

DISTRIBUTED E-SERVICES FOR ROAD CONTAINER TRANSPORT SIMULATION

Csaba Attila Boer
Erasmus University Rotterdam
Faculty of Economics
Department of Computer Science
P.O. Box 1738, 3000 DR Rotterdam
The Netherlands
acboer@few.eur.nl

Arjen de Waal
TBA Nederland
Vulcanusweg 259a, 2624 AV Delft
The Netherlands
arjen@tba.nl

Alexander Verbraeck
Delft University of Technology
Faculty of Technology, Policy and Management
System Engineering Department
P.O. Box 5105, 2600 GA Delft
The Netherlands
a.verbraeck@tbm.tudelft.nl

Bas van Eck and Jerry Seager
ILLYAN
Sarphatistraat 642, 1018 AV Amsterdam
The Netherlands
info@illyan.nl

KEYWORDS

Planning and Scheduling, Distributed Simulation, Architecture, Container Handling, Port, e-Services.

ABSTRACT

This paper describes a recently carried out project that aims to improve the handling process of trucks at new container terminals in ports. The simulation models of the truck companies and the ones of the container terminals might be separately developed by different parties applying different simulation packages. In this paper we give an approach for creating a distributed environment that supports the interoperability between these different models. Further, we introduce a planning and scheduling system that carries out the negotiation between the port terminal and truck companies to negotiate time slots for arriving at the terminal. This planning and scheduling system is included in the distributed environment as well and thereby becomes part of the simulated port and transport system.

1 INTRODUCTION

The FAMAS research program (de Hartog et al., 2001) intends to conceptualise and design new container terminals for the future port of Rotterdam. The involved organisations intend to apply the recent technical innovations and attempt to avoid the occurring problems on the present container terminals. One of the difficulties that terminals are faced with is the handling of the truck arrivals. The current situation sometimes results in large number of trucks waiting in excessively long queues, as they arrive more frequently than they are served. This situation especially arises in peak hours and in particular days when the number of trucks drastically in-

creases. Due to the limited number of serving cranes and the limited capacity of parking places these trucks are confronted with delays and a costly situation. This paper introduces a real planning system for scheduling the arriving time of the trucks at the terminal. The new planning system requires the truck companies to register the trucks at the terminal administration before delivering or picking up a container. The terminal administration together with the administration of the truck companies negotiate an arrival time that is acceptable for both sides. We deal with two main problems: *negotiation* from both sides and the *interoperability* of the systems of the negotiating parties. We approach the negotiation problem with an agent-based planning and scheduling system, which intends to negotiate based on some business rules provided by the truck and port authorities. To solve interoperability we use web-services architecture based on Extensible Markup Language (XML). Testing the effectiveness of the proposed scheduling system and its algorithms takes place in a distributed simulation environment where the trucking companies and the detailed port handling system are built as separate simulation models, and the planning system is available as a separate application that has to interface with the simulation models.

The paper is structured as follows. Section 2 gives a short introduction of the trucks scheduling problem and proposes a distributed modelling approach. Section 3 describes the conceptual distributed simulation model and additionally covers the interoperability between the simulation models. Section 4 introduces the participant models (federates) of the distributed system. Furthermore, the implementation aspects are given in this section as well. Section 5 contains some results of different experiments. Concluding remarks and directions for future research are provided in Section 6.

2 DETAILED PROGRAM DESCRIPTION

In order to deliver or pick up a container from the terminal in the proposed system, we distinguish different processes. First of all the truck companies that intend to deliver or pick up their containers to/from the terminal have to contact the terminal operator in order to make an appointment. The truck companies usually have a request, which is a desired arrival time at the terminal. Because of the fact that the container terminal has limited capacity (limited cranes, limited parking places, etc.) it might happen that many truck companies request the same time slot. To be able to guarantee truck companies a reasonable maximum turn around time (e.g. 30 minutes 95% reliable), the terminal will set a maximum value for the amount of trucks it grants a timeslot during the booking process. The reservation of a timeslot takes place in a multi-agent system. The agents negotiate on reservations on behalf of the truck companies and the terminal operator. There are some constraints, such as the currently available trucks, the currently available drivers, the time limits for the trucks and drivers (e.g. not driving at night), which the negotiator agents need to take into account. After obtaining a time slot (that describes the proposed arrival time at the container terminal) which is adequate for both parties the truck can drive to the container terminal. Unfortunately, the driving time cannot be precisely predicted due to traffic delays that might occur. Therefore the drivers sometimes have to count on some delays. Finally, the truck arrives at the terminal where it delivers or picks up the containers.

Based on the previous description we distinguish four separate processes (Figure 1). The truck requests can be either generated automatically by the truck simulation models or they can be generated in real time by a user. The request includes the generation of the desired arrival time, the type of container, etc. The planning and scheduling system attempts to provide the best time slot for each truck. Furthermore, the driving process might generate some expected delays due to traffic jams. The container handling operation is carried out in a detailed container terminal model.

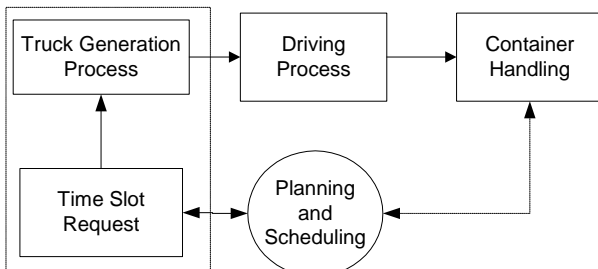


Figure 1. The main interoperability processes

The design and development of the whole complex model can be done monolithically (one big model using one package) or in a distributed way (well-distinguished models designed and developed in one or more packages). In the final stage of the project we would like to test the simulation models with the real planning and

scheduling system that uses high level negotiation mechanisms with recent technologies, such as intelligent agent based technology, web services, XML, etc.

The simulation model of the container terminal is highly detailed, which is essential in order to give a realistic handling time prediction, as the efficiency of the truck handling is very dependent on the other processes in the terminal, and there are many shared resources. This complex model already exists and can run without any external planning and scheduling system.

The second ingredient for the simulation is the driving model for the trucks that go from the trucking company to the port, with or without container(s). Different types of driving models exist (micro and macro traffic simulations) depending on the level of detail in which they were developed. Choosing a highly detailed model will lead to a more accurate result, but also requires very detailed input. Several simulation environments for road traffic were developed in the past years by several groups from different institutes.

Finally, a model component for the trucking company is needed. This model part generates a request, negotiates with the intermediary, and generates a truck to be sent to the container terminal using the road system. This generator is quite simple compared to the previous models.

Choosing a monolithical approach for designing and developing this complex system in one simulation model would result in a lot of work. Either the detailed port model, or the traffic models should be rewritten in order to integrate it with the other model parts. Furthermore, it would be extremely difficult to implement the agent based negotiation system in a simulation language – and actually again a waste of resources, as this system has already been built in the case we are studying. The monolithical approach that most simulation environments support as the only choice, always pose such problems in complex modelling tasks where different model parts from different background disciplines need to be integrated. In cases where other types of systems have to be included as well, the problem is even more aggravated.

In contrast with the monolithical approach we have used a distributed approach in which the possibility to interface with real planning and scheduling software is possible, and the existing developed simulation models and algorithms can be preserved.

3 THE CONCEPTUAL DISTRIBUTED MODEL

By, applying the distributed modelling approach the whole complex system can be designed and developed quite realistically. Different participants (organisations) can develop their own model without sharing their business logic. For example the truck companies concentrate on the generation process while the terminal operators focus on the container handling. They share only the relevant information between them, namely that which is necessary for the negotiation process. The planning and scheduling system is also unique, and is not consid-

ered as a part of a certain simulation model. It keeps the contact with the whole participant models and requires only internal information that is necessary for the negotiation process. A big advantage of this approach is thus, information hiding and furthermore, it increases the individual work because different modellers (from different organisations) can work in parallel (Taylor, et al. 2003).

The challenge we are faced with by applying distributed simulation modelling is that the individually designed and developed simulation models need to be coupled in order to form a consistent simulation federation. During the simulation run the different models interoperate with each other. For this reason *interoperability* must be achieved between the different simulation models (Fujimoto, 2000).

In order to achieve interoperability between the simulation models we apply a *distributed simulation architecture*. Several different distributed simulation architectures exist. One of them is the High Level Architecture (HLA), which is a standard architecture for modelling and simulation activities in the Department of Defense in the United States (Defense Modeling and Simulation Office 1996). Another distributed simulation architecture is the FAMAS Backbone (FAMAS MV2 Backbone Project, 2001), (Boer et al., 2002b). A comprehensive comparison between these two distributed simulation architectures can be found in (Boer et al., 2002c). For our purpose we chose the FAMAS Backbone Architecture because it is lightweight and it can be more easily interfaced with the simulation languages in which the already developed simulation models were written than HLA.

The FAMAS Simulation Backbone Architecture is represented by technical and functional components. Whereas the functional components represent the simulation models themselves, the technical components provide common tasks used by the functional components. The functional components can be simulation models, control programs or even real equipments.

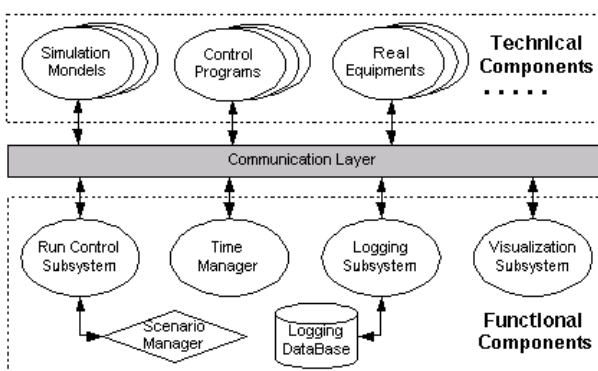


Figure 2: FAMAS Simulation Backbone Architecture

Figure 2 gives a clear picture of the separately defined functional and technical components. There are five well-defined subsystems, namely the Run Control Subsystem, the Backbone Time Manager Subsystem,

the Logging Subsystem and the Visualization Subsystem (Boer et al. 2002a), (Veeke et al. 2002). The functionalities of the technical subsystems are the followings: overall control of experiments, starting, stopping and periodically monitoring the simulation process (Run Control), synchronizing the simulation time among different simulation subsystems (Backbone Time Manager), collecting logging information from the distributed functional and technical components into a central database (Logging subsystem), providing separate or combined visualization views for the subsystems or the entire simulation (Visualization subsystem), completely defining a simulation run of a distributed model (Scenario Management).

Although most of the models use the FAMAS communication protocol for time synchronization and data exchange, other communication protocols are applied as well. The planning and scheduling tool communicates with the generator model through SOAP (Simple Object Access Protocol) by using the World Wide Web's Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange (Schmelzer et al., 2002). Figure 3 depicts the conceptual distributed model of the whole system including the interoperability between different models.

4 FEDERATES OF THE DISTRIBUTED MODEL

4.1 Truck Generator Model

The truck generation process is based on requests for picking up and/or delivering containers. The generation of the requests and trucks is accomplished by the Truck Generator model. The Truck Generator can be either a simulation model that automatically generates requests or can be a real time model controlled by users (Figure 3). Requests are generated in the form of timeslots that refer either to a *desired departure* time (when the truck starts driving to the port) or a *desired arrival* time (when the truck should arrive at the port). It is not guaranteed that the truck can start the driving process at the desired departure time. Due to the limited capacity of the container terminal (limited parking places, limited cranes, etc.) it can accept only a certain amount of trucks at a given time. As there can be more truck organisations that reserve the same timeslot for arrival time, the container terminal might not be able to handle some of them at the desired time as it cannot accept more trucks at the same time. In order to avoid congestion and time waste caused by waiting at the terminal for handling, a negotiation process is taking place between the truck companies and the port authorities. The negotiation process provides a *scheduled* time slot regarding the departure and arrival time of the truck at the container terminal.

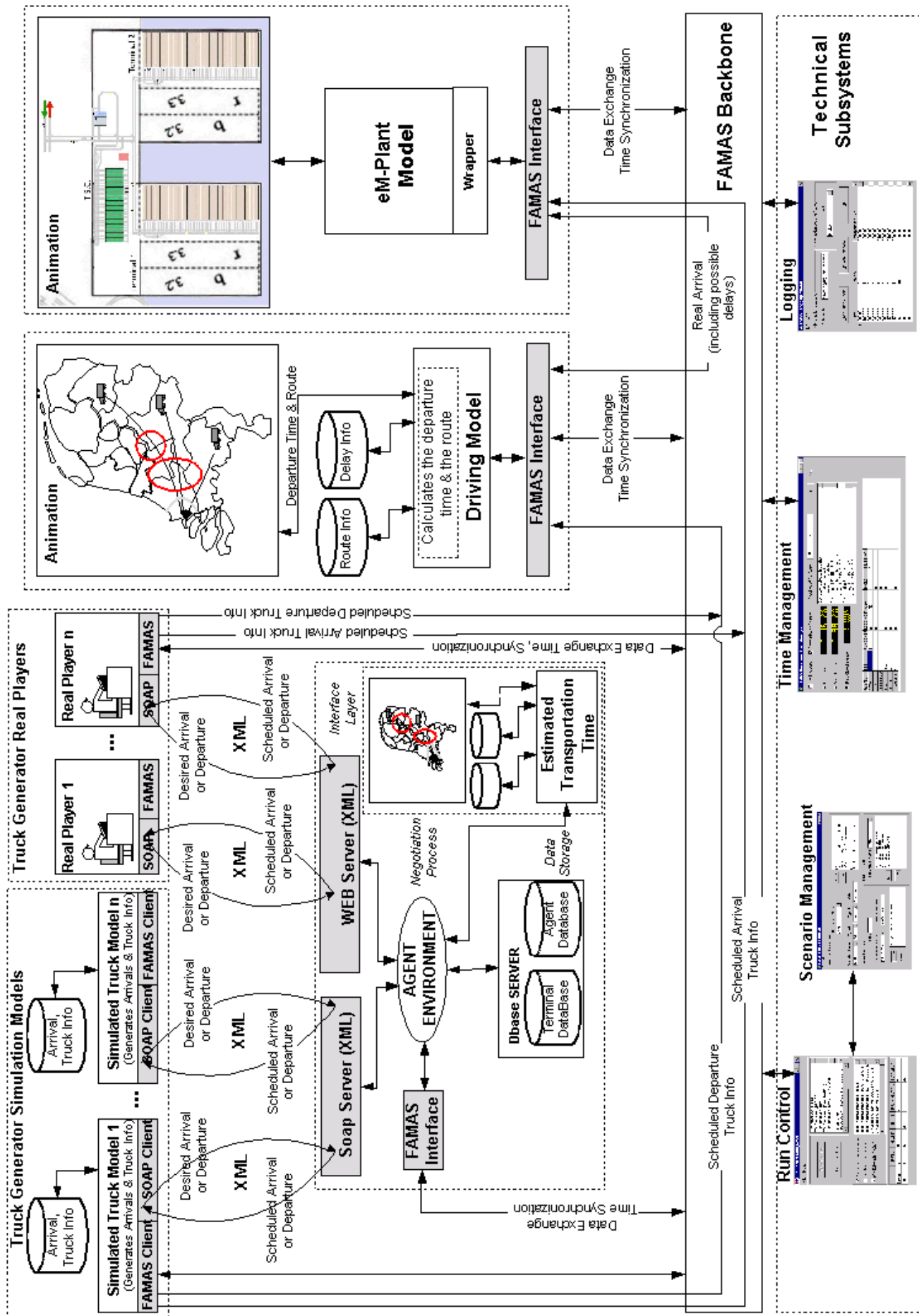


Figure 3. The Conceptual Distributed Model

The planning and scheduling model, which performs the negotiation, is using a model of the road traffic system in order to estimate the transportation time. This is needed in order to find out the approximate arriving time if the departure time is given or vice versa. Having the scheduled arrival and departure time, the scheduled departure time is provided to the road traffic model and the (approximated) scheduled arrival is provided to the container terminal model.

The simulation model of the truck generator stochastically generates the requests and creates the trucks based on earlier observed historical data (Figure 4). The Truck Generator uses special mechanisms in order to simulate the reservation of desired arrival or departure requested by truck companies. It allows the truck companies to make reservation for a timeslot for a certain time period in advance (in our case one week). The trucks that make a reservation for a time slot earlier get the desired time slot more easily compared to the companies that register late.

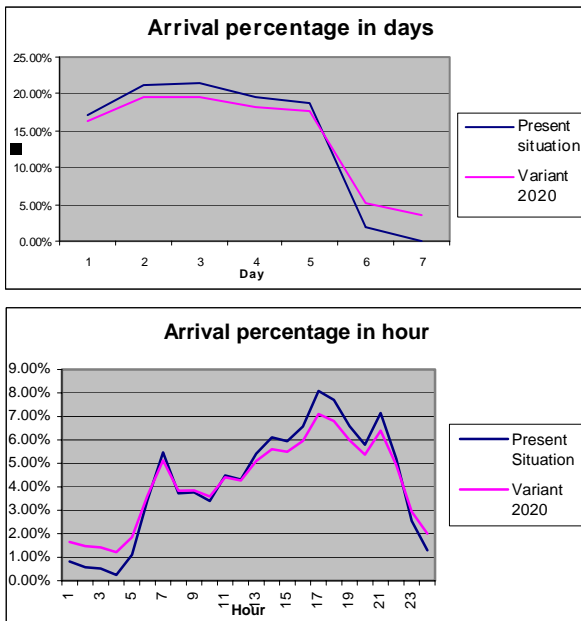


Figure 4. Arrival percentage in days and hour

We implement a general Truck Generator that can be instantiated in different ways for different truck companies. The company models use a configuration file with company specific data, such as nr. of trucks available, driving time, etc. Using several instances at the final test we can run several scenarios for different truck companies. We can analyse for example the different outcomes for earlier and later registered companies.

The truck generator simulation model is designed and developed in Java. The interoperability between the Truck Generator and the other models is achieved by the FAMAS Backbone and Simple Object Access Protocol (SOAP). The negotiation process between the Truck Generator and the Planning and Scheduling model is achieved through XML. Therefore the Truck Generator model is developed as a SOAP client for the planning and scheduling model.

4.2 Road Traffic Model

The aim of the road traffic model is to simulate the driving phase of the trucks from the companies to the port based on a given starting point and time provided by the truck generator, and external route and delay information provided by input data. Due to the fact that everything is modular in the distributed environment, we can use existing preserved models of the road traffic systems, although we need to solve the interfacing with other models. As we mentioned before we distinguish between micro and macro traffic simulation models depending on the level of detail at which they were developed. For a more accurate result, it is advisable to choose a highly detailed model, although it needs more input. We intend to carry out experiments with both micro and macro simulation models.

We use a road traffic model for two purposes. On the one hand one of its instances is used by the planning and scheduling model for estimation of the transportation time (here the animation is irrelevant), on the other hand another instance of this general model is used to simulate the driving of the trucks to the port. In this case the animation plays a crucial role, as it is indispensable for demonstration purposes. Both instances define the driving time stochastically, which highly depends on the exact day, hour and routes driven.

The input of the Road Traffic model contains the route and delay information stored in a database. Depending on the level of detail the external data might provide further information regarding the distance between two points (e.g. two intersections, two cities, etc.), the name of the road (e.g. A1), the maximum speed on that road, the earlier measured delays on this distance considering different days and hours, etc.

The Road Traffic model includes a demonstration by means of animation. The animation represents a map (e.g. map of Netherlands or even map of Europe) and visualizes the driving process of the trucks.

Data exchange and time synchronization with other models is solved through the FAMAS Backbone. The Truck Generator model provides to the road traffic model a scheduled departure time through this backbone. At that time a truck starts driving to the port. As we mentioned earlier this process is not deterministic, we can count on unexpected delays, or accidents, therefore the earlier provided scheduled arrival time to the port might be different to the real arrival. Although, the container terminal is informed about an approximately scheduled time when the truck arrives to the port, the driving model is responsible for sending a pre-arrival notice, which informs the real arrival of the truck. In this way the Automated Stacking Cranes (ASC) are able to start the preparing of the requested container for the truck (if it intends to pick up containers).

4.3 Container Terminal Model

The complex model of the container terminal already exists and can run without any planning and scheduling system. Furthermore, it has its own simple truck genera-

tor. However, we aim to improve the handling process of trucks at new container terminals by introducing a planning and scheduling system. Currently there are two independent container terminal models designed and developed for the Maasvlakte. The only aspect that they have in common is the simulation of the truck generator, which generates trucks according to the density per day and per hour as indicated in the description of section 4.1. Both of these concepts are designed and developed on a detailed level, which is essential to provide a realistic prediction.

4.3.1 Detailed Truck Handling Model at the Container Terminal

Regarding the truck processes at the container terminal we distinguish two concepts (van Til, 2003):

1. Compact Terminals (named as Concept 1)
2. Compact Terminals with a Central Gate and Truck Service Centre (named as Concept 5)

Concept 1: Compact Terminals

The schematic overview of the truck processes using this concept is depicted in figure 5.

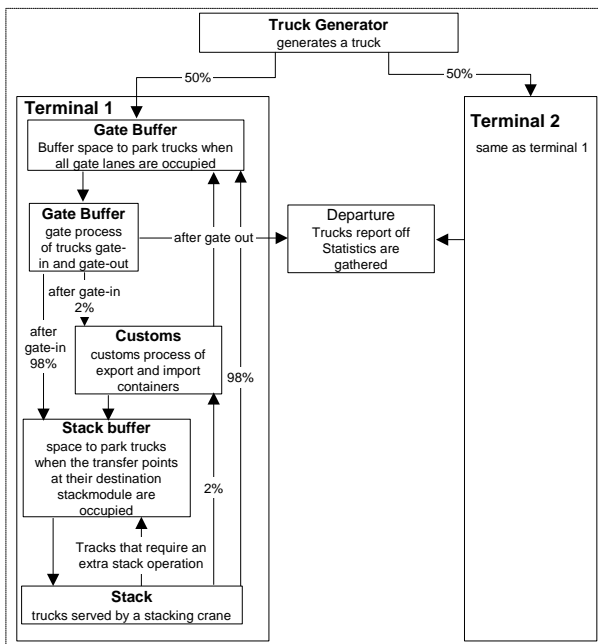


Figure 5. Schematic overview of concept 1

When a truck arrive at the terminal, the truck is assigned to the normal gate or to the Info Lack gate. The truck is positioned in the buffer in front of the correct gate. If one of the gate lanes is available, the truck continues immediately. After the treatment at the gate, the truck leaves towards the terminal. At customs, 2% of the trucks are being checked. These trucks are scanned after which they go to the substack of their destination. If no transfer points are available at the substack, the truck is positioned in the stack buffer, where it waits until one of the transfer points is available.

Trucks that are not checked at customs have passed the same routine before arriving at the substack. At the substack, the trucks are loaded or unloaded by the ASC. If the truck still has orders after the stack treatment, it will preferably stay in the same substack for the remaining orders. When this is not possible (e.g., because the truck has to pick up a container from another substack) the truck will go to another substack, if necessary, via the buffer. A favourable substack may be chosen for containers that are delivered by a truck. This is a substack where few orders are planned on the landside or a substack where the truck has to pass anyway to pick up a container. If all orders of the truck are carried out, the truck drives back to the gate to sign out. If necessary, the truck first takes place in the buffer prior to drive through the lane. After signing out the truck is removed from the model. All measured handling times are registered.

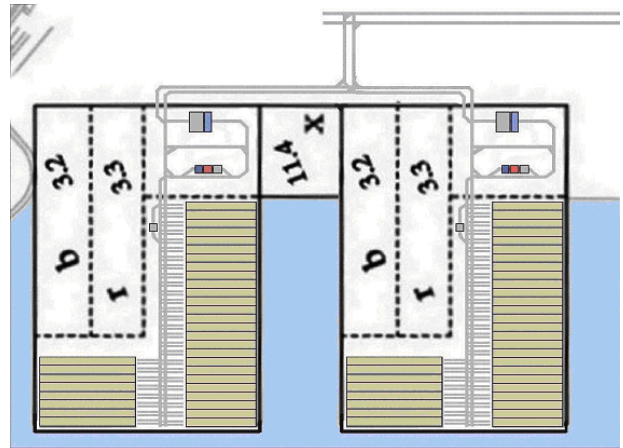


Figure 6. Layout terminals for concept 1

The simulation model of this concept is implemented in eM-Plant (eM-Plant official website, 2003), a commercially available simulation package. Figure 6 depicts the design of the compact terminal concept. The grey lines represent the roads and the grey blocks the buffers, placed in front of the trucks. The gate lanes are represented by the light blue colour, customs by the red and the sub stacks by the brown colour.

Concept 5: Compact Terminals with a Central Gate and a Truck Service Centre

The schematic overview of the truck processes using this concept is depicted in figure 7.

agent, this agent will implement enterprise specific strategies and will handle the message streams much faster than human operators will be able to (Leenaarts et al., 2003). Therefore an agent-based system has been selected for the timeslot negotiations at the container terminal. Requirements for this system were: Automated multi-channel communications, support for planning and scheduling, facilitating automated negotiations and eventually personalisable strategies.

In the agent-based system, every party (truck company or terminal operator) is represented by its own agent. A request with a desired arrival time from a truck company arrives at the system and is picked up by the representing truck operator agent. The request is passed on to the terminal operator agent, who will check availability of resources in the requested time slot. If there are sufficient resources in the requested slot, the terminal agent confirms the booking to the truck agent and updates the terminal database. If the terminal is fully booked at the requested time slot, the terminal agent will propose another slot, taking into account the bandwidth for negotiation that was sent with the request, the estimated transportation time based on the truck's departure information and information about the deep-sea vessel on which the container has to be loaded or from which the container has to be unloaded.

Because the generation of trucks with arrival times is completely automated during the simulation runs, there is no room for feedback from the truck company in case a requested timeslot is rejected. To show the functioning in a simulated "real" environment, a real player interface has been developed to submit time slot requests manually. Figure 10 below illustrates the real player applet interface.

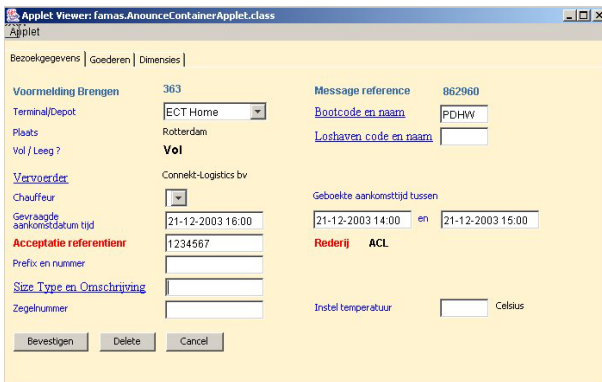


Figure 10. Real player applet interface

When a request for a specific timeslot has been rejected and a new timeslot has been proposed by the terminal, the truck company can accept the proposed time slot or try to get another slot which suits him better.

The system is based on the ILLYAN Agent Framework which is built in Java. It uses a SOAP Server and a Web Server for the communication with the other models and the interfaces. The messages are based on Extensible Markup Language (XML) for flexibility and interoperability.

The selected database management system (DBMS) is from the open source Firebird project. Because the standard JDBC 2.0 protocol is used to communicate with the databases, the system is effectively independent of the chosen DBMS. For visualisation purposes, a Java Swing GUI has been developed to display an overview of the requested and assigned timeslots. An example is shown below in Figure 11.

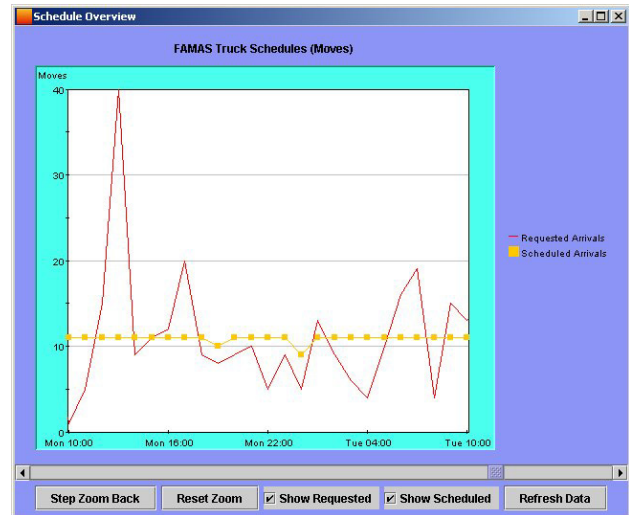


Figure 11. Visualization of assigned timeslots

5 EXPERIMENTS

Evaluation of the distributed models

Several tests have been carried out to test the proper working of the individual models and of their interfaces to the other federates. Using the DLL that was described in section 4.3.2, it was very easy to connect the detailed eM-Plant terminal models to the FAMAS backbone. The truck generator models and road traffic models have been written in Java. As the FAMAS backbone architecture is based on plain TCP socket communication, which is well supported in Java, interfacing these models to the backbone also took place without problems. The ILLYAN Agent framework for planning and scheduling the timeslots is also a Java application, and could therefore be easily included in the federation as well. All the technical subsystems of the FAMAS backbone (section 3 / Figure 2) were already available from earlier projects. Interfacing tests showed that all information exchange and synchronisation took place as indicated. Efficiency tests still have to be carried out.

Evaluation of the two container terminal concepts

Both terminal concepts discussed in section 4.3.1 are designed to serve 95% of the visiting trucks in 30 minutes. In concept 5 one standard stack module more is required to meet the performance requirements. On the other hand, an extra off-standard stack module is needed in concept 1. The fact that in concept 1 an extra stack module for off-standard containers is required, results in

a considerably lower efficiency of the present off-standard stack modules than in concept 5. Concept 1 requires a few extra gate lanes: two for the standard trucks and one for the Info Lack trucks.

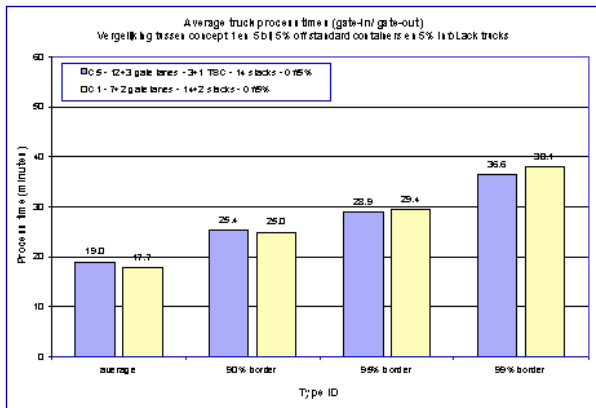


Figure 12. Comparison of handling times

Figure 12 shows a comparison between the handling times in concept 1 and 5. The average handling time of trucks in concept 1 is well over a minute shorter than in concept 5. The variation in handling times however, is smaller in concept 5. 99% of the trucks in concept 5 are treated within 36,6 minutes, whereas in concept 1, 99% is treated within 38,1 minutes. Concept 5 is more reliable concerning the handling times.

6 CONCLUSIONS

In this paper we introduce a distributed model that aims to improve the handling process of trucks at container terminals. In contrast to the already developed simulation models of the container terminal, which are monolithic, this approach integrates these models with other models to consider several additional aspects. In this sense it harmonizes the arrival time so as to be acceptable both for the container terminal and for the truck companies, and takes into account delays that might occur during the driving process of the truck to the terminal. Further it guarantees truck companies a reasonable maximum turn around time (e.g. 30 minutes 95% reliable).

This approach combines several independent models using a distributed modelling techniques. This entails a truck generator, a simulation model of the road system and the application of an agent based planning and scheduling model. The models of these systems are from different background disciplines and some of them already exist. In order to use these models together with the model of the container terminal we need to integrate them. In this paper we present a distributed environment that allows for the integration of these models. Although there is some collaborative work of the participants, this approach allows individual and parallel work, and the internal information in the models can remain hidden

for the other partners as they only share the relevant information for the interoperability process. This distributed environment can be extended and used by other participants as well.

Some final tests will be carried out to research the overall working of the federation, measuring the number of messages, the delays as a result of the distribution, and the efficiency of the model execution of the overall federated model.

REFERENCES

- Boer, C.A., Y.A. Saanen, H.P.M. Veeke, and A. Verbraeck. 2002a. "Final Report Simulation Backbone FAMAS MV2. Project 0.2 Technical Design." Research report to Connekt, 34 pages, Delft, The Netherlands. (March).
- Boer, C. A., A. Verbraeck, and H.P.M. Veeke. 2002b. "Distributed Simulation of Complex Systems: Application in Container Handling." In *Proceedings of the 2002 European Simulation Interoperability Workshop* (Harrow, Middlesex, UK, June 24-26). SISO, pp. 134-142.
- Boer, C. A., and A. Verbraeck. 2002c. "Connecting High Level Distributed Simulation Architectures: An Approach for FAMAS-HLA Bridge." In *Proceedings of the 14th European Simulation Symposium* (Dresden, Germany, October 23-26), SCS, pp. 398-405.
- Defense Modeling and Simulation Office. 1996. *HLA Specification*. Washington DC, USA. Available online via <https://www.dmsomil/public/transition/hla/>, [accessed June 27, 2002].
- de Hartog, A. H., et al. 2001. *International state-of-the-art in container logistics and performance requirements for mega hubs. A vision for container logistics in the port of Rotterdam*. Connekt, Delft, The Netherlands.
- eM-Plant official website. 2003. Technomatics Technology Ltd. Available online via www.emplant.de/simulation.html [accessed June 27, 2003].
- FAMAS MV2 Backbone Project. 2001. Research Program FAMAS Maasvlakte II Project 0.2 - Simulation Backbone. Delft, The Netherlands. Available online via <http://www.famas.tudelft.nl> [accessed September 24, 2002].
- Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. John Wiley & Sons, Inc., New York.
- Law A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3d. ed. McGraw-Hill, New York.
- Leenaarts M., and M. Kentrop. 2003. "Distributed Planning of container terminal resources with agent technology". Application for BNAIC 2003, Amsterdam, The Netherlands.
- Schmelzer, R., T. Vandersypen, J. Bloomberg, M. Siddalingaiah, S. Hunting and M. Qualls. 2002. *XML and Web Services Unleashed*. Sams.
- Straßburger, S. 2001. *Distributed Simulation Based on the High Level Architecture in Civilian Application Domains*. Ghent : Society for Computer Simulation International, Magdeburg, Germany.
- Taylor, J. E. S., S. Robinson, and J. Ladbrook. 2003. "Towards Collaborative Simulation Modelling: Improving Human-to-Human Interaction through Groupware" In *European Simulation Multiconference* (Nottingham, UK, June 9-10), SCS.
- Van Til, K. P. 2003. "FAMAS MV2: Simulatie Logistieke Performance", TBA Nederland (Februari).

Veeke H.P.M., Y.A. Saanen, W. Rengelink, A. Verbraeck. 2002. "Final Report Simulation Backbone FAMAS MV2. Project 0.2 Functional Design." Research report to Connekt, 20 pages, Delft, The Netherlands. (April).

AUTHOR BIOGRAPHIES

CSABA ATTILA BOER is a Ph.D. student at the Department of Computer Science of the Faculty of Economics at Erasmus University Rotterdam, The Netherlands. He received his M.Sc. degree in Computer Science at the Babes Bolyai University, Cluj Napoca, Romania. Since April 2001 he has been involved in the FAMAS MV2 Simulation Backbone project and since January 2003 in FAMAS MV2 Road Container Handling project. His research focuses on multi-level distributed simulation of complex systems. His email address is [<acboer@few.eur.nl>](mailto:acboer@few.eur.nl).

ALEXANDER VERBRAECK is an associate professor in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology, and part-time full research professor in supply chain management at the R.H. Smith School of Business of the University of Maryland. He is a specialist in discrete event simulation, both for real-time analysis and control of complex transportation systems and for modelling business systems. His current research focus is on the development of generic libraries

of distributed object oriented simulation building blocks. His email address is: [<a.verbraeck@tbm.tudelft.nl>](mailto:a.verbraeck@tbm.tudelft.nl).

ARJEN DE WAAL is a consultant at TBA Nederland, a company that specializes in simulation of complex logistic systems. Arjen graduated his study Operations Research and Management at the University of Amsterdam in 2000. He is an expert in discrete event simulation. His email address is [<arjen@tba.nl>](mailto:arjen@tba.nl).

BAS VAN ECK is a consultant for ILLYAN in Amsterdam, The Netherlands. He received his MSc in Chemical Engineering at the University of Amsterdam. After a career in process automation he co-founded ILLYAN and now focuses on e-business solutions in logistics. Since September 2001 he has been involved in the FAMAS MV2 Road Container Handling project. His email address is [<bvaneck@illyan.nl>](mailto:bvaneck@illyan.nl)

JERRY SEAGER is a consultant for ILLYAN in Amsterdam, The Netherlands. He received his BSc Upper Second Honours in Mathematics at Heriot-Watt University, Edinburgh, Scotland and his MSc in Advanced Software Engineering at Sheffield University, England. He joined ILLYAN in 2000 after a career in aircraft simulation in the UK. He has been involved in the FAMAS MV2 Road Container Handling project since January 2003. His email address is [<jseager@illyan.nl>](mailto:jseager@illyan.nl).