Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

A SIMULATION MODEL FOR THE PLANNING AND CONTROL OF AGVS AT AUTOMATED CONTAINER TERMINALS

Berry Gerrits
Martijn Mes
Peter Schuur

Department of Industrial Engineering and Business Information Systems
University of Twente
P.O. Box 217
7500 AE Enschede, THE NETHERLANDS

ABSTRACT

This paper presents a scalable and flexible agent-based simulation (ABS) model for the planning and control of AGVs at automated container terminals (ACTs). A comprehensive and generic multi-agent system is presented to effectively manage the container handling process between the quay and stacks of ACTs. Specifically, we design an agent-based Traffic Manager, which serves as a control application layer for all processes related to the horizontal transport of containers using AGVs. Our ABS model provides a playground to easily test out the effectiveness of different types and layouts of ACTs.

1 INTRODUCTION

The last couple of decades have witnessed a trend towards increasing container vessel sizes. Since mega-container ships, with a capacity of up to 18,000 TEU, increasingly replace smaller vessels, terminals worldwide are struggling to turn ships around in a timely manner. The Vessel Turnaround Time (VTT), the time it takes between the arrival of a vessel and its departure from port, is of critical importance. To improve operational productivity, container terminals worldwide have increasingly adopted automated systems, such as automated guided vehicles (AGVs) and automated stacking cranes (ASCs) (Montoya-Torres et al. 2015). However, despite the increase in vessel size, the overall terminal productivity has not improved. Ocean carriers are asking for shorter VTTs due to increasing delay costs. Soren Skou, CEO of Maersk stated that “the industry is stuck at 25 to 30 moves per Quay Crane (QC), per hour”, while the technical productivity of a QC is between 45 and 55 moves per hour (Tirschwell 2015). Also for terminal operators, port productivity is important. Singapore-based Container Transport International calculated that if the Gross Moves Per Hour (GMPH) goes up from the current 28 to 32, this would result in net savings of $800,000 per QC per year (Gangwani 2015). As the VVT is highly determined by the productivity of the QCs, measured in GMPH, keeping the QCs utilized is of vital importance for port productivity. The current poor productivity of QCs is mainly caused by the unavailability and failures of horizontal transport equipment (e.g., AGVs). Studies show that this factor determines around 40% of the total VTT delay, corresponding to a loss in productivity of approximately 12 moves per crane and per hour (Gayathma et al. 2015).

The availability of horizontal transport in an ACT can be increased using a combination of (i) a more robust AGV navigation system to decrease AGV failure rates and (ii) a smart Traffic Manager to effectively plan and control the fleet of AGVs in a stochastic environment. This paper focuses on the latter by designing an agent-based Traffic Manager. In our design, the Traffic Manager is the control application layer for all processes related to the horizontal transport of containers using AGVs. The goal of this paper is to develop an agent-based simulation (ABS) model for the Traffic Manager. The simulation model should be flexible in the sense that different types and layouts of container terminals can be easily implemented.

The remainder of the paper is structured as follows. Section 2 reviews the literature and states our contribution. We describe the problem setting in Section 3 and present our multi-agent system (MAS) to control the container handling process in Section 4. Section 5 describes our ABS model to analyze the proposed MAS. Our model is verified and validated in Section 6. We close with conclusions in Section 7.

2 LITERATURE

Simulation is broadly used as a tool to analyze container terminal operations (see Dragović et al. 2017 for a literature review). Other recent reviews focus on specific topics, including scheduling and routing (Fazlollahtabar and Saidi-Mehrabad 2015; Kaoud et al. 2017), resource allocation problems (Huang et al. 2014), smart technologies (Cimino et al. 2017) and future challenges (Kim and Lee 2015). Furthermore, agent-based modeling approaches have been applied to container terminals as well, with papers focusing on port management (Wibowo et al. 2015), terminal configurations (Sun et al. 2013), inter-terminal cooperation (Nabais et al. 2013), collision avoidance (Marinica et al. 2012), yard crane scheduling (Fotuhi et al. 2013), dispatching of straddle carriers (Garro et al. 2015), terminal planning (Mes and Douma 2016), and bay planning (Parthibaraj et al. 2017). Other studies focus on the comparison between centralized and decentralized approaches, such as yard crane coordination (Sharif and Huynh 2012) and port regulation modes (Zheng and Negenborn 2014). This study aims for a more holistic approach, by designing a MAS suitable for the operations at the quayside of a container terminal and an ABS model to test its performance. This paper takes an approach similar to Henesey et al. (2009); however, they did not explicitly model the AGV movements, thereby ignoring congestion, which has a major impact on the VTT. The novelty of this research is therefore twofold: (i) the design of a MAS capable of controlling all the operations at the quayside of a container, including elements such as conflict resolution and deadlock avoidance and (ii) the design of a scalable and flexible ABS model to evaluate the MAS for different container terminals.

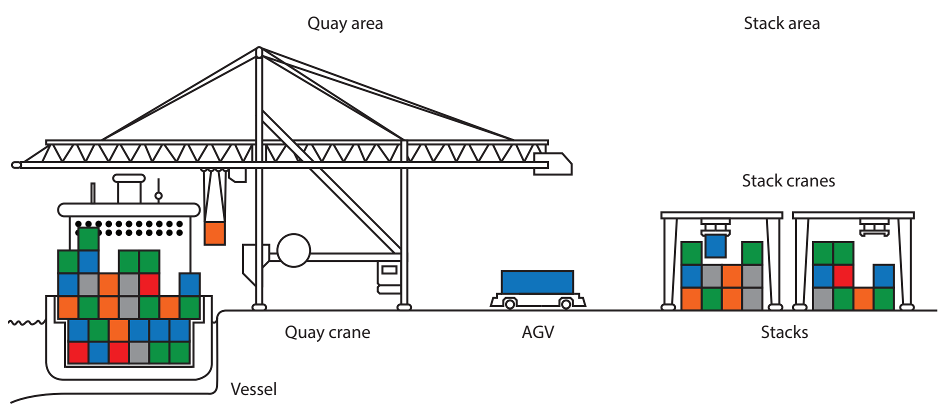

Figure 1: Container terminal operations.

3 PROBLEM DESCRIPTION

Figure 1 provides a cross-section of a typical container terminal. While berthed, the vessel is served by QCs to load and unload containers from the vessel. AGVs are available to transport the containers between the QCs and the stack. The containers are loaded/unloaded on/off the AGVs at the quay using QCs and at the stack using Stack Cranes (SCs). We focus on two types of jobs for the AGVs: (i) receiving a container from a QC and moving it to the stack (import job) and (ii) receiving a container from a SC and moving it to the vessel (export job). We thus exclude empty movements and marshaling as this is typically done while no vessels require service, and thus does not influence VTT.

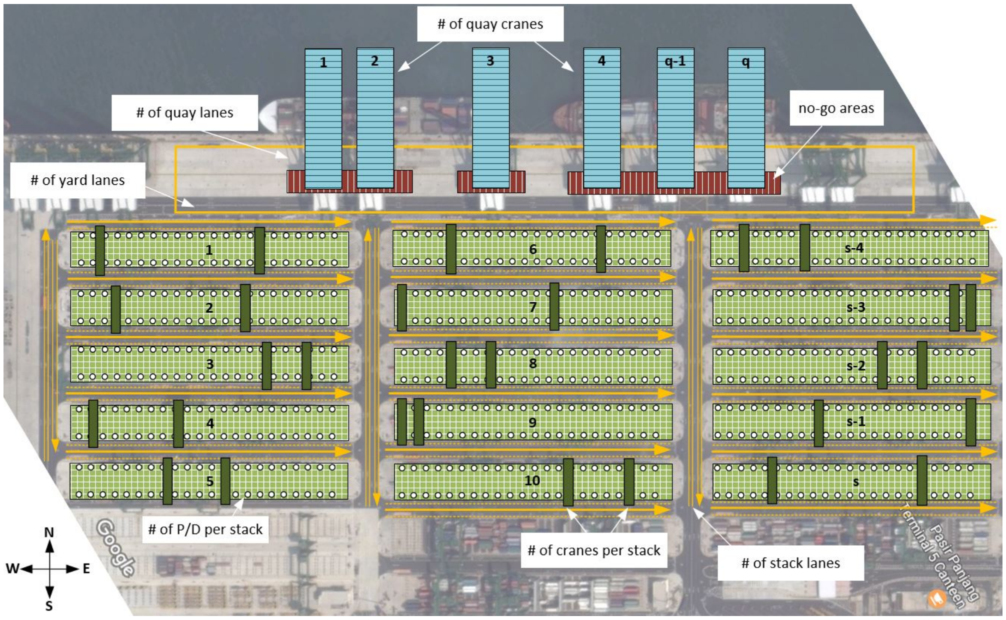

Figure 2: Top-down view of a container terminal in parallel layout.

We focus on container terminals where the stacks are oriented in parallel to the quay (commonly referred to as parallel or Asian layout). Figure 2 provides a top-down view of the quayside of a container terminal. To simplify the discussion, let us relate the notions East (E), West (W), North (N) and South (S) to the orientation of Figure 2. We characterize a container terminal using the following nine elements:

- Vessels. Vessels are berthed at the quay and are characterized by berth location, bays, rows, and tiers (i.e., levels). The stowage plan contains information on where the containers are positioned in the vessel. The call size states how many containers are to be loaded/unloaded.
- Quay cranes. Multiple QCs (typically 5 to 7) serve a single vessel. We focus on QCs that process one container per move from and to the vessel, also referred to as single-cycle. The QCs operate with a cycle time based on the storage or retrieval position at the vessel.
- Quay lanes. Underneath the QC, a fixed number of unidirectional quay lanes are available, where AGVs can drive from W-E, N-S, or S-N to facilitate the (un)loading of containers by QCs.
- No-go areas. These areas at the QCs are forbidden for AGVs, either due to safety restrictions (to avoid collisions with the cranes) or due to temporary space for special cargo.
- Yard lanes. At the N-side of the stackyard, an even number of unidirectional yard lanes are available (alternatingly from W-E and from E-W), where AGVs can drive between the stackyard and the quay.
- Stacks. The stacks are used to temporarily store containers. They are characterized by width, depth, and stacking height. The total number of stacks determines the total stack area.
- Pick-up and drop-off locations per stack. Alongside the stack, lanes are available for AGVs to facilitate loading and unloading. AGVs typically stop at the location where the container is stored in the stack, to minimize crane movements.
- Cranes per stack. One or more cranes are available at the stack to load and unload AGVs and to store and retrieve containers from the stack.
- Stack lanes. Between the stacks there are two types of stack lanes available to connect the inner stack area with the yard lanes: (i) unidirectional stack lanes from W to E and (ii) unidirectional stack lanes from N to S and from S to N. Typically, between the stacks, one lane is available of type (i) and two lanes are available of type (ii).

Using these nine elements as input parameters for the simulation model, we can develop a model which is scalable, applicable to other Asian-based container terminals, and thus reusable.

4 SYSTEM DESCRIPTION

Given a certain layout of a container terminal, we construct three control layers: low-level, mid-level and high-level. The mid-level control we call Traffic Manager. It is used to make operational decisions for horizontal transport (i.e., for the AGVs) to move containers to and from the stack. The high-level control is done at the Terminal Operation System (TOS) layer. This layer makes decisions on berth allocation, quay crane assignment, stowage planning, and workload- and capacity planning. From there on, the mid-level control layer takes over and its goal is to efficiently utilize the given equipment and orders to be processed. At the low-level control layer (i.e., vehicle control) decisions on path generation, navigation, steering and braking are made. The Traffic Manager is modeled using an agent-based approach based on Gerrits et al. (2017), who designed and simulated a MAS for the planning and control of semi-trailers at distribution centers. We adopt this design and incorporate the ACT-specific peculiarities to fit the framework to this case study. These modifications are discussed at the end of this section. First, the overview of the entire system is given in Figure 3, including simplified interactions between the different layers.

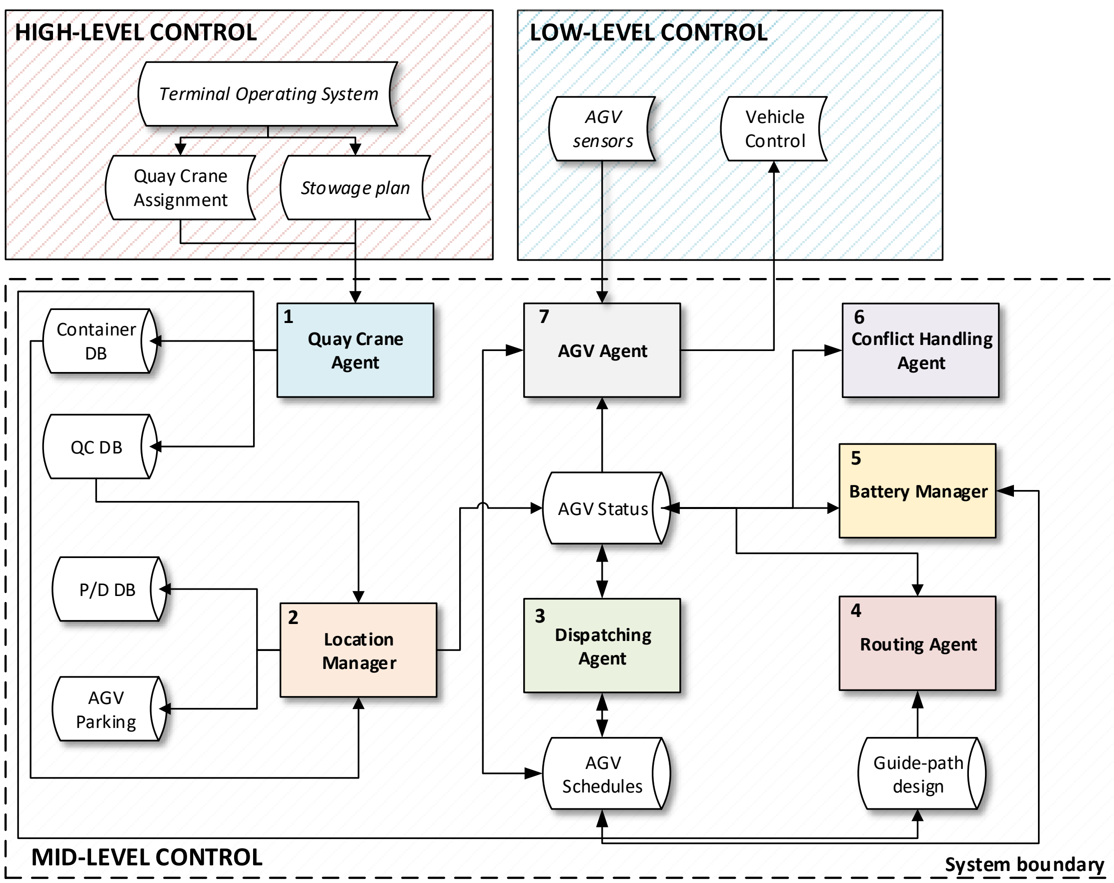

Figure 3: Data coupling diagram of the proposed MAS.

From Figure 3 we see seven agents emerging. These agents are chosen such that they together provide a functional decomposition of the Traffic Manager, whilst respecting the software engineering criteria of cohesion and coupling. In this way, we provide an integral approach with loosely coupled agents, which improves scalability and generality. We distinguish the following seven agents:

- Quay Crane Agent (one per QC). The Quay Crane Agent is responsible for the loading and unloading of containers to and from the vessel. It maintains a loading and unloading schedule based on the stowage plan, and it triggers the dispatching of AGVs based on a fixed look-ahead period via the Location Manager and the Dispatching Agent. Given an ACT, this look-ahead period is chosen such that (i) AGVs are able to respond in a timely manner to a request and (ii) the number of AGVs waiting underneath the crane is balanced out such that the system is not flooded.
- Location Manager (one per system). The Location Manager assigns pick-up, drop-off, idling, and parking locations to all AGVs and containers. More specifically, it assigns import containers to stacks (based on input from the TOS), export containers to QCs, and origin and destination locations to all AGVs such that they can transport the containers to the appropriate locations.
- Dispatching Agent (one per system). This agent assigns transport requests (movement of a container from an origin to a destination given by the Location Manager) to AGVs using an auction protocol. AGV agents place bids to a call proposed by the Dispatching Agent. The Dispatching Agent then evaluates all the bids and selects the AGV with the best bid to perform the transport.
- Routing Agent (one per system). This agent determines the route between origin and destination taking into account current congestion levels. During operations it can dynamically reroute AGVs to avoid conflicts and/or to reduce travel times.
- Battery Manager (one per system). This agent maintains the battery status of the AGVs and is responsible for effective charging schedules.
- Conflict Handling Agent (one per system). This agent resolves all possible conflicts between AGVs and maintains a conflict-free environment by making stop-and-go decisions (i.e., at intersections).
- AGV Agent (one per AGV). The AGV Agent processes all data from the AGV controller, maintains the state of the AGV (e.g., position, speed, and battery status), responds to transport requests from the Dispatching Agent, and can negotiate with other AGV agents to swap jobs.

This design has two distinguishing features compared to the work of Gerrits et al. (2017). First, our case study calls for a multitude of agents connected to the high-level control layer. In an ACT, the jobs (i.e., request to transport a container) are fed to the system via the QCs. As we have multiple QCs in the system, it seems logical to represent each QC by an agent. Moreover, typically multiple vessels are berthed at the same time and each requires service. Therefore, QCs assigned to a vessel might compete for AGV capacity with QCs assigned to other vessels. To capture the essence of this trade-off between agents, we require each QC to be represented by an agent. Second, we require additional agent intelligence given the larger scale of this case study. Specifically, we require (i) additional intelligence to cope with deadlock avoidance and (ii) a mechanism to update the allocation of jobs to AGVs (via the Dispatching Agent) as disruptions in the process might render previously assigned schedules sub-optimal. The latter is taken care of by implementing a rescheduling policy where AGV agents can reconsider the jobs that were allocated to them and negotiate with other AGV agents to try to swap jobs such that the expected utilization of the QCs is maximized. With a keen eye on deadlock avoidance, both the Routing Agent and the Conflict Avoidance Agent have been enhanced. When an AGV agent responds to a call initiated by the Dispatching Agent, it requests a route from the Routing Agent which returns the shortest route between origin and destination. However, when the AGV is ready to execute the job at a later point in time, the initially assigned route might be congested. Therefore, during the execution of the job, the AGV Agent is able to request a new route from the Routing Agent based on the current congestion levels in order to avoid deadlocks and to increase the flow of traffic. Especially the quay area is sensitive to congestion due to AGVs waiting to be served by the QC and the limited number of lanes available. This level of intelligence provides more flexibility than calculating a congestion-free route beforehand as the information on the future positions and routes of all other AGVs is unreliable. Also, the Conflict Handling Agent needs to be more intelligent as there are many crossings, junctions, and traffic-intense areas at an ACT. The area between the stacks is sensitive to deadlocks when the wrong precedence rules are applied. The Conflict Handling Agent uses an increased set of precedence rules, recognizing the peculiarities related to traffic flow of the different areas of an ACT (e.g., quay area and stack area) to minimize the blocking of lanes and avoid deadlocks.

5 SIMULATION MODELING

To evaluate the effectiveness of the MAS presented above for some ACT, we use discrete-event simulation (DES). As stated by Law (2015), DES is suitable to model agent-based systems as in virtually all ABS models state changes occur at a countable number of points in time. We implemented our model in Tecnomatix Plant Simulation from Siemens (see Figure 4).

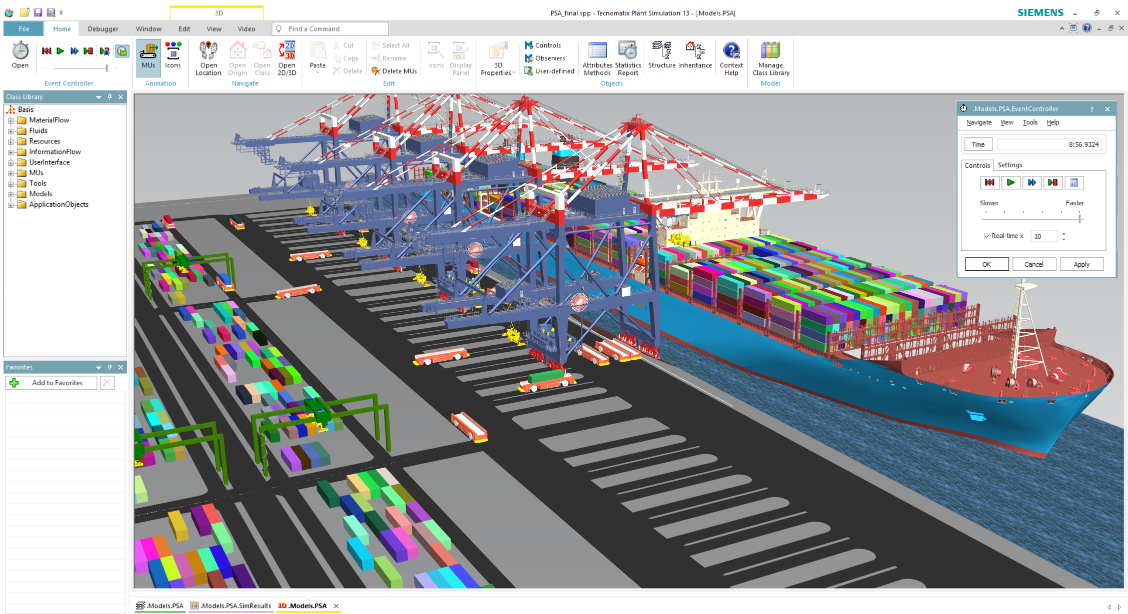

Figure 4: Screenshot of the animated simulation model.

5.1 Layout of an ACT

To build a flexible and reusable model, we first need to incorporate a systemic modeling approach to integrate all the static and moving elements of an ACT as shown in Figure 2. The basis of the static elements of an ACT is the layout of the roads or guide-paths on which the AGVs are able to drive to perform their tasks. These consist of W-E, E-W, N-S, and S-N tracks, characterized by length, width and curvature, and connect the quay-, yard-, and stack areas, as shown in Figure 5. In this figure, we model an ACT with four stacks, 47 quay lanes (three W-E, 22 N-S, 22 S-N) and four yard lanes (two W-E and two E-W). Every lane is made up of multiple small tracks to allow AGVs to make turns. With the parameters shown in Table 1, we can scale and adapt the model to fit any parallel-oriented container terminal using a single method at the initialization of the simulation run.

After the track layout is initialized, all the individual track segments are connected with their successors and predecessors to create a network of tracks. This network is used to form a route between the origin and destination of a transport order. This modeling approach allows for tailoring the model to multiple ACTs, making it flexible and reusable. The other elements of the simulation model are discussed below.

5.2 Stack Area

The stack area serves as a temporary storage area for containers that are waiting for further processing. They are either transshipped or continue their journey via another mode of transport, further inland. The stacks

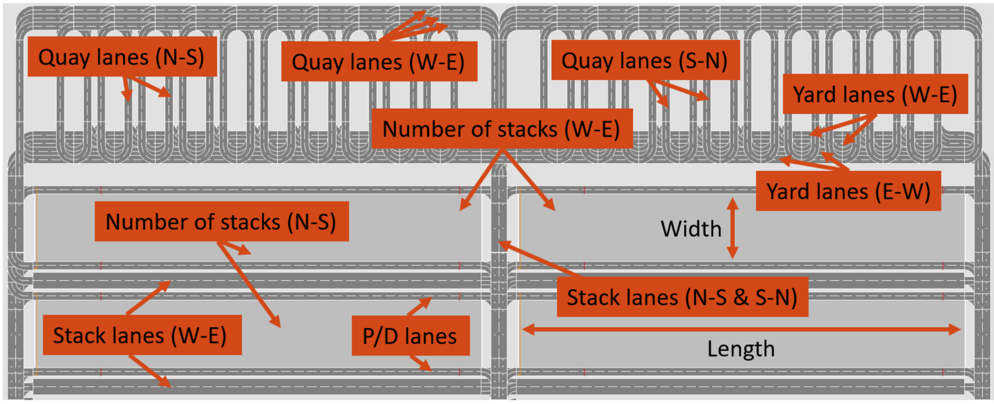

Figure 5: Layout of the roads or guide-paths at an ACT.

Table 1: Simulation parameters to define the ACT layout.

Parameters	Data type	Notes
#Quay lane tracks	Integer	W-E, N-S, S-N
#Yard lane tracks	Integer	Supporting an even number of unidirectional yard lanes (alternatingly from W to E and from E to W)
Dimensions lane track	Real	Length, width and curvature (only for turns)
#Stacks	Integer	E-W and N-S direction
Dimensions stack	Real	Length, width and stacking height
P/D tracks between stacks	Integer	Two P/D lanes at opposite sides of stack lane
#Stack lane tracks	Integer	W-E, N-S, S-N

are characterized by their length, width, stacking height, number of cranes, crane speed, and the stacking policy. The latter determines where containers are stored in the stack and how blocked containers (e.g., located underneath other containers) are retrieved. As in our model the stacks only serve as an origin or destination location for the AGVs, we do not model the stack in detail, but suffice with straightforward values of the characterizations. We use homogeneous dimensions for the stacks, one crane per stack, deterministic crane speeds, and a Nearest-Available stacking policy. This provides enough detail to realistically model the service and waiting times of the AGVs at the stacks. We model the stacks by extending the standard StorageCrane library object of Plant Simulation. It models a three-dimensional stock location where a rail-mounted crane stores and retrieves items. We initialize the stacks by setting their dimensions such that they fit standardized 40-foot containers in all directions, keeping in mind that at both ends of the stack a pick-up and drop-off lane is required for the loading and unloading of AGVs by the stack crane. The standard graphics are adjusted to resemble a realistic 3D stack area.

5.3 Quay Area

The quay area features two important components: (i) a berthed vessel with a call size and stowage plan and (ii) quay cranes to facilitate the loading and unloading of the vessel. A container vessel typically uses the Bay-Row-Tier system, which determines the length, width, and height of the container storage area. We model the vessel using a collection of bays and tiers, for which we use the standard library object Store. That is, a Store is a single-tier, single-row storage area where the length depends on the bay size. For our experiments (see Section 6), we modeled a vessel of the commonly used Triple-E Class. This class of vessels contains 22 bays, where each bay contains up to 20 containers with a maximum stacking height of 19 containers (i.e., 19 tiers). The typical call size of such vessels is 5,000 to 8,000 containers.

Every berthed vessel is served by a fixed set of QCs. We model the QCs using the MultiPortalCrane object of Plant Simulation, which consists of multiple cranes on a runway where every crane has a trolley and a hook, each with independent speed parameters. The runway is used to move QCs when they are required to serve a different bay. The trolley runs along the rails at the top of the crane and moves from and to the quay. The hook is used to model the spreader that hoists and lowers containers. At every simulation run, we are able to adjust the number of QCs serving a vessel by adjusting the parameters of the MultiPortalCrane. Furthermore, the standard graphics were replaced by 3D quay crane graphics.

5.4 AGVs

The AGVs facilitate the horizontal transport of containers between the quay and the stack area. The AGVs are modeled using MUs (Movable Units) and are characterized by their dimensions, speed, and load capacity (i.e., one 40-foot container). At the initialization of the model, a set of AGVs is created at an idling area. This area is also used for buffering AGVs that have no job at hand. In this way, the quay and stack area are not unnecessarily blocked due to idling AGVs. Furthermore, each instance of an AGV has a set of methods to model the agent intelligence as discussed in the next section.

5.5 Agent Intelligence

For brevity we focus on the modeling of the enhanced agent intelligence as discussed in Section 4. This enhanced intelligence includes dynamic scheduling and dynamic rerouting. The latter is used by AGV agents to avoid conflicts and potential deadlocks. As discussed, the area particularly prone to deadlocks is underneath the QCs. We illustrate the rerouting based on this area. As soon as an AGV turns towards the quay and enters an S-N quay track, the rerouting algorithm, as shown in Figure 6 (left side), is initiated where the AGV tries to find an empty route for the last part of its trip.

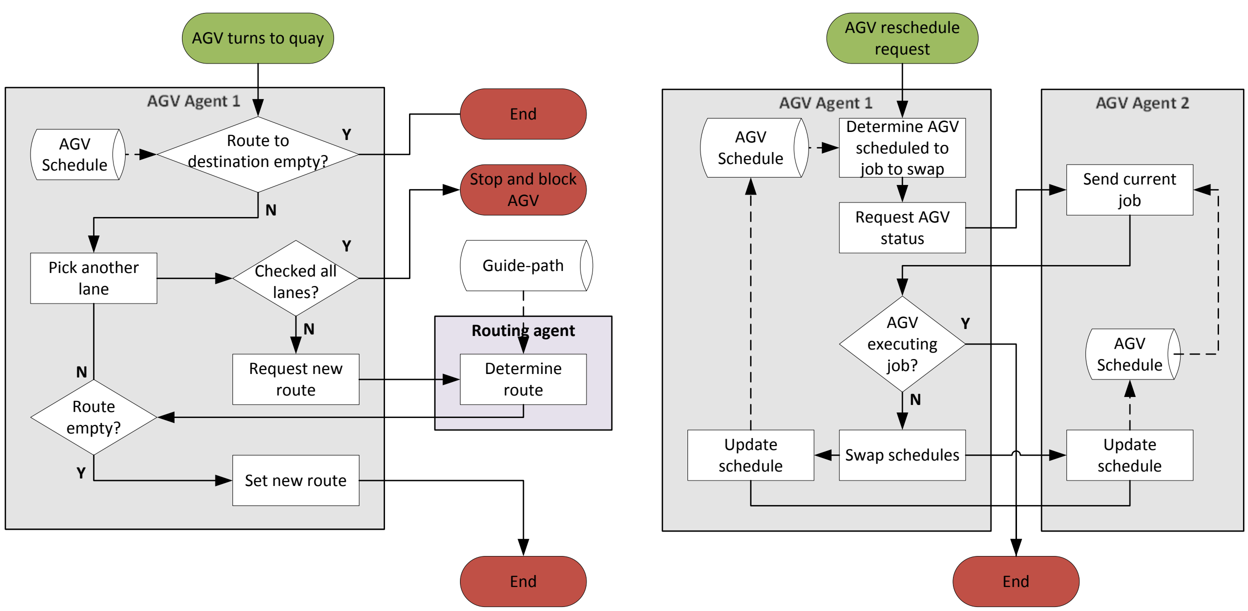

Figure 6: Flowcharts for rerouting AGVs (left) and rescheduling AGVs (right).

The AGV agent checks the availability of the quay lanes at the latest feasible point in time. When it finds a new empty route, it has time to maneuver to this lane as it is still on the S-N track. When all quay lanes are occupied, the AGV is stopped on this S-N track to avoid a deadlock underneath the crane. This AGV is then released when the first AGV underneath the crane leaves the area. In the situation that the arriving AGV is the first in line to be served at the QC and all quay lanes are occupied, an AGV underneath the crane is sent away to free up space. This AGV is rerouted to make a loop back to the QC.

Another form of agent intelligence is dynamic rescheduling. A rescheduling attempt is made each time an AGV agent wants to start the next job in its job list (as originally defined by the Dispatching Agent). Whenever it is time for the actual execution of the job, the AGV agent first checks the priority of the job. The priority of a job is defined by how many other jobs need to be processed by the QC before this job. A job with priority 1 thus means that the QC is ready to process it. AGV agents strive towards high priority jobs as they do not want to queue underneath the QC. When the next job on the job list has a priority equal to n, where $n > 1$, the AGV initiates a maximum of $n-1$ rescheduling requests, as shown in Figure 6 (right side). The AGV agent thus tries to make a swap with other AGV agents. It starts with the AGV agent who has the highest priority job. If this swap cannot be agreed upon, it continues to try to swap with the next AGV agent. In the example of Figure 6 (right side), AGV Agent 1 tries to swap jobs with AGV Agent 2 to get a higher priority job. The logic being that when AGV Agent 2 is scheduled to a priority 1 job and this job is not first on its job list, the QC will encounter waiting time as AGV Agent 2 first needs to process one (or more) other jobs with significant processing time. Therefore, it is better to swap this job to AGV Agent 1, which is available immediately, to lower the waiting time of the QC. On the other hand, AGV Agent 2 gets in return a lesser priority job, which is unfavorable to it. In our approach, this agent still accepts the swap, as it expects that AGV Agent 1 processes the job before AGV Agent 2 is ready to process the job that was swapped.

The agent-based modeling approach described in the previous subsections gives us the flexibility to build a model that can easily be tailored to an ACT and thus provides a high degree of reusability.

6 VERIFICATION AND VALIDATION

An important step of any simulation study is the verification and validation process. We followed the eight verification techniques listed by Law (2015).

- Debugging modules or subprograms. During the entire development of the model we made sure that every new sub-component was first tested separately. After the addition of any new component, we checked whether the component worked properly by debugging the component while running the larger simulation model.
- Running the model under a variety of settings. During guide-path development, we used a broad variety of input values to test whether the model responded appropriately.
- Traces. Traces were extensively used throughout the development period. This was done to verify whether the right methods were triggered at the right moments in time.
- Run under simplifying assumptions. During development we tested parts of the simulation model in a simple setting, before increasing the complexity. This was done, for example, by first allowing QCs only for unloading containers from a single bay to test the functionality of the QCs.
- Animation. The 3D animation capabilities helped us to verify many parts of the simulation model. For example, the loading and unloading of containers to and from a vessel was visually checked using the 3D animation (the z-component is not visible in 2D). Furthermore, the movement of AGVs was checked to see whether the predetermined routes were used and whether rerouting was handled correctly. Animation was also used to visually check whether the Conflict Handling Agent resolved any conflicts properly.

To validate the model, we used both black-box and white-box validation. The first was used to analyze the overall model behavior by performing experiments. The latter was used to validate sub-components of the model to see whether they perform as expected. We validated the model considering the loading and unloading of a single vessel, with sufficiently large and common call size of 5,000 moves using the following outputs: (i) deadlocks at the QCs, (ii) the traveled distance of the AGVs, (iii) AGV utilization, and (iv) the traveled distance per AGV. While varying the number of AGVs in the system, we analyzed these outputs to see whether the system behaves as expected. Based on this analysis, we concluded that the model is valid (see Figure 4 for an impression of the implemented model).

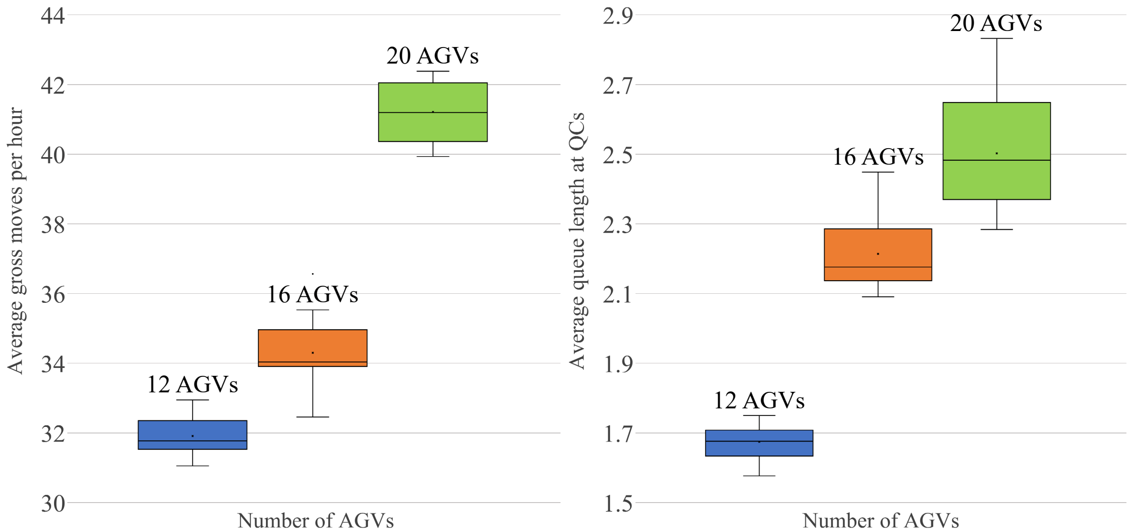

Figure 7: Simulation results for varying number of AGVs.

The validated model was used to analyze the system performance under various settings. Ultimately we would perform a more in-depth validation and test our model with various terminal characteristics as defined in Table 1. Here we illustrate the effectiveness of our agent-based approach by comparing the simulation results with typical values from practice. For this purpose, we use a terminating simulation using the same call size of 5,000 moves. The vessel is served by four QCs and 12 to 20 AGVs, with 5 replications per run. The stack area consists of nine stacks. The output of particular interest is the VTT, which is directly determined by the GMPH of the QCs serving the vessel. We know from the technical specifications that the GMPH per QC is around 50 to 55 moves per hour, whilst the actual operating performance for the terminal under study is around 30 moves per hour. From the left side of Figure 7, it can be seen that we obtain similar results with our ABS model. For the terminal under study, the 30 GMPH per QC are achieved with 16 AGVs, whereas we obtain a performance of 34 GMPH per QC with the same number of AGVs. The increased performance can be attributed to the use of our intelligent agent-based approach. When extrapolating the measured utilization of the QCs to 100%, we obtain roughly 52 moves per hour per QC, which is in line with the technical specifications.

An important feature of the enhanced agent intelligence is the ability to avoid deadlocks. During the service time of the vessel, no deadlocks occurred. By using dynamic routing and scheduling, the number of AGVs waiting at the QC decreased, whilst the QCs are sufficiently utilized. The right side of Figure 7 shows the average number of AGVs waiting at a QC. As expected, the average number of AGVs waiting at a QC increases with the number of AGVs deployed. Interestingly, the difference between 16 and 20 AGVs is relatively small compared to the increase in GMPH. This shows that our approach makes efficient use of the AGVs whilst avoiding crowding and deadlocks. Further results show that, given the number of AGVs, there is a proper balance between AGV utilization, AGV waiting time, and QC utilization, but these have been excluded for brevity.

These experiments have illustrated that our agent-based TM is a suitable way of modeling the container handling process at ACTs. The impact on the GMPH looks promising and more research is required to further quantify the impact on the GMPH given various scenarios at different ACTs.

7 CONCLUSIONS

This paper presents the design and implementation of a scalable and flexible agent-based simulation (ABS) model for the planning and control of AGVs at parallel oriented automated container terminals. We develop an agent-based Traffic Manager that serves as control application layer for all processes related to the horizontal transport of containers using AGVs. This Traffic Manager is capable of controlling all the operations at the quayside of a container, including elements such as conflict resolution and deadlock avoidance. Our ABS shows an increase in the GMPH at our case study and is a useful tool to test out the effectiveness - in terms of Vessel Turnaround Time - of agent-based control for different types and layouts of container terminals.

Future research directions include (i) developing a benchmark based on real-life data to further quantify the effectiveness of our agent-based approach, (ii) overcoming the weaknesses related to ABS such as scaling and real-time simulation, (iii) adapting the ABS to fit perpendicular oriented ACTs, and (iv) applying results at other terminals to assess flexibility and generality of the ABS.

REFERENCES

Cimino, M. G., F. Palumbo, G. Vaglini, E. Ferro, N. Celandroni, and D. La Rosa. 2017. "Evaluating the Impact of Smart Technologies on Harbor's Logistics via BPMN Modeling and Simulation". *Information Technology and Management* 18 (3): 223–239.

Dragović, B., E. Tzannatos, and N. K. Park. 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 (1): 4–34.

Fazlollahtabar, H., and M. Saidi-Mehrabad. 2015. "Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study". *Journal of Intelligent & Robotic Systems* 77 (3-4): 525–545.

Fotuhi, F., N. Huynh, J. M. Vidal, and Y. Xie. 2013. "Modeling Yard Crane Operators as Reinforcement Learning Agents". *Research in Transportation Economics* 42 (1): 3–12.

Gangwani, K. 2015. "In 2015 Terminal Productivity is the Key". https://www.ajot.com/premium/ajot-in-2015-terminal-productivity-is-the-key, accessed February 1st, 2018.

Garro, A., M. F. Monaco, W. Russo, M. Sammarra, and G. Sorrentino. 2015. "Agent-based Simulation for the Evaluation of a New Dispatching Model for the Straddle Carrier Pooling Problem". *Simulation* 91 (2): 181–202.

Gayathma, K., I. Sigera, and S. Cahoon. 2015. "The Factors Affecting the Container Vessel Turnaround Time at Port of Colombo". In *Proceedings of the IAME 2015 Conference*, August 23rd–27th, Kuala Lumpur, Malaysia, 1–17.

Gerrits, B., M. Mes, and P. Schuur. 2017. "An Agent-based Simulation Model for Autonomous Trailer Docking". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 1324–1335. Piscataway, New Jersey: IEEE.

Henesey, L., P. Davidsson, and J. A. Persson. 2009. "Evaluation of Automated Guided Vehicle Systems for Container Terminals Using Multi Agent Based Simulation". In *Multi-Agent-Based Simulation IX*, edited by N. David and J. S. Sichman, 85–96. Berlin, Heidelberg: Springer.

Huang, J., F. Wang, and N. Shi. 2014. "Resource Allocation Problems in Port Operations: A Literature Review". In *Proceedings - 2014 7th International Joint Conference on Computational Sciences and Optimization*, July 4st–6st, Beijing, China, 154–158.

Kaoud, E., M. A. El-Sharief, and M. G. El-Sebaie. 2017. "Scheduling Problems of Automated Guided Vehicles in Job Shop, Flow Shop, and Container Terminals". In *2017 4th International Conference on Industrial Engineering and Applications*, April 21st–23rd, Nagoya, Japan, 60–65.

Kim, K. H., and H. Lee. 2015. *Handbook of Ocean Container Transport Logistics*, Volume 220. Cham: Springer International Publishing.

Law, A. M. 2015. *Simulation Modeling and Analysis*. 5th ed. New York, NY, USA: McGraw-Hill.

Marinica, N. E., M. Capiluppi, J. A. Rogge, R. Segala, and R. K. Boel. 2012. "Distributed Collision Avoidance for Autonomous Vehicles: World Automata Representation". In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, June 6th–8th, Eindhoven, The Netherlands, 216–221.

Mes, M., and A. Douma. 2016. "Agent-Based Support for Container Terminals to Make Appointments with Barges". In *Computational Logistics*, edited by A. Paias et al., 80–95. Cham: Springer International Publishing.

Montoya-Torres, J. R., J. L. Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla. 2015. "A Literature Review on the Vehicle Routing Problem with Multiple Depots". *Computers & Industrial Engineering* 79:115–129.

Nabais, J. L., R. R. Negenborn, R. B. Carmona Benitez, and M. A. Botto. 2013. "Setting Cooperative Relations Among Terminals at Seaports Using a Multi-Agent System". In *IEEE Conference on Intelligent Transportation Systems*, October 6th–9th, The Hague, The Netherlands, 1731–1736.

Parthibaraj, C. S., P. Palaniappan, A. Gunasekaran, and N. Subramanian. 2017. "Multi-agent System with Iterative Auction Mechanism for Master Bay Plan Problem in Marine Logistics". *Maritime Policy & Management* 44 (6): 705–726.

Sharif, O., and N. Huynh. 2012. "Yard Crane Scheduling at Container Terminals: A Comparative Study of Centralized and Decentralized Approaches". *Maritime Economics & Logistics* 14 (2): 139–161.

Sun, Z., K. Tan, L. Lee, and E. Chew. 2013. "Design and Evaluation of Mega Container Terminal Configurations: An Integrated Simulation Framework". *Simulation* 89 (6): 684–692.

Tirschwell, Peter 2015. "Port Productivity is not Improving, Maersk CEO Says". https://www.joc.com/maritime-news/container-lines/maersk-line/port-productivity-not-improving-maersk-ceo-says_20150226.html, accessed 1st February, 2018.

Wibowo, R. A., A. Hidayatno, K. Komarudin, and A. O. Moeis. 2015. "Simulating Port Expansion Plans Using Agent based Modelling". *International Journal of Technology* 6 (5): 864–871.

Zheng, S., and R. R. Negenborn. 2014. "Centralization or Decentralization: A Comparative Analysis of Port Regulation Modes". *Transportation Research Part E: Logistics and Transportation Review* 69:21 – 40.

AUTHOR BIOGRAPHIES

BERRY GERRITS is a PhD candidate within the department of Industrial Engineering and Business Information Systems at the University of Twente, The Netherlands. He received a MSc in Industrial Engineering in 2016. His research interests are transportation logistics, multi-agent systems, supply chain management, and business process optimization. His email address is b.gerrits@utwente.nl.

MARTIJN R.K. MES is Associate Professor within the department of Industrial Engineering and Business Information Systems at the University of Twente, The Netherlands. He holds a MSc in Applied Mathematics (2002) and a PhD in Industrial Engineering and Management at the University of Twente (2008). His research interests are transportation, multi-agent systems, stochastic optimization, discrete event simulation, and simulation optimization. His email address is m.r.k.mes@utwente.nl.

PETER C. SCHUUR is Associate Professor within the department of Industrial Engineering and Business Information Systems at the University of Twente, The Netherlands. He received a PhD in Mathematical Physics from the University of Utrecht, The Netherlands. His research interests are: (closed-loop) supply chain management, vehicle routing, distributed planning, multi-modal networks, dynamic pricing, reverse logistics, layout problems, and warehouse modeling. His email address is p.c.schuur@utwente.nl.