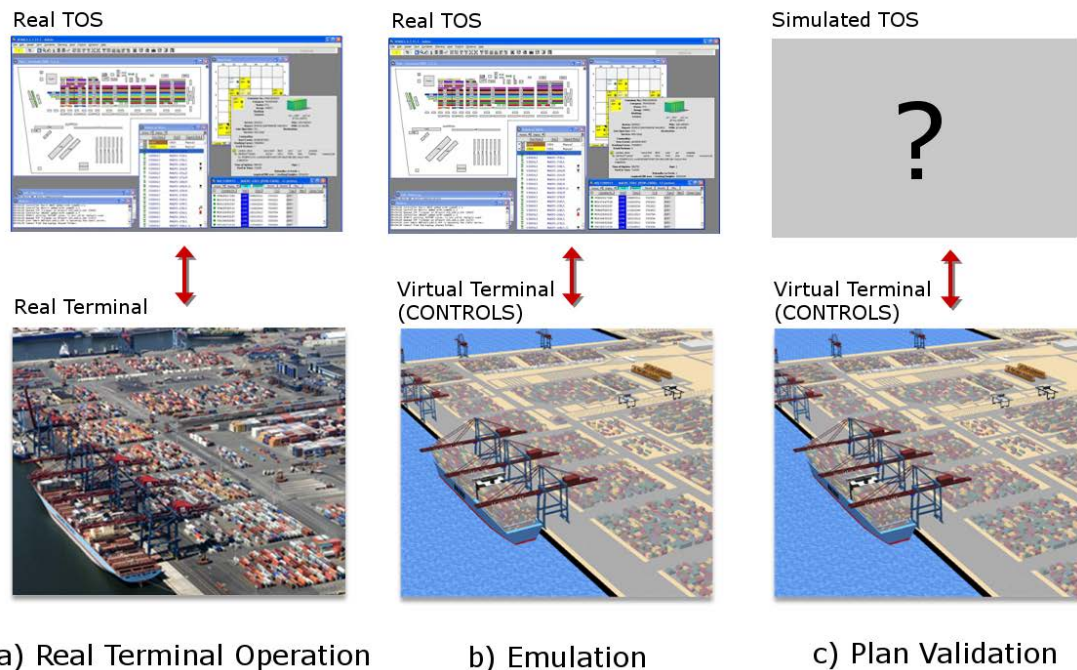

Figure 1: Real terminal operation vs. emulation vs. plan validation (full simulation)

The idea sounds simple and promising but it implies high complexity in designing and developing an accurate simulation of the terminal operating system, that is able to handle a live dataset, and within a very restricted timeframe. One of the most important challenges of the simulation models is that their implementation should be as realistic as possible, and they should be properly verified and validated, otherwise they cannot predict realistic output to the planners.

The paper is structured as follows. First, we present an existing approach to create a simulation model of a container terminal. After that, we discuss the proposed approach to create a simulation of a terminal operating system, as well as the way to come from a plan in the real system to the plan in the simulated TOS. Subsequently, we compare the new approach to an emulation model (using the same simulation model of container terminal but real terminal operating system). Finally, comparing the real TOS with the simulated one we highlight the added value of this approach and answer to the above research question.

2   SIMULATION OF CONTAINER TERMINALS

At TBA we developed a product, called CONTROLS (CONtainer TeRminal Optimised Logistics Simulation) that supports the creation of simulation models of container terminals (Boer and Saanen 2008). The simulation models created by using CONTROLS are full representations of all relevant elements and processes at the terminal, for example the lay-out (yard, rail terminal and quay cranes), models of the equipment (kinematics, driver behavior, routing, disturbances and availability) and performance measurement functionalities. We refer to a simulation model of a container terminal as a virtual container terminal. A virtual container terminal together with a real terminal operating system (such as SPARCS from NAVIS) allows for off-line experimentations and has been applied in a large amount of emulation projects (Boer and Saanen 2012a). All interactions that take place between the equipment at the terminal (including the gate and rail terminal) are defined and supported by the interface between the terminal operating system and the virtual container terminal. This approach, as discussed in (Boer and Saanen 2012a) and (Boer and Saanen 2012b) is used for testing and tuning terminal operating systems and training terminal planners.

Figure 1 depicts three scenarios. Scenario a) presents the real terminal operation where a real terminal interacts with a real terminal operating system. In scenario b) an emulation is represented where a real terminal operating system interacts with a virtual container terminal, CONTROLS in this case (Boer and Saanen 2008). Scenario c) presents a situation where both the container terminal and the TOS are represented by their simulated model.

One might ask: *why we are not considering emulation as an answer to our research question?* A good reason for that is the *execution performance*. The simulation of the virtual terminal (CONTROLS) is capable of running up to 30 times faster than real-time depending on the size and complexity of the terminal. Although we can achieve a relatively high speed, it cannot be always used faster than real time because of the concrete TOS with which it interacts. In order to run faster than real time there is a need for time synchronization between CONTROLS and TOS. Although there are different mechanisms for time synchronization (Boer 2005), not all TOS systems provide this functionality. We achieved time synchronization with SPARCS (Boer and Saanen 2008). However, running faster than two to three times real time causes unexpected behavior of the TOS due to the heavy calculation needed for planning and scheduling, as well as fixed time loops in the code. As formulated in the research question, one of the requirements is to provide an answer for decision making within a limited time frame. This requirement implies that the TOS system should also be capable of running faster than two or three times real time. Based on experience, we found that with an actual TOS, this is not possible yet to date. Therefore, this can be possible if the TOS is also simulated.

In a full simulation setting (see figure 1, scenario c), both the container terminal and the TOS are simulated. This setting aims *plan validation.* The virtual container terminals created by CONTROLS have been used, verified and validated for more than 30 emulation projects. We can use therefore CONTROLS with confidence for the creation of the simulated container terminals that we aim to use for plan validation. The challenge that remained is the simulation of the TOS, in a valid way, and still able to run together with the virtual terminals much faster than real time.

3   SIMULATION OF TERMINAL OPERATING SYSTEMS

Terminal operating systems are by origin administrative systems meant to make sure that all information about containers is securely kept and readily available. Information is used to locate a container quickly, to assess whether a container needs customs inspection, and to determine whether it needs special treatment; for instance, because it contains dangerous cargo. TOS have emerged step by step from an administration system to real-time control systems, with which nowadays terminals cannot work efficiently. A TOS can be seen as the heart of a terminal, mission critical for operations. They support all core processes in a terminal including vessel planning, yard planning, equipment control, gate

management. They also support the information flow around a container visit to the terminal, including the information necessary for invoicing. There is a limited group of off-the-shelf TOS vendors world-wide, of which the four largest are NAVIS, Total Soft Bank (TSB), Real-time Business Solutions (RBS) and Tideworks, spanning jointly approximately 70–80% of the market of off-the-shelf solutions. Besides these, in-house developments still span at least 40–50% of the entire market (Saanen 2010).

On a very high level a terminal operating system has three ingredients: the data, the business logic and the communication. The *data module* contains the data repositories (e.g., databases, setting and configuration files) to store all data used for planning and scheduling. The *business logic module* contains the implementation of algorithms used for planning and scheduling. The *communication module* comprises the implementation of communication protocols towards real equipment, as well as to third party systems. In order to create a simulation model of a terminal operating system we have to consider these three ingredients.

3.1 Dataset – the Snapshot of an Operation

Prior to executing a shift operation, certain information is available, such as the expected time of arrival and departure of vessels and rails, and the list of containers that should be loaded or discharged to the vessels and rails. Using this information, the planner(s) plans all containers using the TOS either manually or by using business logic modules, like a vessel planner module (e.g., Autostow in SPARCS) provided by TOS. This planning leads to a modification of the data module, which we address as the *dataset*. The dataset contains data on the work queues of quay crane and stack cranes, the shuffle moves, dispatched horizontal equipments, all actual container locations, etc.

One can imagine a dataset as a snapshot of a data module of the TOS at any given moment. Typically the dataset contains both steady information such as information about yard locations or vessel structures, and variable information such as information about crane work queues or container locations. During the operation, the planners and the business logic modules in the TOS are adjusting the variable information. For instance, by dispatching a truck to pick up a container for a stack crane leads to changing the location of the container and the availability of a stack slot.

Both during emulation and plan validation we use the *dataset* as an input for creating the models. In the plan validation approach, the dataset is also used to feed the algorithms of the simulated business logic modules.

3.2 The Business Logic Module

The case study presented in section 4 considers an RTG (Rubber Tired Gantry) terminal with approximately 1.6 million TEU (twenty feet equivalent) throughput capacity. In this terminal, there are five types of equipments used: quay crane, terminal truck, RTG, reach stacker and road truck.

The quay crane follows a work sequence containing loading or discharging (unloading) orders. These working sequences are prior to execution prepared (during shift preparation) by a planner. Both in real operation and in plan validation it is important to follow this sequence. In certain situations, however, it happens that a truck arrives earlier with a loading container (out of sequence). There are regulations regarding container attributes (e.g., positions, destination, weight, etc.) that are loaded and unloaded. If the out of sequence container satisfies the regulations it can be loaded, otherwise has to wait. These regulations should be part of the simulated TOS as well. In the next subsections, we present three main components of the business logic module we implemented in plan validation.

3.2.1 Planning Grounding Location

Proper yard planning strategies help to assign the containers to an optimal slot in the yard. This can result in decreased re-handle moves and yard shifts, and an increased yard utilization and productivity. In theory there exist various yard planning strategies, such as pre-marshalling (Chen 1999, Lee and Hsu 2007), sort and store (Kim and Kim 1999a, Kim and Park 2003) controlled random strategy (Dekker et al. 2006), and

more. The algorithms TOS uses rely on container information and stack information. Typical container information refers to: size, type, weight, port of destination (PoD), and category. Information regarding the stack module consists of: type of the stack (export or import, reefer), number of container in the stack module, maximum allowed stacking height and weight and category of each pile.

In the algorithm that we use for plan validation, there is a two step approach defined: first find the appropriate stack module and then find the appropriate slot in the stack module (similar to what is typically used in a TOS). In order to find the appropriate stack module, a weighed scoring mechanism is applied (giving penalties) that considers the characteristics of the stack and the availability of the piles of that specific container. The scoring identifies a number of available stack modules, each having different scores. The best scoring stack module is chosen and considered for the next phase, selecting a slot in the stack module. For selecting a slot again a scoring is applied that considers the type, weight, characteristic of the containers and the height of each available slot that satisfies the container requirements. This results again in a score that finally defines the planned location of the container. During the whole process it is important that the decision made regarding the location of the container store should be determined as late as possible. This mostly happens at the moment when the container is put onto the truck at the quay crane and the hatch clerk (also simulated) enters the container number, or when a road truck arrives at the gate. In order to be able to tune the simulated ground algorithm, it is important to provide high flexibility regarding the configuration of the scoring values (penalties) and the sequence regarding the steps applied to score the stack modules and the slots. In addition, choosing different configurations the simulated grounding algorithm can be used to test different yard planning strategies mentioned above.

3.2.2   Job Dispatching of Horizontal Equipment

During terminal operations, most of the moves take place between the quay cranes and the stack modules, and between the gate and the stack modules. It is very important to use an optimal number of horizontal equipment (terminal trucks, straddle carrier) with a minimal waiting and idle time. Therefore, it is essential to have an appropriate job dispatching mechanism for the horizontal equipments responsible to transport containers. The literature describes several optimization approaches for job dispatching of horizontal equipments (Castilho and Daganzo 1993, Kim and Kim 1999b, Bish et al. 2005) but the largest terminal operating systems (like SPARCS or CATOS) supports two approaches: dedicated job dispatching and pooled job dispatching. In our simulated TOS, we implemented both of them.

In case of the *dedicated job dispatching* approach the horizontal equipment is assigned to a specific point of work (POW), such as a quay crane or a rail crane, and executes the jobs of that POW only. Before dispatching a job it is important to check whether the job is available for dispatching or not. There are several situations that can mark the job unavailable to dispatch for a given period of time, such as occupied or limited number of transfer points at stack module, unplugged reefer containers, or a straddle carrier busy in the same or adjacent stack lane. The dispatcher should neglect unavailable jobs. Several techniques are available to dispatch available jobs, among which: the assign in sequence technique and the score jobs on time and distance approach. If *assign in sequence* is applied the dispatcher assigns to the equipment the first un-dispatched available job in the sequence. In case of *score jobs at time and distance* the dispatcher determines the follow up job from a list of jobs by selecting the job with the lowest starting time. The starting time of a job is calculated based on driving distance to and from job destination and yard handling.

In case of the *pooled job dispatching* approach the horizontal equipments are assigned to multiple POWs. The dispatching algorithm is responsible to decide which job to assign for the given equipments at certain moment in time. This dispatching module also uses a *weighed scoring mechanism* to consider the variety of factors (type of work priority, job urgency, drive distance, congestion at origin or destination of the job, etc.) to find the optimal combination of the available equipments and the un-dispatched available jobs. We implement both of these mechanisms for plan validation and we provide high flexibility regarding the configuration of the rating values (penalties).

Dedicated job dispatching is widely applied at container terminals due to its simplicity, and reigning labor practices. However, when considering performance, often the pooling approach is chosen.

3.2.3   Job Dispatching of Stacking Cranes

The main goal of a stacking crane (e.g., Rubber Tired Gantry, Rail Mounted Gantry) dispatcher is to maintain an equal distribution of the yard jobs among stacking cranes and minimize the gantry moves of stacking cranes. In reality, by using a real TOS, a planner continuously monitors the available jobs and assigns stacking cranes to these jobs. In order to support the job assignment to the cranes and avoid potential collisions, the planners are using so-called CHE (container handling equipment) ranges. A CHE range contains yard jobs assigned to a stack crane and it can cover a part of a stack module (stack bays) or even more stack modules being in one lane. The size of the CHE range is mainly dependent on the number of jobs available in that area. The simulated stacking crane dispatcher imitates the behavior of the planner and thus periodically checks the available yard jobs and it determines accordingly the CHE ranges and dispatches stack cranes to the corresponding areas. The dispatching algorithms that we use for plan validation, are similar to the logic used by the operators in the control room, and rely on yard job information and stack crane information.

The way CHE ranges are created and assigned depends on a number of criteria. In the first place, for each lane there are as many CHE ranges created as stacking cranes. For each CHE range the algorithm calculates the CHE range weight taking into account the jobs weight (importance of the job to be executed) in that CHE range. Based on the results, the size of the CHE ranges are adjusted within a lane such that the amount of jobs in each CHE range matches the average amount of jobs per stack crane. When the CHE ranges are properly balanced, another algorithm checks whether the jobs among the stack lanes are properly balanced (similar amount of jobs per stack crane planned in each stack lane). This algorithm improves the jobs/stack crane at stack lane ratio by moving stack cranes between the lanes. Moving or removing stack cranes from stack lanes involves reconsideration (splitting or merging) of the CHE range. Once the algorithms found the proper number of stack cranes to be assigned per stack lane and the proper number of CHE ranges per lane, the dispatcher assigns each stacking crane to the CHE range in which it is supposed to operate. The simulated stack crane dispatcher provides a high flexibility regarding the configuration of the rating values used in algorithms.

In this section we mainly focused on the dataset and a part of the business logic of the TOS. In fact, a TOS is not just a collection of algorithms but also a communication layer towards real equipment and a frontend graphical user interface towards the planner. A real TOS is thus not replaceable with the simulated one. Real TOS's are still used to plan the operation and create the datasets before we can use the simulated TOS to run several replications to check the quality of the planning. The planner on its turn uses again the real TOS to make the adjustments that will be the input for real operation. In the next section we present some results we obtained from running plan validation and we compared it to the emulation that uses real TOS.

4   CASE STUDY: SIMULATED RTG CONTAINER TERMINAL

In the remaining of this paper we present the simulation of an RTG container terminal, a case study that we carried out at TBA. First, we describe the container terminal model that is used for emulation and plan validation. Then we present and compare the results obtained by multiple replications of emulation and the results of the plan validation runs.

4.1   The RTG Terminal in a Nutshell

For this case study, we selected an RTG terminal model that we used for a large number of projects for training vessel- and yard planners. The terminal has a static capacity of 45,597 TEU (approximately 1.6M TEU throughput capacity), and there are 10 quay cranes, 36 RTGs, 85 terminal trucks and 11 reach

stackers available. The quay cranes are loading and discharging the vessels planned in the dataset. The containers are stacked in the yard by the RTGs. The containers are carried between the quay cranes and RTGs using terminal trucks. Terminal trucks are carrying containers also between RTGs, these are the so-called housekeeping moves. Road trucks are visiting the container terminal via a gate with the purpose to deliver or receive one or two containers. The road trucks are driving either to a reach stacker or to an RTG stack to complete their request. Figure 2 depicts a snapshot of the terminal.
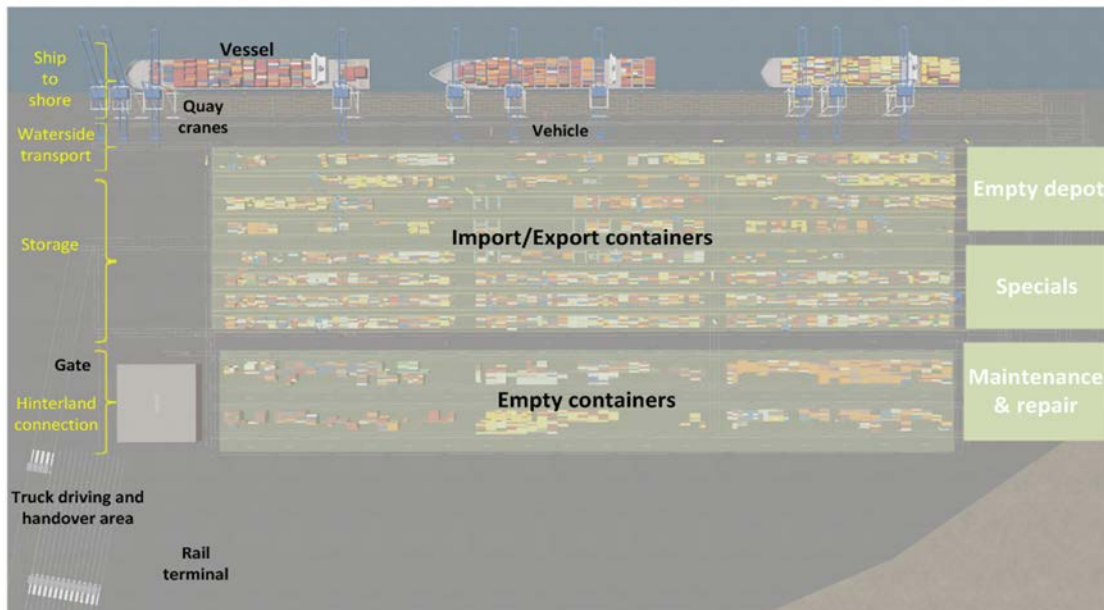
Figure 2: RTG terminal

In the dataset that is used for experimentation, there are 8 quay cranes planned to load and to discharge 3 vessels. The load containers are moved from the yard to the quay cranes and the discharge containers are placed from the quay cranes to the yard. There are 21 RTGs enabled and planned in the dataset which serve in total 40 terminal trucks. The dedicated approach is used to plan the terminal trucks to the quay cranes, meaning that pools consisting of 5 terminal trucks each are assigned during the whole experiment to a single quay crane. In the first 8 hours of experiment, there are 185 road trucks expected to visit the terminal. They are served by the RTGs and by 6 planned reach stackers (handling the empty containers).

The settings used in plan validation, including the strategy of the RTG dispatcher and the applied dedicated truck dispatching, are in line with the settings used in the real terminal operating system. Since most of the activities are planned and carried out during the first 8 hours, we focus on this time period during the experiment.

4.2   Comparing the Real and the Simulated Terminal Operating Systems

In order to compare the real and the simulated terminal operating systems we carried out 8 replications in emulation mode and a similar number of replications in plan validation mode. Both emulation and plan validation interact with the same simulated terminal (CONTROLS).
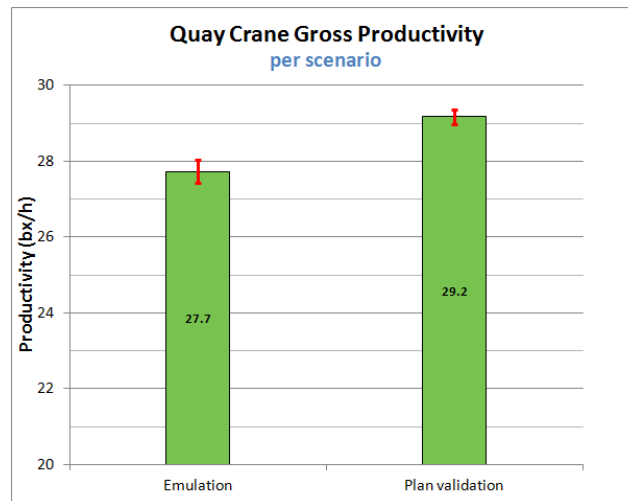
Figure 3: Performance comparison emulation mode versus plan validation mode

The performance of quay cranes is considered as one of the most important performance indicators of a terminal. Therefore, for performance comparison benchmark we consider the average quay crane gross productivity during the 8 hour experiment. The above figures show that the average quay crane productivity is lower in case of emulation than in case of plan validation (both having 95% confidence interval). The figures indicate that the simulated TOS performed better than the real TOS. In order to find out the reasons behind differences, in the rest of this section we analyze closely the average key performance indicators of these runs for different equipments controlled by the real and by the simulated terminal operating system.

4.2.1   Key Performance Indicators

Both during emulation and plan validation all containers are discharged from the vessel. The plan validation has performed better with respect to the loading operation. In average, it loaded with 5% more containers during the first 8 hours of the experiment than the emulation approach. It is interesting to further investigate why does the loading operation perform better when using plan validation.

Regarding the status distribution the two approaches provide more or less the same results (see figure 4 – Quay crane status distribution). The figure shows the average status of the quay crane, but the user has the possibility to request individual quay crane statuses as well. For the shift planner it can be extremely useful to see the large idle or waiting times and find out the reason behind it (clashing quay cranes, less terminal trucks dispatched, etc.)

In case of RTGs we selected a chart that shows the cumulative distance in meter driven by RTG (see figure 4 – RTG driven distance). The number of extra quay crane load moves present in plan validation discussed before (5%) is reflected in RTG box count chart as well. Furthermore, it can be also observed that there are more shuffle moves during plan validation than during emulation. This difference can be explained with the extra load moves. In case of loading a specific container there is a real chance that there are some other containers that are placed on top of the loaded container. In this case the containers from the top need to be shuffled as well. We can also observe that during plan validation the RTGs have moved more often than during emulation. This is another important aspect that needs to be further investigated.
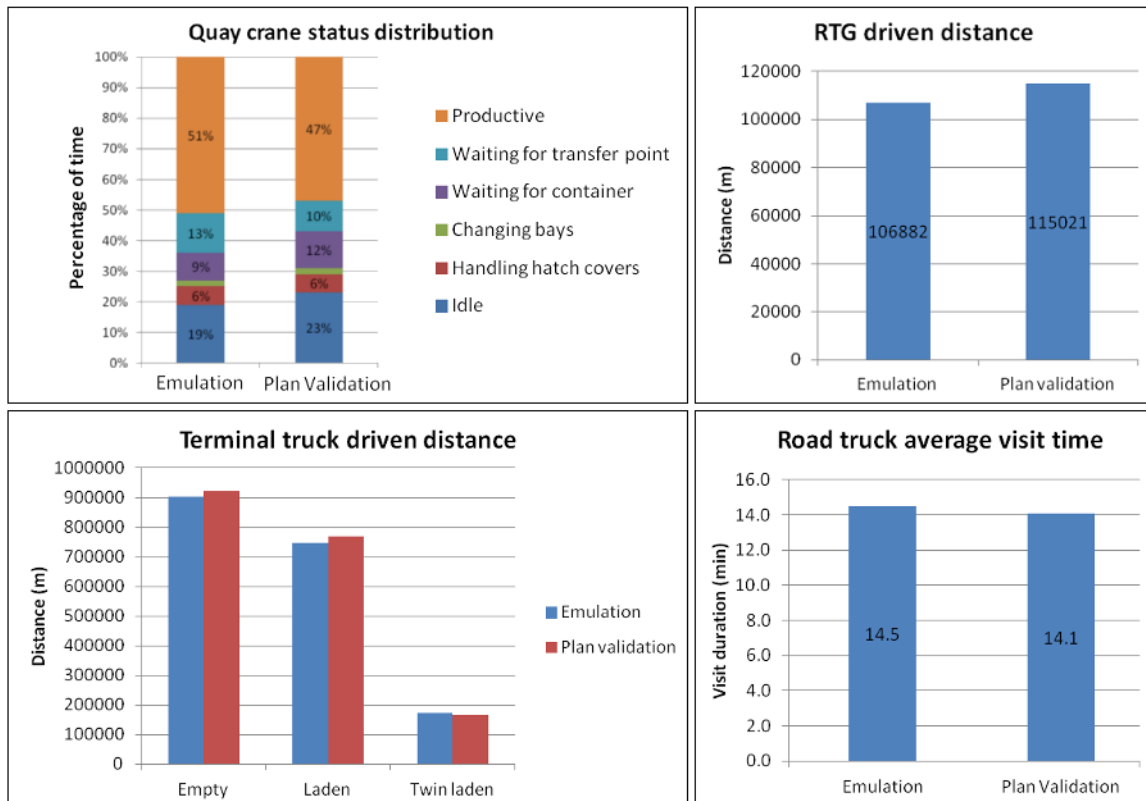
Figure 4: Quay crane, RTG, terminal truck and road truck performance comparison

In order to calculate the driven distance for terminal trucks we consider three statuses: driving empty, driving with one container (laden), or driving with two containers (twin laden). Figure 4 (Terminal truck driven distance) indicates that the driving distances are very similar even for different statuses.

The road trucks visit the terminal to pickup or deliver one or two containers. The average visit time that a road truck spent at the terminal (see figure 4 – Road truck average visit time) are close to each other during both cases, indicating that the plan validation gives a fair estimation of the land side operation.

4.2.2 Validation conclusion

In this section we highlighted some statistics regarding the behavior of the equipments present in operation. In general, the identified differences are small (±5%), and we consider the results promising to fine tune the simulated TOS in order to get it to the same level as the real TOS.

4.2.3 Next steps: results of plan validation

The subsequent step is to find out whether the findings from the plan validation runs can lead to an improved plan. So what can we learn from plan validation, and which results can we apply in the upcoming plans?

We aim to facilitate the learning by two means: by detailed statistics and visualization. The statistics enable a planner to find performance hick-ups and define solutions to overcome these. Detailed visualization of the operation includes all the logical information about the equipment and container flow (see Figure 5).
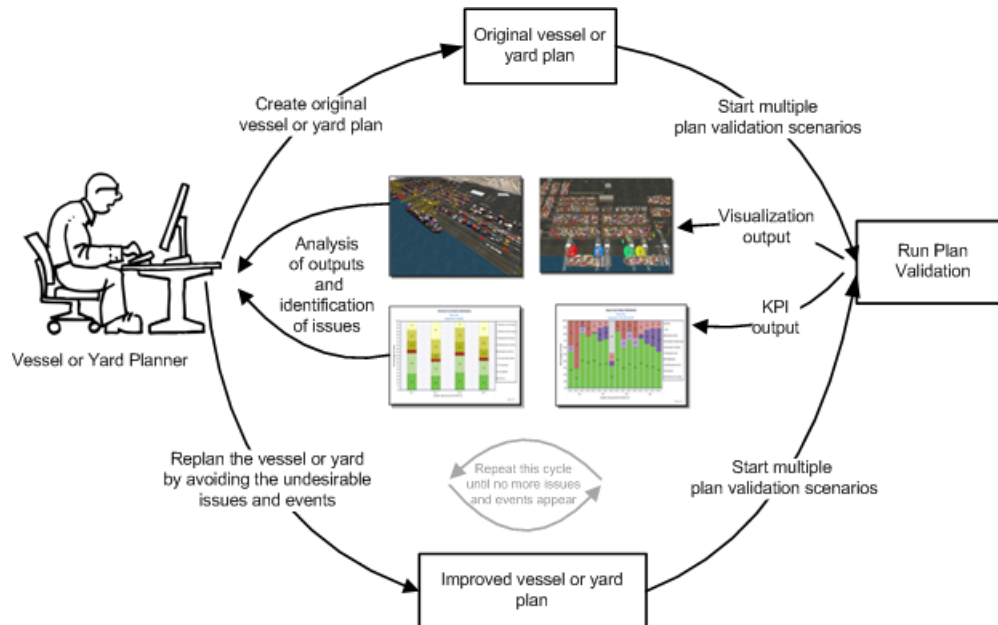
Figure 5: Proposed learning cycle to continuously improve the plan using plan validation

Currently we are involved in an ongoing research that considers a learning cycle to continuously improve a vessel and yard plan using plan validation. Although the research is not yet finalized we can already report some preliminary results. As Figure 6 illustrates, the consecutive action taken by the vessel planners lead to an increase in the performance. Moreover, the planners were able to carry through the improvements within the limited time before the plan had to be executed.
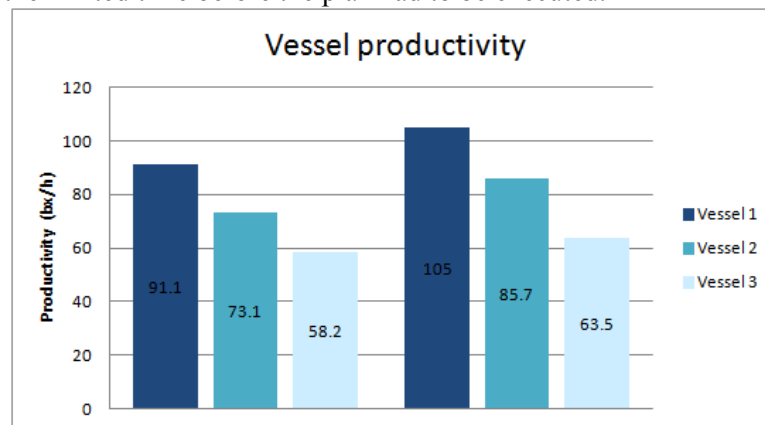

Figure 6: First results from applying plan validation in a learning cycle (Magnúsdóttir 2014).

5   CONCLUSION

The answer to the research question raised in the introduction "*How can we support the planners' decision making to provide a quality shift plan within a limited time frame?*" is by means of plan validation. The plan validation consists of a simulated container terminal (CONTROLS) and a simulated terminal operating system that has the ability to take a real shift plan as an input and run multiple replications as fast as possible. The results of these replications presented in a form of KPI reports support the planners in their decision making. We also found – yet indicatively – that turning these findings into plan improvements lead to an increase in the objective KPI's.

Answering this question and experimenting more and more with plan validation raises new questions and challenges that we have to answer if we plan to apply it in practice. First of all, the setup process to

import a shift plan from a real TOS, taking into account different equipment and terminal configurations, starting a large number of replications with the different configurations should be as easy and as fast as possible. We need a certain level of automation and a framework that hides the unnecessary technical details from planner. In other words after finishing a shift plan prepared in TOS the planner should "press a button" and wait for the results of plan validation.

The second challenge is to support the planner to interpret the obtained result. As time is limited and the planner should act instantly the presented results should be simple but still pointing to complex real planning issues (long waiting queues, expected clashes, busy areas, unproductive equipments, etc.). Using a large amount of KPIs needs time to interpret and conclude the planning issue, and as time is crucial in decision making we believe that we need to find another approach for data interpretation. Currently we are working on both of these challenges.

ACKNOWLEDGMENTS

REFERENCES

Agerschou, H., I. Dand, T. Sorensen, T. Ernst. 2004. *Planning and design of ports and maritime terminals*. 2nd ed. London: Thomas Telford Ltd.

Bish E.K., F.Y. Chen, Y.T. Leong, B.L. Nelson, J.W.C. Ng, D. Simchi-Levi. 2005. "Dispatching vehicles in a mega container terminal." *OR Spectrum* 27(4): 491–506.

Boer, C. A. 2005. *Distributed Simulation in Industry*. ERIM Ph.D. Series Research in Management: Rotterdam, The Netherlands. Available via http://hdl.handle.net/1765/6925 [accessed March 25 2014]

Boer C.A. and Y. Saanen. 2008. "CONTROLS: Emulation to Improve the Performance of Container Terminals." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Monch, O. Rose, T. Jefferson, and J. W. Fowler, 1094-1102, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Boer C.A. and Y.A. Saanen. 2012a. "Improving container terminal efficiency through emulation.", *Journal of Simulation* 6(4): 267-278.

Boer C.A. and Y.A. Saanen. 2012b. "Testing, tuning and training terminal operating systems. A modern approach." In *International Conference on Logistics and Maritime Systems (LOGMS)*, edited by H.O Gunther, K.H. Kim and H. Kopfer, 25-35, Bremen, Germany.

Castilho B.D. and C.F. Daganzo. 1993. "Handling strategies for import containers at marine terminals." *Transportation Research* 27B(2): 151–166

Chen T. 1999. "Yard operations in the container terminal—A study in the 'unproductive moves." *Maritime Policy and Management* 26(1): 27–38.

Dekker R, P. Voogd and E. van Asperen. 2006. "Advanced methods for container stacking." *OR Spectrum* 28(4): 563–586.

Gambardella L.M., A.E. Rizzoli and M. Zaffalon. 1998. "Simulation and planning of an intermodal container terminal." *Simulation* 71: 107–116.

Kim K.H. and H.B. Kim. 1999a. "Segregating space allocation models for container inventories in port container terminals." *International Journal of Production Economics* 59(1-3): 415–423.

Kim K.Y. and K.H. Kim. 1999b "A routing algorithm for a single straddle carrier to load export containers onto a containership." *International Journal of Production Economics* 59(1-3): 425–433.

Kim K.H. and K.T. Park. 2003. "A note on a dynamic space allocation method for outbound containers." *European Journal of Operations Research* 148(1): 92–101.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Lee Y. and N. Hsu. 2007. "An optimization model for the container pre-marshalling problem." *Computers & Operations Research* 34(11): 3295–3313.

Magnúsdóttir J.G. 2014. "Exploring container terminal planning: Effects of vessel plan forecasting and event-based visualization on planning and situation awareness." Master Thesis, Delft, The Netherlands, 2014.

Ottjes J, H. Veeke, M. Duinkerken, J. Rijsenbrij and G. Lodewijks. 2006. "Simulation of a multiterminal system for container handling." *OR Spectrum* 28: 447–468.

Saanen Y. 2010. TOS market overview. *Internal report*, TBA B.V., Delft, The Netherlands.

Stahlbock R. and S. Voss. 2008. "Operations research at container terminals—A literature update." *OR Spectrum* 30(1): 1–52.

Sun Z., L. H. Lee, E. P. Chew and K. C. Tan. 2012. "MicroPort: A general simulation platform for seaport container terminals." *Advanced Engineering Informatics* 26(1): 80-89.

Van Ham, J.C. and J.C. Rijsenbrij. 2012. *Development of Containerization*. IOS Press, Amsterdam.

Zeng Q. and Z. Yang. 2009. "Integrating simulation and optimization to schedule loading operations in container terminals." *Computers and Operations Research* 36: 1935–1944.

AUTHOR BIOGRAPHIES

CSABA A. BOER is a senior product manager and head of emulation department within TBA BV, one of the leading logistics and simulation consultancy firms in Europe. He is responsible for several products within the organization, one of which is CONTROLS and plan validation. He holds a Ph.D. in Computer Science and Logistics from Erasmus University Rotterdam. His research interests include distributed simulation, distributed virtual environments, port logistics, and port simulation and emulation. His e-mail address is csaba.boer@tba.nl.

YVO A. SAANEN is managing director and founder (1996) of TBA B.V., a leading simulation consultancy company in The Netherlands. He heads the department that supports ports and terminal operators all over the world in their design process of container terminals by means of simulation. He holds an M.Sc. in Systems Engineering and a Ph.D. on the design and simulation of robotized container terminals, both from Delft University of Technology. In addition, Yvo Saanen is a lecturer at Erasmus University in Maritime Economics and Logistics. His e-mail address is yvo.saanen@tba.nl.