Proceedings of the 11th European Simulation Symposium (ESS 1999).
October 1999. Erlangen [SCS]. ISBN 1-56555-177-x

Detailed Simulation of the containerflows for the IPSI-concept

Hans P.M. Veeke and Jaap A. Ottjes,
Faculty of Mechanical Engineering and Marine Technology
Delft University of Technology
Mekelweg 2, 2628 CD Delft, the Netherlands
e-mail: H.P.M.Veeke@wbmt.tudelft.nl
J.A.Ottjes@wbmt.tudelft.nl

Key Words
Transportation, Discrete Simulation, Process Oriented, Decision Making.

Abstract
This paper describes the way of detailed modeling and simulation for a new container-handling concept.. The simulation-study is part of the European project 'Improved Port/Ship Interface'. Simulation was asked for at a moment in the project where lots of decisions had to be made yet. Therefor the emphasis of the simulationstudy was on conceptual modeling followed by a programming phase divided over several parties. The simulation supported decision-making, proved the feasibility of the concept and contributed to a detailed cost-calculation.

Introduction

During the last three years a new terminalconcept has been developed for short-sea containershipping, the so-called Improved Port Ship Interface (IPSI) [1]. In this concept Automatic Guided Vehicles (AGV's) are being used for transport of containers between stack and ship.
Thereby AGV's move into the ship to get or put the containers; so no quaycranes are needed. Special ships will be needed to accommodate AGV-traffic; they have a capacity of about 500 TEU in two layers and must have a berthtime of less than 2 hours. These heavy demands on throughput lead to a concept with virtual AGV-trains, transporting containers on "frames" between ship and a central transfer-area (the marshalling area).

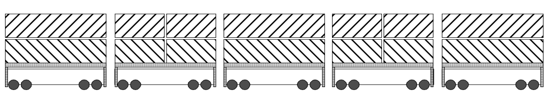

Fig. 1. AGV-train with frame and containers.

On the landside of the terminal special service-centers for trucks and rail handle containers by means of straddle-carriers.
During the project simulation was needed for two purposes:

1. Gain insight into dimensions of capacity needed and sensitivity of the concept for different arrivalpatterns and shiploads.
2. Prove the feasibility of the AGV-concept.

This article deals with the second goal. The first goal is extensively described in the paper of J. Ottjes.
To prove feasibility a detailed simulationmodel was developed. Par. 1 describes the modeling method used. Par. 2 shows the structure of the model, par. 3 explains the simulation-aspects and finally par.4 shows the main results of the model.

1. Modeling concept

At the start of the modeling, many decisions had to be taken yet. The layout of the terminal was not clear. Also the precise containerflow was not defined. Should there be 'buffers' between service-centers and AGV-system and should carriers enter the marshalling area or should containers between service-centers and marshalling area be transported by AGV's?
Because of these uncertainties, it was decided to develop a general terminalmodel. Also, several parties (Delft University, Fraunhofer Institute) were involved and should be able to work independently on the model.

Looking at containerterminals, there will always be well-defined positions, where containers are transferred from one transportmodality to another. Trucks arrive at identified positions where carriers receive the container, AGV's drive to specific points to receive or deliver containers, cranes put containers on stackpositions to pick them up in a later stage etc. Containerhandling will always be an alternation of transfer- and transport-functions. Fig. 2 shows an example.

Transfer is being done on *transferpoints,* which are denoted by triangles in fig. 2 and depending on the choice of equipment for each part of the flow, different terminalmodels can easily be build, if the procedures for transferpoints are defined.
The transfer of containers on transferpoints can be immediate or postponed. In this last case the transferpoints are in fact stack-positions, so containerstacks can also be modeled as transferpoints. Therefor: *in general a containerterminal consists of a set of "transferpoints" to accommodate physical transfer*

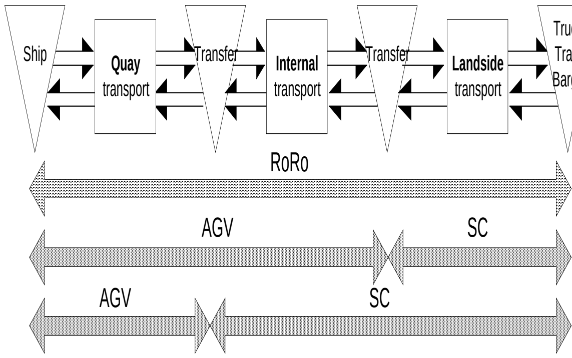

Fig. 2. Transferpoints in a containerflow
RoRo = Roll-on/Roll-off SC = straddle-carrier

of containers between transportmodalities. Each transferpoint has a unique position and may contain one or more containers. In our case also the handling of frames on transferpoints should be facilitated. Only the transferprocedures on a transferpoint must be generalized; the process between transferpoints however is specific for the equipment being used.
So, the first 'common' part of the model is a "transferpoint"-unit; each specific service-center can be developed using this unit.

Now, we are able to model the containerflow physically; but how do we *control* the flows? How do we decide which container must be transported between which transferpoints? Normally each part of the containerflow, handled with the same equipment or assigned to one modality is considered a 'subsystem'. Each subsystem has its own local control. Without central control however, coordinating the actions of each subsystem our terminal will not function optimal. This central control doesn't have to know **how** a subsystem performs its jobs, only **what** jobs it is doing and assumed to be doing. This depends solely on the information exchanged between subsystems and central control, which is therefor called the 'information unit'. This unit also expands containerjobs into 'equipmentjobs. An equipmentjob can be performed completely by one subsystem. The information unit controls the release of equipmentjobs. This is shown in fig. 3.

On startup of the model each subsystem must identify itself to the information-unit and define its transferpoints, that are shared with other subsystems. Local internal transferpoints can be defined without notifying the information unit. Transferpoints are grouped in so-called Transfergroups, for ease of routing.

2. Model structure

The complete model structure now becomes clear. First we have two common modules: the

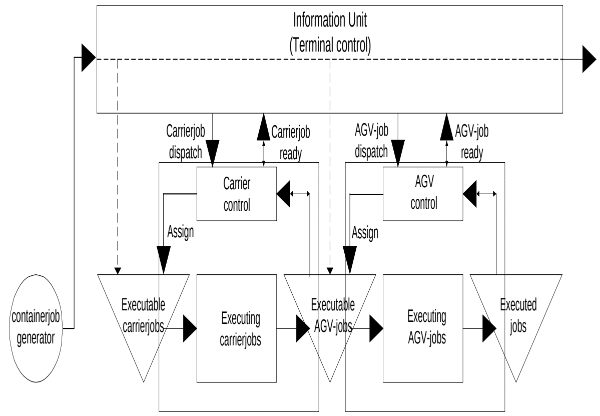

Fig. 3. Controlling the containerflow

"transferpoint-unit" for the physical flows and the "information-unit" for controlling the physical flows. Next to it several subsystems can be modeled and they will be called 'servicecenter"-units. At this point in the project the decision was made to develop 3 servicecenters: one AGV-servicecenter, one Truck-servicecenter and one Rail-servicecenter. The last two will be equipped with straddle-carriers. The position of a separate Barge Servicecenter however was not clear yet., but could be added later if necessary (see fig. 4a).

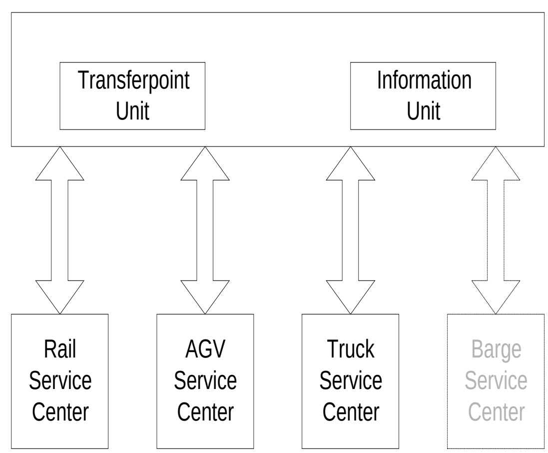

Fig. 4a. Model Structure
There were also 3 basic terminalconfigurations developed, but for this article we will concentrate on one. In this configuration the carriers of the truck- and rail-servicecenters must handle the containers in the central transferarea, so AGV's only deal with container-transports between the ship and this transferarea (see fig. 4b).

Proceedings of the 11th European Simulation Symposium (ESS 1999).
October 1999. Erlangen [SCS]. ISBN 1-56555-177-x

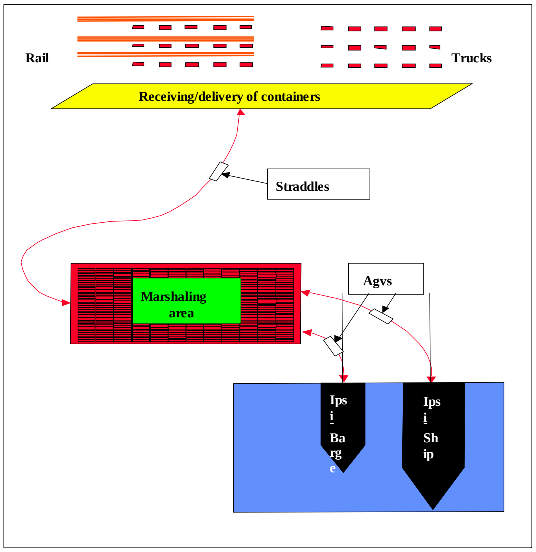

Fig. 4b. Schematic Layout

Because each servicecenter is connected to one external modality, jobs will be generated there. For example, when a ship arrives, jobs must be generated for each container in the ship by the AGV-servicecenter. The information-unit however contains a common method to create a containerjob and a routing through the terminal, depending on the available servicecenters and transferpoints. For example, when a container is meant for a truck the containerjob will be expanded into an AGV-job and a carrierjob for the Truck-servicecenter. It's also up to the information-unit at what moment the carrierjob will be released to the Truck-servicecenter.

3. *Simulation*

Lots of effort has been put in defining the modelstructure and describing transferprocesses. This pays off when starting to write the simulation-model. It became clear that about 80% of the programmingtime would be spent in writing control-algorithms. And although the transportprocesses look quite simple, the local control (traffic, collision-prevention) of the AGV's is complex. The next demands were formulated for the simulation-technique:

- the model must be object-oriented
- the model must describe processes, not events
- the AGV-control must be realistic, so collision-detection must be programmed and not simulated in the animation-code.
- Detailed animation must show the correct working and verify the feasibility of the concept.

The project decided therefor to develop the model in the simulation language MUST, combined with its animation extension MAX. MUST is in fact a toolbox for Borland Pascal, so all facilities of this general programming language are available (in the meantime a similar package TOMAS became available, which is a toolbox for Delphi).

First the objectdefinitions were made for the TP-unit and Info-unit. The global structure of objects in the TP-unit is given in fig. 5.

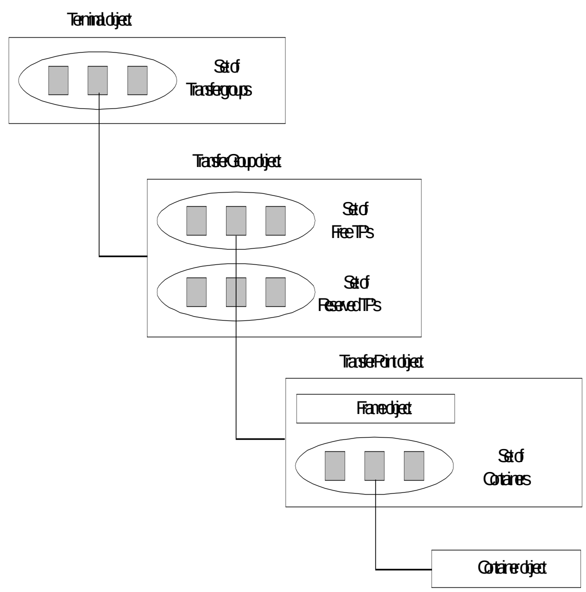

Fig. 5. Objectstructure of TP-unit.

After this the servicecenters were modeled independently. The AGV-servicecenter models the AGV-behaviour in great detail. Speed, acceleration and curves were implemented according to realistic AGV-specifications. Collison-prevention was developed by means of a claiming-mechanism that used rectangular 'claims' ahead of the vehicle. Claims are granted by a central traffic control, based on the knowledge of current AGV-positions and routes. Claims may normally not overlap, but

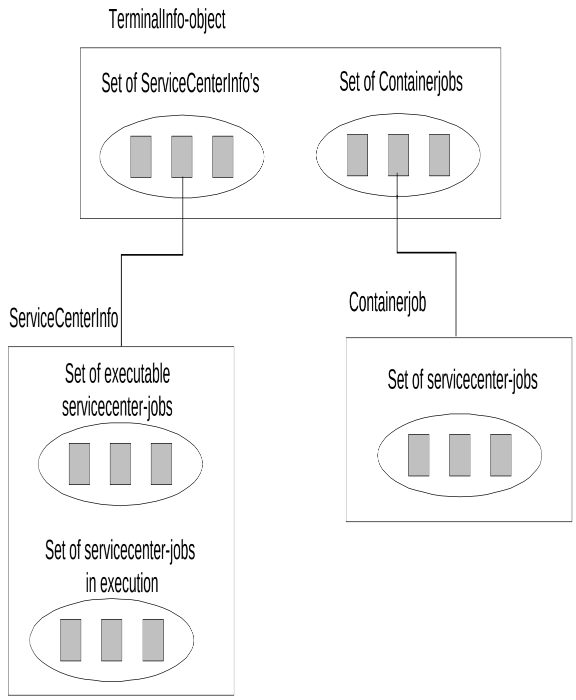

Fig. 6. Objectstructure of Info-unit.

when driving in a train-formation overlap is transformed to train-level rather than AGV-level. The routing of AGV's is in this case quite simple: there are only two ways for each deck of the ship. Routing however will become more complex at the moment barges will be included in the IPSI-terminal.

4. Results

AGV-trains of 10 AGV's are considered in the model. In reality the row length of IPSI-vessels varies between 7 and 11 containerframes. For proof of feasibility the consequences are:

- the simulation must show that there is enough time left to unload and load the 11-th frames. So the total handling time of a ship must be significantly less than 2 hours.
- In case of shorter rows a shorter train is needed. Then the unneeded AGV's may already be sent to a next row in the marshalling area. These routes are much shorter than routes to and from the ship, so this can always be accomplished.

Given the layout of fig. 4b. the model showed that the mean time for a cycle (from marshalling area to ship and back) with a single AGV-train was 425 sec. 20 Train-moves must be done for a single ship, so this can be done in 20 * 425 = 8500 sec. So to meet the 2-hour demand we need 2 AGV-trains of 10 AGV's.

Driving with 2 AGV-trains we have to prevent deadlocks and unnecessary waiting. For this goal a controlling mechanism was added, that uses "traffic-lights": one for leaving the marshalling area and one for leaving the IPSI-vessel. These lights determine the moment an AGV-train is allowed to leave. By these lights the cycletime will increase. The next experiment proved this: the cycletime was now 493 sec, so the complete ship was handled in 4930 sec, still far below the allowed time.

Because increasing the number of AGV's is only useful in steps of 10 AGV's, the projectteam decided to choose for 20 operational AGV's.

After that combined experiments were done with a rail- and truck-servicecenter. From these experiments it was concluded that –under the condition of a 4-hour shipcycle- 11 carriers are needed.

Finally, a detailed animation was made to prove the feasibilty of the concept in a visual way.

References

Pedersen, J.T., Ottjes, J.A., Veeke, H.P.M., 1999. IPSI: A revolutionary concept for intermodal transport, Research report: Project funded by the European Commission under the Transport RTD programme of the 4th framework programme', sub area 'Transport'. Contract No WA-95-SC-140.

Veeke, H.P.M. and Ottjes, J.A. 1999, Problem Oriented modeling and Simulation, Proc. 1999 summer computer simulation conference (SCS'99), Chicago, Illinois, pp. 110-114, ISBN #1-56555-173-7.

in 't Veld, Prof. J., "Analysis of organisation problems", Educatieve Partners Nederland BV, 1998, ISBN 90 11 045947 (in Dutch)

Mannarino, G.S., Henning, G.P., Leone, H.P.,Metamodels for Information System Modeling in Production Environments, Proc. Second International Conference on the Design of Information Infrastructure Systems for Manufacturing DIISM '96