Simulation of an Intermodal Container Terminal to Assist the Management in the Decision Making Process

Andrea E. Rizzoli*, Luca Maria Gambardella*, Gianluca Bontempi‡
*IDSIA, Istituto Delle Molle di Studi sull'Intelligenza Artificiale
Lugano, Switzerland
{andrea, luca}@idsia.ch
‡IRIDIA, Université Libre de Bruxelles
Bruxelles, Belgium
gbonte@iridia.ulb.ac.be

Abstract A decision support system for the management of an intermodal container terminal is presented. Among the problems to be solved, there are the spatial allocation of containers on the terminal yard, the allocation of resources and the scheduling of operations in order to maximise a performance function based on some economic indicators. These problems can be solved either using techniques such as job-shop scheduling and genetic algorithms or simply by the experienced terminal manager, using his/her experience. The manager can trust computer generated techniques only by validating them using a simulation model of the terminal. The policy test bed is a discrete event simulation model, based on the process-oriented simulation paradigm. The case study of the Contship La Spezia Container Terminal, located in the Mediterranean Sea in Italy, is examined.

1. INTRODUCTION

The management of an intermodal container terminal is a complex process which involves a vast number of decisions to be taken. Most of the goods which are traded daily over the world are transported via intermodal terminals. Goods arrive and leave on various transportation means such as air cargoes, trucks, trains and vessels. Containers are a convenient and standard way to package and transport finished goods. An intermodal container terminal plays a fundamental role in routing goods to and from their origins and destinations. It is a basic node in a transportation network, where thousand of daily decisions are taken to manage this sustained flow of containers.

The advent of management information services and data processing greatly improved the ability of terminal managers to control the whole process, but yet raw data has to be analysed and treated to provide some insight on the performance of terminal operations. Simulation models have proven to be a reliable and convenient tool to support the decision makers in the daily operations in many cases (Hayuth *et al.* 1994, Blümel, 1997, Bruzzone and Signorile, 1997). They provide a test-bed to assess the validity of management policies and can be used to point out problems such as conflicts in resource allocation and terminal space management. These simulation tools do not provide answers to question such as "how can I minimise the time it takes to unload these two incoming ships?" or "Should I unload the ship, or wait for the train to arrive?". In many cases, these answers are yet to be provided by the terminal managers, basing their decisions on experience in solving these problems.

A substantial help to terminal managers can arrive from Decision Support Systems (DSSs) where planning and management techniques, derived from the Operations Research and Artificial Intelligence fields, can be coupled with simulation models and statistical data analysis tools. The role of simulation becomes of paramount importance in such a setting: human decision makers tend not to trust computer generated management policies, unless they either fully understand the way they were generated or are provided with sufficient evidence of their validity. This behaviour is often proven to be reasonable, since very often computer generated policies are not flexible enough in comparison to the complexity and high stochasticity of real world operations.

A well designed simulation tool can be the middle ground where decision makers compare their own experience with the DSS generated management policies and validate them.

In section 2 of this paper we introduce the problem specifications with reference to the real world case of the La Spezia Container Terminal, operated by Contship SpA. In section 3 we define the requirements of a DSS to assist the Management in the Decision Making Process, describing the optimisation algorithms and the simulation model with its assumptions. Finally, we draw our conclusions and indicate future developments of this work.

2. THE PROBLEM

An intermodal container terminal is a place where containers enter and leave by multiple means of transport, as trucks, trains, air cargoes and vessels (I/O transport means). We focus our attention on the case study of La Spezia Container Terminal (LSCT), located in the Tyrrhenian sea in Italy.

Containers arrive at LSCT by train, vessel or truck and are stored in the terminal yard. Containers then leave the terminal by the same means to reach their next destinations. The flow of containers is composed of an *import flow*, i.e. containers unloaded from ships, to be either transshipped or directed to the final destinations by trucks and trains, and an *export flow*, i.e. containers loaded on ships leaving the terminal.

In the LSCT, containers are stacked up to the fifth level on the yard by rail-mounted cranes (*yard cranes*) which unload trucks and trains. This stack height is quite unusual and is due to the lack of space on the yard. LSCT is a terminal with a high traffic on a small yard and therefore the management of space is a critical issue. *Quay cranes* unload vessels and place containers on *shuttle trucks* which move them to storage locations in the yard. Loading a vessel is a similar process, where the shuttle receives the container from the yard cranes and moves it to the proper quay.

The amount of work processed by a container terminal depends on the quantity of containers in transit.

3. THE DECISION SUPPORT SYSTEM

Storing containers on the yard, *allocating resources* in the terminal, and *scheduling vessel loading and unloading operations* (L/U operations, for brevity) are major problems in an intermodal container terminal. To solve these problems we define an architecture composed of three different but strictly connected modules (see Figure 1):

- a simulation model of the terminal, described in terms of entities (work force, transport means, storage areas, etc.) and processes (vessel load/unload, shuttle truck movements, crane operations, etc.);
- a set of forecasting models to analyse historical data and to predict future events (Box *et al.*, 1994; Vemuri and Rogers, 1993), thus providing estimates of the expected import and export flows;
- a planning system to optimise L/U operations, resource allocation, and container locations on the yard.

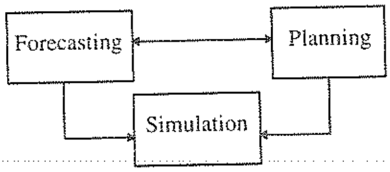

Figure 1. The modular system architecture.

This architecture supports the decision makers in the evaluation of:

- vessels loading and unloading sequences in terms of time and costs;
- resource allocations procedures;
- policies for container storage both in terms of space and cost of operations.

This allows terminal managers to assess "what-if" scenarios; for instance, what happens if the terminal undergoes an increased input/output throughput, or even if structural changes are made (e.g.: new berths are built, new cranes are added).

As the forecasting module is described in our previous papers (Gambardella *et al.*, 1996, Bontempi *et al.*, 1997), in the following sections we introduce the other two modules of our architecture: the planner and the terminal simulator. For each topic, we present the major problems, the resolution methodologies and the experimental results obtained at the current state of the project.

3.1 The Optimisation Modules

In our study, we identified a series of problems, placed at different representation levels, which can be assisted by a computerised decision support system: spatial allocation of container locations in the terminal yard, allocation of terminal resources (yard and quay cranes, work force, etc.), and scheduling of terminal operations (e.g. container movements) in order to maximise a performance function of economical indicators. These problems also have different planning horizons related to the speed of the dynamics of the system they control: the spatial allocation policy has a horizon of about one week, while a few work shifts (about twentyfour hours) is the horizon of the resource allocation policy. The planning horizon of scheduling of terminal operations can be as short as one hour.

Scheduling of terminal operations

This short-term optimisation is aimed at solving the scheduling of daily L/U operations. We decomposed this problem in two steps:

- *off-line scheduling*, based on the assumption of a complete a priori knowledge of the terminal state. This implies knowing at each instant, with certainty, the availability of cranes, the level of occupancy of the yard, the date of arrivals of the vessels. Unfortunately, there is no guarantee that this information is correct and there is the risk that an off-line scheduling is not robust enough to be effective in reality. Then a reactive adaptation of the scheduling is necessary;
- *reactive scheduling* (Kerr and Szelke, 1995), whose task is real time supervision of the L/U procedure. Its initial condition is provided by the previous level optimisation (off-line scheduling) and then it is adapted to unexpected events.

Resource allocation

The role of this optimisation is to solve the problem of resource allocation for vessel loading and unloading operations. Resource allocation spans a time horizon limited to a few shifts of the work force. The problem of resource allocation can be formulated as a stochastic dynamic programming problem (Bertsekas, 1995), with the goal of maximizing the profits over a limited time horizon. The objective function depends on the costs of resource usage, the lateness in vessel loading/unloading and the income of the terminal for each operation. Unfortunately, due to the high number of variables, an approach based on stochastic optimal control is not realistic. Then, in order to maintain a high dimensional representation of the problem, we limited attention to a deterministic model, where nominal costs and profits are considered. We employed a Genetic Algorithm (Goldberg, 1989) approach to search for a near-optimal resource allocation solution. The program we implemented accepts as inputs the list of scheduled ships, their estimated time of arrival, the forecast number of containers to be loaded and unloaded and the yard regions involved in loading and unloading operations. The outcome is a schedule of the yard and quay cranes to employ in the upcoming work shifts together with a deterministic forecast of expected profits.

Spatial allocation

The objective of this optimisation module is to find one or more efficient container storage policies. Our approach consists of two steps: first, we solve a job-shop scheduling problem to determine what yard configuration (arrangements of containers on the yard) optimises L/U operations; then, we define as an optimal policy the more realistic one which minimises the deviation from that configuration. Finally, different allocations are ranked according to a quantitative performance index, which measures how effective the L/U policy would be (in time and/or economic terms) if we started from that configuration.

The outcome of the optimisation module is then a set of management policies which we are assessing either confronting them with the practice of experienced terminal managers or with our simulation model. Simulation allows us to describe the terminal operations at a much finer level and to keep into consideration those stochastic factors which we were obliged to neglect in the optimization phase. This combination of optimisation and simulation, enables us to produce heuristic solutions in reasonable computation times and allow the terminal manager to compare his/her decisions with the ones generated by our algorithms. Some of these modules are currently in the testing phase at LSCT.

3.2 The Simulation Module

3.2.1 The design

The architecture of the simulation module is based on the partition of simulation objects between *simulation agents* and *simulation components*. In an intermodal terminal there are two parallel flows: information and containers; the simulation agents use the flow of information to take decisions on how to direct the container flow.

We founded the design of the simulation module on the object-oriented analysis and design paradigm (Booch, 1994), we modelled simulation agents and components as objects which store and exchange information on terminal inputs, states and outputs and which perform actions according to their local behaviour. There is no unique supervising agent which controls the whole simulation, but the simulation is the result of the interaction of single agents, each one endowed with "local" knowledge on its actions in response to the behaviour of other agents (Zeigler,1984 and 1990).

There is a hierarchy of simulation objects according to their "intelligence" (see Figure 2). Planners, such as yard and ship planners are at the top, since they take the informed decisions on resource and space allocations we were concerned of in the optimisation section (see 3.1). Crane operators (yard and quay) and shuttle truck drivers, occupy the middle layer since they have the local knowledge which allows them to perform container movements, avoiding local conflicts and inconsitencies, such as two yard cranes competing to place containers in the same yard area. At the bottom, there are the terminal components, such as yard areas and the containers and other agents such as ships, trains and trucks, which in principle are "intelligent" but that were modelled as "dumb" since their behaviour is imposed as an external constraint and not directly controllable by the terminal operator.

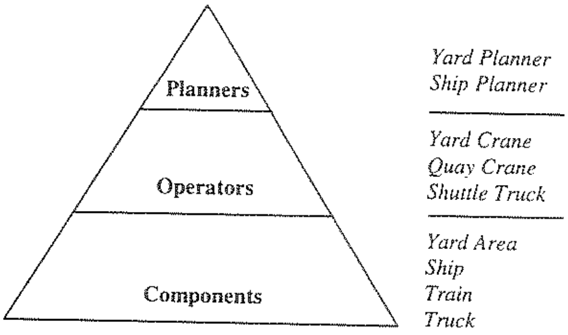

Figure 2. The hierarchy of simulation agents.

3.2.2 The model

The simulation model tries to replicate the terminal activities and it is based on the principle that external events generate responses by the simulation agents which in turn operate on simulation components. The responses of simulation agents are determined according to the policies which can either be generated by the optimisation modules or by a representation of the experience of terminal operators.

External events are: trucks arriving at terminal gate; ships arriving at terminal pier; trains arriving at terminal. The arrival generator is a part of the simulation module which generates these arrivals. Ship and train arrivals are read from a database, since they are known in advance, while truck arrivals are generated according to statistical distributions.

When a ship, train or truck enters the terminal, it has a list of containers (or just one, in the truck's case) which is imported and a list of containers to be exported. These lists are used by the yard and ship planners.

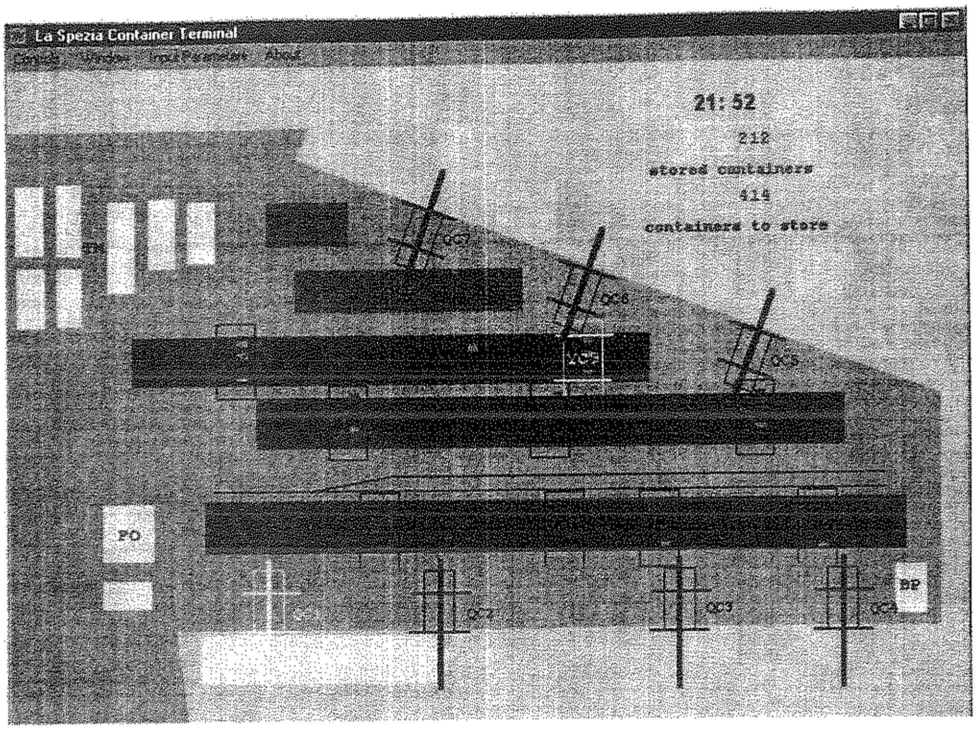

Figure 3. The graphical user interface of the simulation module representing the terminal layout.

The ship planner is a simulation agent dedicated to organise the loading and unloading operations of a ship. The ship planner performs the following tasks:

1. Allocate the quay cranes work shifts needed to load and unload the ship, given the ship import and export list. This task can be performed either using the resource allocation optimisation module or by entering the resource allocation strategy decided by the human operator.

2. Compute the bay plan. Resource allocation assigns to each quay crane a set of bays to work on. In general, unloading occurs before loading, and these two activities must respect the ship structural stability constraints (Sha, 1985), these constraints result in the work order of the bays.

3. Ask the yard planner to assign destinations on the yard to the containers to be unloaded. These containers are unloaded in order as stowed. The unloaded containers will be stored in subregions of the yard areas, named *import areas*. The size and location of import areas is a decision variable.

4. Communicates to the yard planner the containers to be loaded. This list is ordered by a set of constraints which imposes a sequence to be respected in stowing containers aboard according to their size, weight, port of destination, and to a series of distinctive characteristics such as hazard class, kind of good transported, etc.

5. Put the quay cranes to work according to the plan previously determined. Supervise loading and unloading operations, collect statistics and evaluate performance.

The lists of import and export containers (see items 3 and 4 in the previous numbered list) are used by the yard planner to build the schedule solving the job-shop problem associated with yard crane operations.

The task of the yard planner simulation agent is to organise the container allocation on the yard in order to maximise yard crane performance, avoid crane deadlocks (when two cranes try to work on the same yard area), and minimise the time to access containers during storage and retrieval. In detail, its tasks are as follows:

1. Allocate the yard cranes work shifts, given the list of containers to be loaded and unloaded by all the ships and trains that are present or are due to arrive.

2. Organise the yard space according to a given policy, selected among one of those assessed with the optimisation module (*automatic parking*).

3. Solve the job-shop scheduling problem, using the available data on trains, trucks and ships to be loaded and unloaded. The result is the work list for each yard crane (the ordered list of containers to be moved). These work lists are computed using the reactive scheduling algorithm implemented in the optimisation module. We will show later how this centralised policy can be replaced by a distributed policy generated by local rules used by crane agents.

Besides this high level management performed by the ship and the yard planners, there are the local management decisions taken by "less intelligent" simulation agents such as cranes and shuttle trucks.

Quay cranes start to work when the ship planner assigns them a list of containers to be loaded and unloaded. They stop working when they have finished to process their lists. Quay cranes move containers to and from shuttle trucks which run between the quays and the yard cranes. When the quay crane unloads a container, it asks the yard planner which yard crane is assigned to it, the truck will therefore travel to the yard area where that yard crane is working.

Yard cranes pick up and put down containers on the yard. They have a queue of jobs to be performed. A job is a container movement, either picking it up from a truck and placing it on the yard or vice versa, and even temporary moves to unpack stacked containers are jobs. As we have seen before, this queue of jobs (the work list) can be automatically optimised by a job-shop scheduling, or can be managed by local rules which try to emulate the behaviour of the human operator. Yard cranes are also provided with tie-breaking mechanisms to avoid deadlock: it can happen that, given the randomness associated with the time a crane moves a containers, the job queues push the cranes towards a conflict, such as trying to move two containers which are stored in the same bay in the same time. The yard cranes can acknowledge this potential deadlock and reassign one of the container moves to contiguous crane (this is a sub optimal solution, but avoids computing again the whole job-shop problem).

3.2.3 The implementation

The simulation module has been implemented using Modsim III, a commercial simulation language which supports the process oriented simulation paradigm, and the object-oriented programming paradigm.

Modsim III allows the programmer to define multiple threads of execution defining class methods which can be asynchronous or synchronous. Simulation time flows only in calls to synchronous methods. Different methods calls are possible to synchronise processes, such as *waitfor* calls, which suspend execution until the called routine returns, and *tell* calls, which spawn a process and then continue in their execution trail.

Modsim III also allows to call external C/C++ routines, which is useful to integrate the optimisation modules and to integrate the database interface.

The database is implemented using Microsoft Access and is queried via the ODBC interface. It contains the data describing the terminal characteristics (yard areas, yard cranes), the input data used to generate the incoming trains and ships and the transported containers, the initial state of the terminal (the containers and their position on the yard), and can be used to store the simulation results (crane performance, waiting times, loading and unloading times, etc.).

4. EXPERIMENTING WITH SIMULATIONS

In Figure 3 we report a typical screen-shot of the terminal during a simulation. A ship is moored on the west pier (north is to the left of the picture) and it is being unloaded by two quay cranes *QC1* and *QC2* (only *QC1* is active, though). Containers are to be positioned on yard areas *CA*, *CB*, and *CC*. On these yard areas the yard cranes *YC* from 1 to 9 are working.

Before launching a simulation, which can be controlled using the interface we have just described, the decision maker can set up terminal parameters, such as the location of yard areas in the terminal, their storage capacities, their use as import and export areas, the initial state of the yard, and the sequence of arrivals of trains and ships. Moreover, the decision maker can tune simulation parameters which govern the stochastic distributions used to characterise the time it takes a crane to move a container, to move from a bay to another one, and so on.

Once the set up of simulation parameters is completed, the decision maker can start the simulation and observe either the on-line graphical results and the bi-dimensional animation, which highlights moving cranes, conflicts, container stacking heights. For instance, in Figure 4 a series of histograms which are updated on-line is reported. Each histogram is associated with a yard crane and it reports the number of containers which have waited from the moment they joined the queue of truck waiting to be unloaded to the moment they were placed on the yard.

Simulation can be run in "quasi-real time" (one simulation second corresponds to one real minute) to observe the details of the animation, but can be greatly accelerated (one second corresponds to one or more hours) to collect reports and statistics.

Using this interface the decision maker will be able to collect results that can be used to compare different management policies.

5. CONCLUSIONS

An intermodal terminal is a complex dynamic system characterised by an high level of uncertainty and it is non-stationary. The terminal management is constantly trying to improve the overall performance and to improve the co-ordination among the various decision makers (the ship planners and the yard planners) to eliminate conflicts and increase the efficiency of the used resources. For these purpose, optimisation algorithms and methods can be successfully employed to support the decision makers in their daily operations. In this framework a simulation tool plays a fundamental role to verify and validate the applicability of the computer generated solutions in comparison to the experience of the management, especially given that computer optimisation is performed using an approximate model of the terminal. Moreover, a simulation tool can be used to assess the terminal performance and to identify conflicts in resource usage and critical decision paths. The first results of this study are being validated in LSCT, while an application of the system to the Gioia Tauro container terminal, the major terminal in the Mediterranean Sea, is under study.