INVITED

Graphical Representation Of Computer Simulation Outputs Of Port Operations As A Verification And Debugging Tool

Lal C. Wadhwa
and
Dominique L. Morel
Department of Civil and Systems Engineering
James Cook University, Townsville, Queensland 4811, Australia

Abstract Computer simulations have now been used for over twenty years to model port operations. They have been used as planning tools for port operations as well as for designing ports. Graphical representation of simulation output has been used to more easily and effectively convey the results to port managers. The focus of this paper is on demonstrating that the graphical representation of simulation outputs is an effective and powerful tool for debugging and validation of discrete event computer model simulations such as that of port operations. The bar chart representation of movements and/or activities of machinery and ships displaying the important event times can show at a glance scheduling errors that may be hard to detect in conventional debugging techniques. The advantage of using bar charts representation of outputs for ship movements at a port is illustrated with examples from simulation of a real-world shipping terminal. Tracing movements and events in a tabular form can show many scheduling errors, but close and meticulous examination of the data is required and some errors can easily be overlooked. We postulate that there are distinct situations in which it is not easy to validate certain attributes or parameters using tabular output. Graphical representation of the same data can make obvious those errors that are easy to overlook when the data in tabular form. As a result of the graphical representation of output data from our study, few errors that had eluded detection from the tabular output became apparent. We discuss the software used to produce these charts and its merits and demerits. We also explore other software approaches that may be feasible, and endeavour to characterise situations where graphical representation of output of discrete event simulations is highly recommended.

1. INTRODUCTION

Computer simulations have been used as planning tool for port operations as well as tool for designing ports Wadhwa [1992]. Graphical representation of simulation output has been used to more easily convey the results to readers, Hayuth et al. [1994]. The Department of Civil and Systems Engineering at James Cook University has been involved in developing port simulation models since 1977, Kenyon and Wadhwa [1978]. Recently, a simulation model was developed to ascertain the capacity of a shipping terminal under different configurations. A number of simulations have been carried out with this model over the past six years to assist the port authorities in achieving increased efficiency and throughput. In a recent request, the port management asked the university to run the model at the annual throughput expected for the coming year and to produce a trace of ship movements. The port management wanted to use the results to plan staffing levels for the following year. Bar charts showing major events were produced to represent the ship movements more easily, readily and effectively. These charts were directly used in human resource planning. The variety of applications of this model by the port management is a testimony to the usefulness of simulation and is the subject of another paper by the authors [1995].

The port in this study is a single commodity bulk export terminal. The port has two berths and one shiploader. The channel from the open sea to the port is 13.03 metres deep. It is a tidal port with maximum predicted spring tide of over 7 metres. Ships ranging in size from a DWT of 20,000 tonnes to just over 200,000 tonnes call at this port.

The annual throughput is currently at 26.5 million tonnes per year.

2. SHIP MOVEMENTS AT THE PORT

For the purposes of identification, ships are sequentially numbered on arrival to the port. The arrival pattern is random. The simulation uses the historical arrival of ships for many of its inputs, such as cargo type, quantity, inter-arrival time, minimum deballasting time, summer draft, immersion, etc. From this and other data such as tides, weather, maintenance schedules, loading rates, loading delays, and shiploader travel time, the model calculates the occurrence of various events during ship's stay in the port.

These include permission to berth, permission to load, completion of loading, and the time of sailing of a ship.

The capacity of the shiploader and the yard machinery is quite high. Thus many smaller ships experience deballasting delays before completing their loading. Ships requiring deep drafts may also experience tidal delays during loading. If the duration of the tidal delay or deballasting delay is predicted to be at least 2.5 hours, the shiploader is moved to load the ship at the other berth. It is brought back to complete loading the first ship as soon as the delay is finished. Ships with a loaded draft greater than channel depth plus the tide at high water minus the required under-keel clearance may experience a tidal delay before sailing. Ships that are draft restricted have a sailing window. This is the period in which the tide is high enough so that the ship can sail without the risk of grounding.

3. MODEL OUTPUTS AND VERIFICATION

3.1 Model Outputs

The model produces an annualised summary of the results, including throughput, number of ships loaded, average values of: shipment size, turnaround and queuing times, queue lengths, net and gross loading rates, berth and shiploader utilisation, shiploader travelling time, delays due to weather, tide, deballasting, industrial stoppages, and maintenance, as well as reporting the surplus capacity and various other relevant statistics. This summary, which is produced by the model includes formatting code for *troff* typesetting program so that the output can readily be presented to the clients. The format of the output was prescribed by the port management to conform to the existing performance reports produced by the central computer facility of the shipping terminal.

3.2 Checking for Correctness of Model Outputs

The model simulation also produces a file of outputs to check the correctness of the model. This file contains many values that have known ranges and can therefore be used to verify that these are valid whenever the program is modified, or the input data is changed. The program also produces a binary file of the important times and parameters for each ship that is processed by the model. These are

ship number,
berth number,
cargo loaded,
cargo not loaded due to draft restrictions,
departure draft,
ship dead-weight,
whether the ship departure was restricted by tide,
arrival at anchor,
pilot on board berthing,
all secure at berth,
permission to commence loading,
commencement of loading,
loading interrupted
  to complete ship at other berth,
  because of deballasting delay,
  because of draft restrictions,
loading recommenced
  after completing loading ship at other berth,
  after tidal or deballasting delay,
tidal delay before sailing,
sailing time,
start of sailing window,
end of sailing window, and
pilot disembarks ship.

With this extensive record of each ship, various traces can be produced to verify that operations were modelled correctly. The traces are produced by separate programs that reads the ship records and produces *troff/tbl* formatting code. This code can either be viewed by a previewing program on an X-terminal or printed.

3.3 Recurring Need for Debugging

The model was first developed in 1989 and has been consistently modified and refined over the last six years as the requirements of the port management have changed. Some of these changes have been minor, but many have involved significant developmental effort. As is usually the case with programming, most changes have caused some bugs. A significant proportion of time spent in debugging has been in identifying the programming errors. Often these "bugs" were not errors as much as the result of conditions arising that were not envisaged. The program was originally developed to simulate a single berth, single shiploader, and single rail receival station scenario, and has developed into a model that can simulate two berths, two rail receival stations and two shiploaders. Furthermore, the shiploaders are not dedicated to individual berths and one ship can be loaded with one or two shiploaders. The input data has also changed over the model development period. These changes to date reflect the changes in ship size distribution, ship arrival patterns, and cargo requirements. Many "bugs" that arose came out of situations that did not occur with data collected from earlier periods of the operations of the terminal. Some of these cases caused much time to be spent modifying the model to handle the new situations which resulted from changes in operations, throughput, or ship size distribution, reflected in the new data.

3.4 Output Formats

The simulation model produced several summaries of output results. These included port performance indicators, berth statistics, shiploader and other machinery utilisation, stockpile statistics etc. The detailed movement of ships through the port showing each and every event and activity of each ship (for about 400 ships) was also produced in the form of tabulated trace to assist the managers in planning of operations and staffing. In general, the port managers found these reports quite useful. However they found the tabulated trace somewhat tedious to examine in great detail. Graphical representation of outputs was, therefore, introduced so as to more easily convey trace output to the port management. It was decided to produce bar charts of events of major interest to management that were simulated by the model. Since the simulation program was running on a UNIX platform and *troff* was already being used, it was decided to use *troff/pic* to draw the graphics. A "C" program was written to read the file of records produced by the simulation program and to generate *troff/pic* code. It was decided to have a separate program to generate the graphical output in the same way as the tabulated traces.

4. GRAPHICAL OUTPUT AS A VERIFICATION AND DEBUGGING TOOL

4.1 Graphical Representation Of Output Shows "Bug"

Along with the request for graphical representation of the model output, the port management also provided more recent data on port operations. The program for graphical representation of simulation output was developed to provide bar charts with the desired layout and format.

The graphical output obtained with the use of new data showed that one ship was experiencing a deballasting delay of about 20 hours and a loading time of less than one hour. This was, obviously, not correct. Upon further investigation it was found that the ship was in this port for a top-up load. The ship had loaded all but 5,000 tonnes at another port. The ship did not require the same deballasting time as a ship which would require to be loaded to its full capacity, the latter being a normal case. Such unusual cases had not been considered in our model. Conventional testing and validation techniques used did not make this obvious, but once noticed with graphical representation of the output, it was found to exist in the tabulated output. These exceptional cases require an estimate of the deballasting time required for partly-loaded ships. Unfortunately, this information was not available. However, this example has highlighted the need for additional data or different approach for treating these unusual cases.

4.2 Other Cases

Another recent simulation involved the determination of the impact of any restriction on tug availability on port operations. The objective was to rationalise the use of tugs at two adjacent terminals. Earlier studies ignored tug availability, as tugs were not considered to be a constraint in so far as to affect any ship movements at the terminal or cause any delay to the berthing or sailing of ships. However, with the expansions of berthing facilities and the desire of rationalisation of operations at the two terminals, tug availability was now to be introduced as a constraint.

The simulation program was modified to incorporate the tug constraints and expanded to handle four berths and four shiploaders (the combined facilities at two terminals). The simulation program was subjected to the usual testing and calibrations.

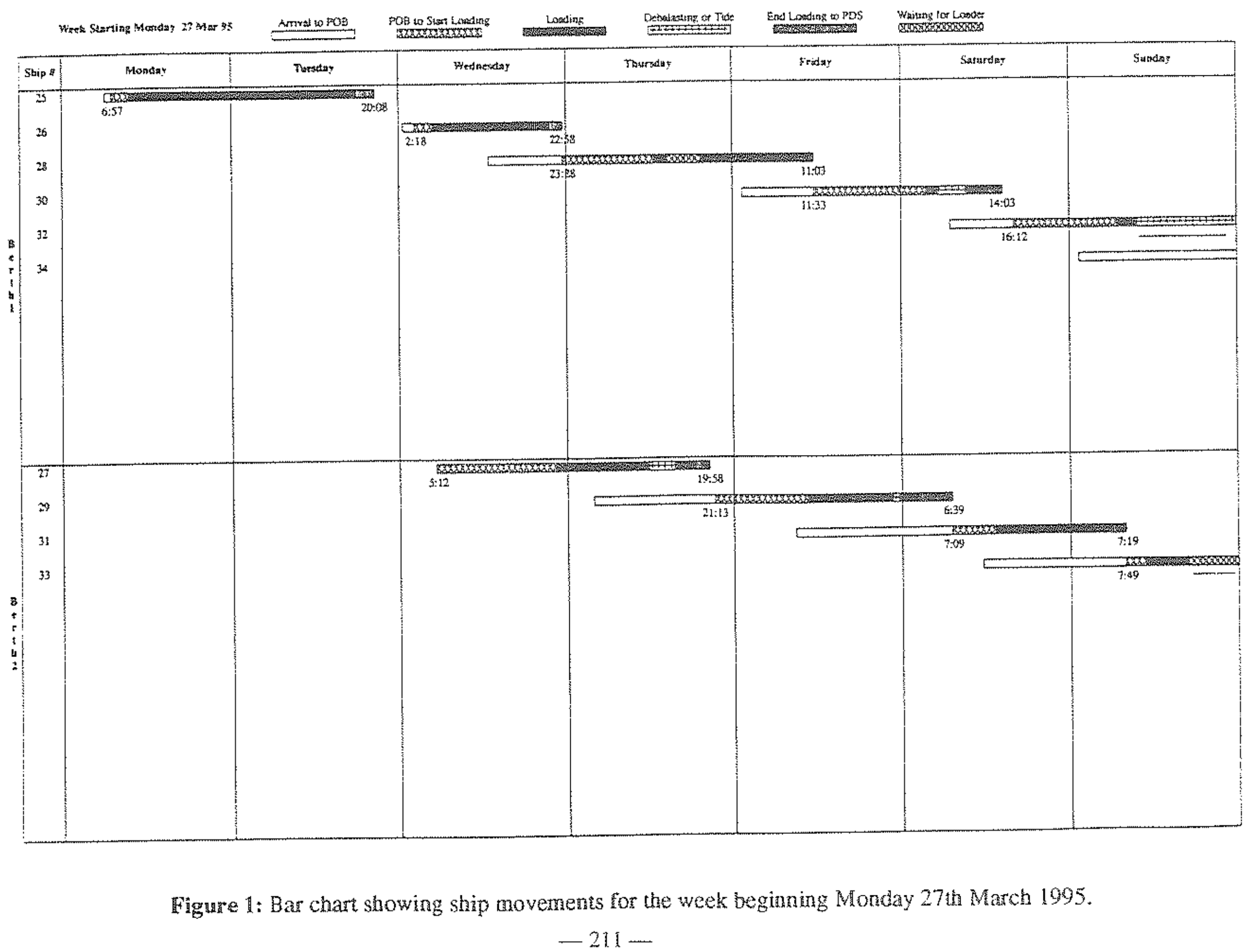

Figure 1: Bar chart showing ship movements for the week beginning Monday 27th March 1995.

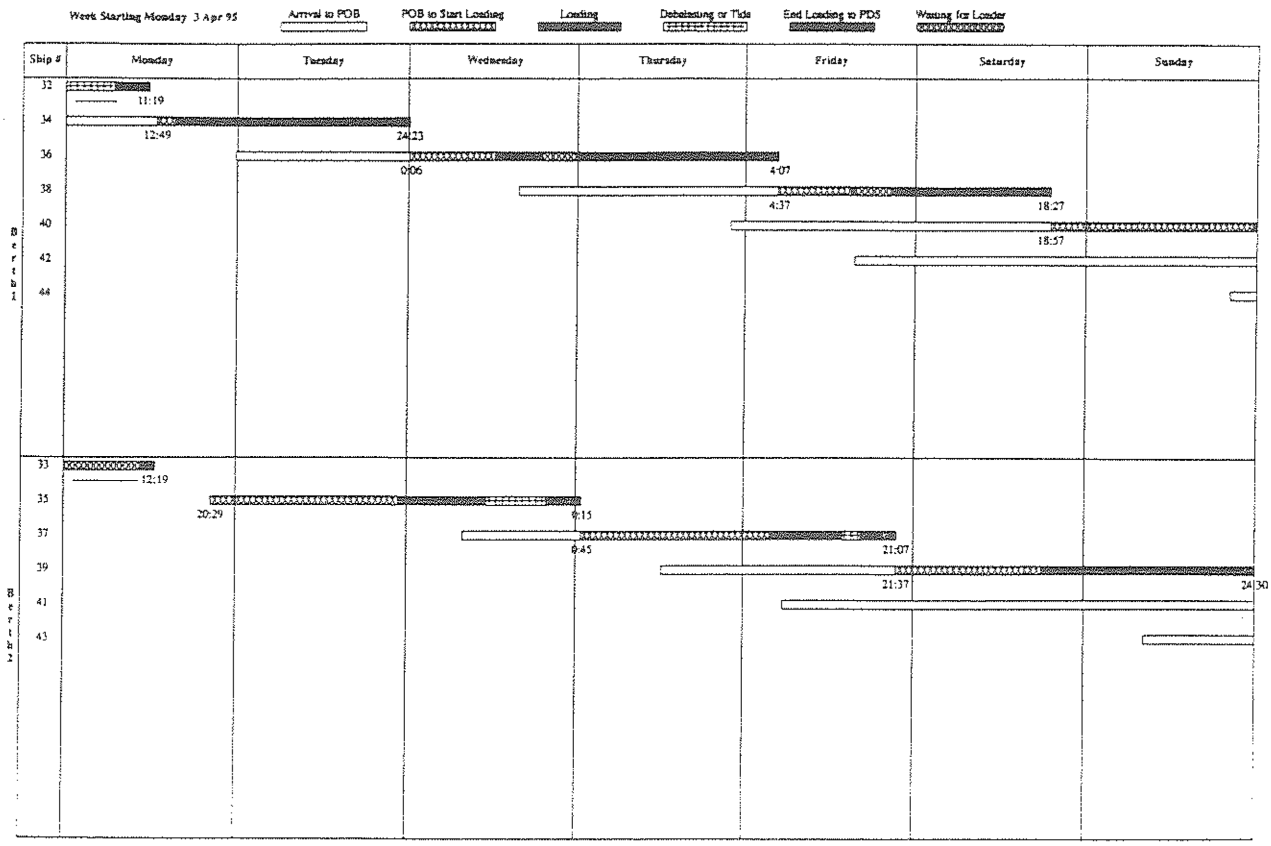

Figure 2: Bar chart showing ship movements for the week beginning Monday 3rd April 1995.

	Berth One	Berth Two
Ship Number	32	33
Tide Delay Before Berthing	1.65	0.00
Cargo Loaded	25000	27000
Arrival Anchor	Sat 01-04-95 06.59	Sat 01-04-95 11.39
Pilot on Board	Sat 01-04-95 16.12	Sun 02-04-95 07.49
All secure	Sat 01-04-95 17.26	Sun 02-04-95 09.04
Permission to Load	Sat 01-04-95 18.42	Sun 02-04-95 10.19
Loading Commenced	Sun 02-04-95 06.49	Sun 02-04-95 10.37
Loading Ceased To Finish Other Ship		Sun 02-04-95 16.54
Loading Recommenced		Mon 03-04-95 10.49
Loading Ceased Debalasting	Sun 02-04-95 09.37	
Loading Ceased Tidal Del		
Loading Recommenced	Mon 03-04-95 06.49	
Loading Completed	Mon 03-04-95 09.49	Mon 03-04-95 10.49
Tide Delay Before Sailing		
Start Window		
Sailed	Mon 03-04-95 11.19	Mon 03-04-95 12.19
Stop Window		

Figure 3: Times and details of ships numbers 32 and 33.

A new bar chart layout was prepared for the four berth model. The results of simulation were represented on the new bar charts. When the graphical output was examined, a few bugs in the model, that had escaped detection, became apparent in the tug scheduling algorithm. One example that the graphical representation of the output showed was that if three ships sailed half an hour apart from each other (according to current practice), then two new ships would berth at the same time. This should not occur as ships required tugs for an hour for berthing. Each of these errors occurred only one or twice a year with the input data used, but could have occurred more often with different ship arrival patterns.

As the errors occurred only a couple of times a year, they would have been hard to detect by examining the tabular trace of the ship movements and would certainly not be reflected in the summary of output results presented to the port management.

Another study simulated the more recent port operational data with the existing configuration comprising of two berths and one shiploader travelling between both berths. The objective of this study was to plan staffing levels at higher throughput that was expected in the coming year.
Upon examination of the graphical output, an error was detected in the model. This example revealed how conspicuous the error appears in the graphical output compared to the tabular output. As can be seen from Figures 1 and 2, ship number 32 is experiencing a deballasting delay, and ship 33 is shown to be waiting for the shiploader. This situation should not occur, as the shiploader is evidently available to load ship number 33. What was actually happening is that both ships were experiencing deballasting delays. This was, obviously, a programming error which was immaterial in most cases but became relevant in case where two ships incur deballasting delay simultaneously. Moreover, as can be seen in Figure 2, as soon as ship number 33 finishes waiting for the shiploader, the loading is shown to be completed. This is obviously an error. Upon examination of the code it was found that the case of two small ships loading and experiencing long deballasting delays was never envisaged in the model. This was easily fixed by the addition of five lines of code.

Figure 3 shows the data for the same two ships, numbers 32 and 33, in tabular form. A careful examination shows that there are errors with the scheduling of the two ships. However, the errors are not apparent, especially if taken in the context of six ships tabled per page, 367 ships processed per year, and the simulation run over numerous years. Examination of each ship in detail becomes a daunting task and the errors can easily be overlooked. In this particular case, the time interval between loading recommenced and loading completed equals zero which, according to the port's operational rules has to be three hours. This could have been checked and the error could have been detected. The problem is in deciding which variables from the vast number are checked and which ones are not. In hindsight it is easy to see what should have been done.

5. DIGRESSION

The work involved in producing the charts used in these studies was extensive, and the effort required to make changes to the charts was also not trivial. On the other hand, thorough examination of the trace of ship movements in tabular form is also a time consuming process, and some errors may still escape detection.

We postulate that if the trace of the output does not have many variables to examine, and that the trace is fairly short then it would probably be quicker to examine the data in tabular form. Whereas if there are large volumes of data and numerous variables to each item in the trace that need to be examined, then it would probably be advantageous to spend the time to present the results in bar chart form or some other graphical form. Since the data in chart form made the results of the simulation more understandable to port management, the use of graphical representation is strongly recommended. Graphical representation acts as a powerful debugging tool and displays the results of the simulation very effectively.

6. SOFTWARE FOR DRAWING CHARTS

The software used in these studies to produce the bar charts is effective, but is not very flexible. Each time the layout of the chart requires to be changed, it involves modifying the program that produces the *troff/pic* code. This can involve a few days in modifying "C" code. It also means that the modeller has to be familiar with *troff/pic/tbl* as well as "C". Another problem with this style of code is that if the programmer is no longer available, the new programmer has to learn the code and decipher how the previous programmer carried out the task. Good program documentation can make this task easier, but this is often not the case. A further restriction is that these tools are only available on the UNIX systems.
The advantage of this approach is that it is very effective. Almost any format is possible, and once set up correctly, numerous runs can be made. Processing can be achieved with little fuss using a shell script to systematically process the output, and prepare the charts.

The tabulated output can also be processed and prepared using a shell script making it easy to include both types of representation into reports.

7. SOFTWARE APPROACH

Today much modelling is done on PC's as is evident from the availability of simulation packages such as **ARENA** for the PC. We have made attempts to acquire a software package to produce graphical representation of simulation outputs trace that is more user friendly than *troff/pic*, but none appear to exist. Programs such as *Microsoft project*, and *Time Line* were examined, but were found not to be suitable. It is believed that if a package that could be used to easily produce charts as in Figures 1 and 2 was available, then bar charts could easily be used for verification and debugging purposes.

To make the package user friendly it would need to be in the WYSIWYG format of most modern word processors and graphing packages. It is envisaged that the data be entered in table format, i.e. one row could represent one ship or other entity being modelled, and that each column be an event duration. If an event was not to occur for a ship then it would have a duration of zero. Table 1, is an example of the input table envisaged. It is also conceive that the software would be able to read the data from file in the more common data formats, i.e., comma delimited text, popular spreadsheet formats as well as some database formats.

The user would be required to specify the background template of the output chart by using a drawing facility similar to either *xfig, MacDraw,* or the picture facility in *Microsoft Word.* The system would have to have provision for specifying the units of time used, and also the start time and/or date. The hypothetical example in Table 1, would uses hours as the units of duration. The arrival time would, in this case, be hours from the start of the simulation. Each column would be given a legend number from which the software would build a legend. The text associated with the legend would have to be able to be edited, and as shown in Table 1, the fill would be the same for columns with the same legend number. In the legend, the user would specify the fill characteristics and/or colour for each legend number by choosing from a pop up menu.

Table 1: Envisaged Input of User-friendly Software.

Legend			1	2	3	4	4	3	5	6
Ship Number	Berth Number	Arrival	Arrival to POB	POB to Start loading	Loading 1	Tidal Delay	Deballasting Delay	Loading 2	Waiting for Loader	End loading to Pilot Disembark
1	1	0	0	3	20	0	0	0	0	3.5
2	2	19	0	5	9	0	5	3	0	3.5
3	1	26	0.5	6.5	3	0	0	14	5	3.5
4	2	41	3.5	10.5	22	0	0	0	0	5.2

8. ACKNOWLEDGEMENTS

We would like to thank the port authority that funded this work and supplied the data and much input into the verification and testing of the models. Our thanks must especially go to the Manager, Transport Co-ordination and Customer Services at the port.

9. REFERENCES

Hayuth Y., Pollatschek M.A., Roll Y., Building A Port Simulator. *Simulation*, 63(3), 179-189, 1994.

Kenyon J. A. and Wadhwa L. C., Analysis and Simulation of Townsville Port Operations. *Simulation Modelling Techniques and Applications, Proceedings of SIMSIG - 78:* Simulation Conference, Australian National University, Canberra 4-8 September 1978.

Wadhwa L. C., Planning Operations of Bulk Loading Terminals by Simulation. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 118(3), 300-315, 1992.

Wadhwa L. C. and Morel D. L., Decision-Making in Port Operations Using Simulation *MODSIM-95*