MODELING OF INNOVATIVE TECHNOLOGIES FOR CONTAINER TERMINAL YARD STACKING SYSTEMS USING AN OBJECT-ORIENTED SIMULATION FRAMEWORK

Nikolaus A. Bornhöft\(^{(a)}\), Prof. Dr.-Ing. Bernd Page\(^{(b)}\), Dr.-Ing. Holger Schütt\(^{(c)}\)

\(^{(a)}\)(b)Department of Informatics, University of Hamburg, 22527 Hamburg, Germany
\(^{(c)}\)Institute of Shipping Economics and Logistics, 27568 Bremerhaven, Germany

\(^{(a)}\)nikolaus@bornhoeft.me, \(^{(b)}\)page@informatik.uni-hamburg.de, \(^{(c)}\)schuett@isl.org

ABSTRACT
Simulation provides appropriate techniques to evaluate the efficiency of container terminals. This is particularly beneficial when it comes to new concepts that have not been applied yet and thus no empirical data is available. In this paper we will discuss the “Environmentally Friendly Automated Container Yard Stacking System” developed by ZPMC Ltd. and distinguish it from existing terminal architectures. Thereto we will first describe the characteristics of the concept and the involved devices. Then we will show the capabilities of the discrete event simulation framework DESMO-J to model and simulate a container terminal design, and evaluate it to gain a qualified estimation about its operating efficiency. Moreover a post simulation animation shall aid to visualize the layout and the devices involved and to understand their interactions in service. Keywords: Discrete Event Simulation, harbor logistics, container-terminal, simulation framework

1. INTRODUCTION
Yard Stacking Systems have a substantial impact on the efficiency of a container terminal. For the last years the average size of newly build container vessels has been considerably increased. Therefore charging times and capital costs of lay times rise. Different yard stacking systems and the dimensioning of technical equipment lead to a wide range of durations of handling times. To pre-estimate the efficiency of a yard stacking system with a given configuration of technical equipment is a complex task, which analytic approaches cannot solve adequately.

ZPMC’s new container yard stacking system is one suggestion to organize intra-terminal transport processes. Until today, no real container terminal makes use of the ZPMC concept – one terminal using it is just in the planning phase (ZPMC 2010). Therefore it can be of interest to design a simulation model of a terminal implementing such a yard stacking system to get an idea of its operating efficiency. In particular through experimentation and animation a better understanding of the interactions of the involved devices, potential bottlenecks in the system and estimations of the efficiency of the entire terminal and the concept itself can be gained.

The remainder of this draft paper is structured as follows: Section 2 provides an overview of two container terminal architectures applied in practice and ZPMC’s new architecture, particularly discussed in this paper. Section 3 gives a rough outlook on different types of simulation software and explains the choice of the object-oriented simulation framework DESMO-J as the appropriate tool for this task. Then DESMO-J will be introduced in short. In Section 4 some aspects of the simulation model will be mentioned. Section 5 pictures the animation tool ViTO 3D and the options to join DESMO-J and ViTO 3D. In Section 6 conclusions will be drawn about the simulation of new technologies of terminal yard stacking systems and the possibilities DESMO-J and ViTO 3D provide for that task.

2. CONTAINER-TERMINAL ARCHITECTURES
There are different architectures in automated container terminals currently in use. These architectures mainly differ in the degree of automation and the types of utilized devices. To outline the new yard stacking system, we will first describe two types of architectures already in service.

2.1. Straddle Carrier-based Architectures
In many existing container terminals, the internal container transport is performed by Straddle Carriers. These are driver-guided vehicles with the ability of lifting, transporting and stacking containers. In Straddle Carrier-based architectures, there are several lanes and container handover points beneath each Quay Crane. Straddle Carriers move containers from their places inside the storage-blocks of the yard to handover points at the Quay Cranes respectively vice versa. Since Straddle Carriers need to be guided by drivers each, this architecture uses the least level of automation of all architectures considered in this paper. As they are human-controlled, container terminals based on Straddle Carriers also have the smallest requirements regarding the system in charge of the transport allocation. In Brisbane/Australia some automated SC are working at a container terminal. Some suppliers developed so called Shuttle Carrier, which are used for
transportation only (not for stacking). (For more information see Schuett 2009)

Figure 1: Straddle Carrier (Konecranes)

2.2. AGV-based architectures
Architectures with the highest level of automation in existing container terminals use AGV-technology. An AGV (Automated Guided Vehicle) is a chassis, led by a computer based central container-terminal control system. As AGV are guided chassis, they do not have lifting abilities. Instead, they are loaded and unloaded directly by quay cranes at seaside or by the gantry cranes at a storage block. In AGV architectures a higher level of synchronization between the chassis and the quay cranes needs to be ensured by the control system than in a Straddle Carrier-based approach. Due to the fact that the loading and unloading process is executed directly by the cranes areas used as container buffers are not required. Moreover, guidance of the vehicles needs extremely precise controlling techniques.

Figure 2: Automated Guided Vehicle (HHLA)

2.3. The new System by ZPMC
ZPMC’s new container yard stacking system separates the container-transport between the stacking-area at the yard and the quay cranes into two steps. One step covers the transport parallel to the quay line, the second step describes the transport into a stack to a dedicated location. The first one is conducted on the Low-Frame-Bridge. This bridge is a steel frame situated below the rear sides of the quay cranes approximately seven meters above ground. It leads over the full length of the quay line and thereby connects the front sides of all stacks of the yard. It consists of several parallel lanes. On the lanes there are two types of devices working – Flat-Trolleys and Lifting-Trolleys. Flat-Trolleys are carriages that move on the Low-Frame-Bridge. They are loaded and unloaded with containers by a quay crane or a Lifting-Trolley. Lifting-Trolleys also move on a lane of the Low-Frame-Bridge. They are mobile cranes and perform container handovers between a Ground-Trolley (Ground Electromotion Flat-Trolley) and a Flat-Trolley. Thereto they are capable of lifting and lowering containers from and to the height of a Lifting-Trolley and a Ground Trolley and rotate it as well. Lifting-Trolleys may pass over Flat-Trolleys but not over other Lifting-Trolleys on their lane. Also Flat-Trolleys may not pass over other Flat-Trolleys. Ground-Trolleys transport containers between the handover point with the Lifting-Trolley below the Low-Frame-Bridge and a handover point inside a stack with a gantry crane.

Compared to the Straddle Carrier approach, this architecture requires much less manpower in service. Compared to an AGV-based architecture it reaches a similar level of automation. The central idea of this concept is a strong simplification of the local control in each system component. Every device involved only has to conduct movements on a straight line. The disadvantage is the increased complexity in synchronization necessary between the components involved. Furthermore the dimensioning for each single component must be chosen appropriately to avoid bottlenecks that slow down the whole system. To estimate the performance of the architecture in a container terminal in service, realistic assumptions about the control and the workload of the terminal must be made, thus allowing building a model as valid as possible. (ZPMC, Shilong 2007)

Figure 3: Low-Frame-Bridge (ZPMC)
3. SIMULATION

To examine the new yard stacking system in a simulation model it is necessary to choose an appropriate simulation tool first. Today lots of different simulation programs are available with a wide range of different functionalities. These subdivide into commercial and free tools. Commercial tools often cover support for various tasks in modeling, simulating, evaluating and animating a system. Some examples are:

- Plant Simulation by Unigraphics Solutions GmbH
- Witness 2007 by Lanner Simulation Technology GmbH
- SPEEDSIM by DUALIS GmbH IT Solution (Lindemann and Schmid, 2007)

In the field of container terminal simulation some specific tools have been developed and are in use for planning and optimising container terminals worldwide. Some of them are described in Schuetz (2009).

- SCUSY (Simulation of Container handling Units SYstem) by Institute of Shipping Economics and Logistics, Germany
- FLEXIM CT by Flexim Simulation Products Inc. (USA)

The commercial tools as well as the container terminal simulation tools mostly contain predefined components for a specific field of application that can be assembled on drag and drop basis to ease modeling effort. Since such tools are proprietary systems and the interface is not fully specified it can be more difficult to model non-standard systems not yet supported by predefined components.

Apart from the commercial tools, there are free alternatives. They mostly are developed in an educational context at universities. Most of them are open source, so that one can define the model’s behavior down to code level. Examples for free simulation tools are:

- DSOL, developed at the University of Technology in Delft, The Netherlands
- JavaDEMOS, from the University of Duisburg-Essen, Germany
- SSJ, developed at the University of Montréal
- DESMO-J from the University of Hamburg, Germany

Since DESMO-J is a powerful simulation tool supporting the process- and the event-oriented modeling style (as well as their combination). Use of the process-oriented modeling style appears appropriate, if most of the events in a system can be associated to a single system entity, the event oriented one, if that kind of association does not make sense. Furthermore there existed previous expertise about it, so we chose to use it as implementation tool of the hybrid model of the container terminal.

3.1. DESMO-J

DESMO-J (Discrete Event Simulation MOdeling in Java) is a Framework for implementing discrete event simulation models. It is developed at the University of Hamburg and licensed under the Apache License 2.0. The reader is referred to Page and Kreutzer (2005) and to the web page http://www.desmoj.de, from where the binaries, the source code and other resources can be obtained. The framework comprises a collection of components to support model design, experimentation and statistic evaluation. DESMO-J implements the separation of model and experiment. The components of the framework subdivide into black box and white box components. The black box components are ready to use without change or adaptation (e.g. subclass creation). They provide the complete simulation infrastructure, encapsulating a scheduler and an event-list behind the façade (Gama et al, 1995) of an Experiment class. Additional ready-to-use components are stochastic distributions of the type Distribution, components for statistic evaluation of type StatisticObject and classes representing queues and other means of synchronization of type QueueBased. The white box components, also called hot spots represent the model-specific behavior. They are realized as abstract classes and need to be implemented by the modeler. Since DESMO-J supports both the event-oriented and the process-oriented world view, system behavior can be described using the more appropriate modeling style.

The event-oriented world view is a bird’s eye-view on the model, interpreting the system components as Entity. Entity behavior, specifically the state changes an entity undergoes at an instant in time, is described by Events and ExternalEvents System dynamics arise from sequentially processing the events scheduled on the event list.
In contrast the process-oriented modeling style takes the worm’s eye view and combines the system components and their behavior (including time-consuming activities) in SimProcesses. All changes in the process-oriented style are directly defined inside the SimProcess objects. System dynamics arise from process interaction.

Combining both world views, DESMO-J provides a hybrid modeling style. Thereto Events may change the state of SimProcesses as well, as SimProcesses may schedule new Events.

The great flexibility of DESMO-J arises from the hot-spot design of the framework. To describe the behavior of a system one can use the whole expressive power of the Java language. Furthermore different Java libraries provide much functionality to extend a simulation model for a certain purpose.

The combination of DESMO-J and ViTO 3D can be used to play post simulation animation. It displays movements, rotations etc. of 3D objects. These need to be created in advance to the simulation as a model layout. During an experiment every spatial change of a model component is notified in an animation file via xml. After a simulation run this file may be viewed in the ViTO 3D visualizing all model behavior during the experiment.

The combination of DESMO-J and ViTO 3D appeared to be suitable for simulation and post simulation animation. Since DESMO-J is a Java framework, it is easily possible to implement appropriate methods for an xml-output by using Java libraries for xml processing. ViTO 3D is a suitable solution to analyze an experiment run through animation since it provides a free movement camera mode, a mode to focus the camera on a single object and slow motion and fast motion, modifiable during the animation and setting up scenes for presentation purpose.

4. THE CONTAINER TERMINAL MODEL
To explore ZPMC’s new container yard stacking system a complete container terminal model was created. In order to receive reliable estimations for the new technology in service, the model was designed using components already running in existing terminals. These establish a frame, embedding the new components. Thus the model is composed of quay area, including quay cranes and arriving and departing vessels, gates, a stacking area and the components of the new architecture in a narrower sense, the Low-Frame-Bridge, Flat- and Lifting-Trolleys as well as Ground Trolley. Thereto the arrangement of the model components is kept close to the suggestion from the presentation materials by ZPMC (see ZPMC).

As the model displays an architecture, which is not yet implemented in a container terminal, only a limited model validation could be applied due to missing real data. Therefore plausibility checks could only be used to validate the model behavior of single components.

To model the container terminal and the control of the involved components the hybrid modeling style has been applied. This anticipates the approach to split up complex model-wide control into much simpler units of local control inside the components on the one hand and a remaining part as a central control on the other hand.

The container terminal components as persistent entities of the model are convenient to be modeled a SimProcesses. They specify a local behavior during a container transport. This includes the interactions with other model components also dedicated to a certain transport and all movements of the system component and its subcomponents.

The other parts and incidents of the system are designed in accordance to the event oriented modeling style. This covers the transient objects of the model as container transport orders, vessels and trucks and their arrivals and departures. They were modeled as Events and Entities. This appeared to be reasonable since the appearance at and the entrance to the system is not triggered from inside the model and therefore not part of the behavior of a model component.

The central model control determines which container transport orders are to be executed at a certain point in time and which terminal resources are involved in the transport. Thereto a SimProcess checks several entrance conditions each time an order arrives at the system or a model component becomes idle. In that case the required resources are allocated to the transport order and a system entrance event is performed, starting the container transport and changing the state of the components. (For a more detailed description of the model see Bornhöft 2009.)

5. ANIMATION
To achieve a further understanding of the interactions of the model and visualize potential bottlenecks and inconsistencies of the behavior of the model an experiment can be animated. For this purpose ViTO 3D Player (the ISL Real-time Animation Player of ViTO) is used.

ViTO 3D can be used to play post simulation animation. It displays movements, rotations etc. of 3D objects. These need to be created in advance to the simulation as a model layout. During an experiment every spatial change of a model component is notified in an animation file via xml. After a simulation run this file may be viewed in the ViTO 3D visualizing all model behavior during the experiment.

The combination of DESMO-J and ViTO 3D appeared to be suitable for simulation and post simulation animation. Since DESMO-J is a Java framework, it is easily possible to implement appropriate methods for an xml-output by using Java libraries for xml processing. ViTO 3D is a suitable solution to analyze an experiment run through animation since it provides a free movement camera mode, a mode to focus the camera on a single object and slow motion and fast motion, modifiable during the animation and setting up scenes for presentation purpose.
6. SIMULATION RESULTS
To receive qualifying analyses, simulation runs of several scenarios were carried out. Comparing the results of these experiments to scenarios of models with other transportation systems, a lower productivity is recognizable. Evaluation of the experiment results and the animation files indicates long waiting times at the handover points between the different system components. While the control of each single component is much simpler than those in other terminal architectures, the number of handovers per container transport is significantly higher. Furthermore all of these points are limited to one handover at once. If a device must wait for another, it is blocked for the whole time period. E.g. waiting of a Flat-Trolley even blocks a part of the lane for all Flat-Trolleys assigned there. This leads to the hypothesis that the sophistication of the control of the whole system has a much greater impact on the productivity than it has on other container terminal systems. Therefore a good estimation for the efficiency of the system is only available with a fairly good global control system.

A conceptual separation of material flow and control flow appears to be necessary to develop such a sophisticated control. Emulation as a special kind of simulation technique offers that separation needed. An emulator is a model of a system component that provides all interfacing behavior of the component itself. Thus a part of the system may be replaced by its representing emulator. Subsystems that may be emulated are devices, groups of devices and controlling systems, as long as they can be subsumed under a common interface.

The Virtual Container Terminal Optimisation tool ViTo, mentioned above, (see Puchert et al. 2008) allows distributed simulation of container terminal processes by combining emulators and supporting components to a working model. Therefore the core components of ZPMC’s concept, the Low-Frame-Bridge, including Flat- and Lifting-Trolley and the Ground-Trolleys will be included as one single emulator into ViTo. Covering the terminal-internal transport, they can be analyzed in different technical environments. In addition different control systems may be set upon the model to analyse their specific productivity.

As part of ViTO, the container terminal model may be reused not only for planning tasks, but also will support the development of an optimized control system and the start-up phase of terminals using this new technology.

7. CONCLUSIONS
Discrete event simulation is an appropriate means to evaluate the efficiency of container terminal processes. Particularly it is suitable for new technologies not applied yet for which no empirical data is available. Some estimations of the operational efficiency as compared to established technologies can be derived from the simulation studies.

Since it is a very flexible tool, DESMO-J proved to be adequate to model and simulate very specific systems, because there is no need for any predefined model components.

The combination of DESMO-J and ViTO 3D appears to be convenient for simulation and post simulation animation since the Java provides libraries for xml-processing. Therefore both tools supplement each other in an optimal way.

The model appeared to be very sensitive to the global system control. Thus the system must be checked using a quite sophisticated control to gain reliable estimations about its productivity. Therefore the model will be included into an emulation framework to allow an analysis as an emulator, interacting with other device emulators and controlled by an exchangeable dedicated controlling system.

REFERENCES


ZPMC. The Environmentally friendly Automated Container Yard Stacking System. Presentation film


AUTHORS BIOGRAPHY

Nikolaus A. Bornhöft received his diploma in Information Systems at the University of Hamburg, Germany in 2009 During his studies he set focal points on computer simulation as well as psychology. His diploma thesis (Bornhöft 2009) bases thematically on cooperation between the University of Hamburg and the ISL.

Prof. Dr.-Ing. Bernd Page holds degrees in Applied Computer Science from the Technical University of Berlin, Germany, and from Stanford University, USA. He has been professor for Applied Computer Science at the University of Hamburg since the mid-eighties and researches and teaches in the field of Computer Simulation as well as in Environmental Informatics. He is (co-) author of more than 120 scientific papers and of 9 text books.

Dr.-Ing. Holger Schütt is the head of the section Optimization and Simulation within the Information Logistics department of the Institute of Shipping Economics and Logistics (www.ISL.org). The focal point of his work is the simulation and optimization of business processes as well as material flow in logistic systems. In former times he worked as a computer scientist and project manager at the biggest harbor and logistic provider in Hamburg. He was responsible for the development of various administrative and planning systems (e.g. dangerous goods information system, berth planning system) as well as for the simulation based consultancy of the new fully automated container terminal Hamburg Altenwerder: Currently he is working in the field of emulation software based on a distributed network as well as on implementing ecological analysis aspects into logistic simulation models.