Modelling a Grain Handling System

James E. Everett, Department of Information Management and Marketing
The University of Western Australia, Nedlands, Western Australia 6009
e-mail: jeverett@ecel.uwa.edu.au

Abstract Grain produced by farmers in Western Australia is delivered to receival points, from which it is transferred to coastal ports by rail or road, and then shipped to overseas markets. Grain deliveries are concentrated in a limited season. The system is subject to a range of constraints, including grain availability, storage capacities, transport capacities and market requirements. Further complications arise because a number of different grain segregations have to be stored, transported and shipped concurrently. This paper describes two models that have been designed, using Extend™, to aid in decision making for the grain handling system. The first model covers the whole system, from receival point, through coastal ports and onto market-bound ships. The second is of a grain receival point, and models at a micro level the individual truck arrivals and rail or road dispatch of multiple grain types. Both models have dynamic graphical interfaces, that can be used to portray the system operation during a simulation run. The graphical interfaces are hierarchical, enabling the user to view the overall system operation or to concentrate upon the detailed behaviour of some part of the system. The models can be used for planning the grain handling system, investigating the effects of proposed modifications to the system, adjusting to unforeseen situations during real-time operation, and as a means of communication between individual stakeholders in the system.

1. INTRODUCTION

This paper outlines the function and structure of two Extend™ simulation models designed to aid the planning and evaluation of the grain handling system from country receival point through to shipping the grain out from the port. Both models deal with multiple grain segregations (or types of grain) simultaneously. The maximum number of grain segregations can be increased if required.

The first model covers the entire system, from receival nodes, which are country receival points where grain is delivered by the producing farmers, down to one or more ports by narrow gauge or standard gauge rail or by road transport. The grain is then shipped out, either on arrival at the ports or according to a pre-set shipping schedule for each port. The time interval for the simulation is one day. The transport and storage costs are accumulated, and the tonnages handled at any point of the system can be reported. The model is of potential use in exploring the effects of changing the routing from receival points to alternative ports, or through alternative modes of transport (narrow or standard gauge rail, or road transport). It can also be used to explore the ability of the grain handling system to cope with increased volumes of grain, or with the division of the grain handled into an increased numbers of segregations.

The second model treats a single receival node in much greater detail. This model uses a time interval of one minute. For each segregation the grain arrives in trucks, whose Poisson arrival rate can be modulated to vary on a daily cycle, with secular variation over the longer term, so that daily operation over a season can be separately modelled for each segregation. This feature is useful, since the different segregations may have different but overlapping patterns of arrival at the receival node. Trucks carrying the various grain segregations form queues for a single unloader. The unloader has to be cleaned, with a delay, every time the grain segregation changes. Each grain segregation is stored first in a limited main storage and then any surplus is stored in an unlimited overflow storage. At preset intervals, a train of prescribed capacity arrives, and its wagons are filled with the stored grain segregations according to a nominated priority. Again for this model, costs incurred and tonnages handled can be monitored as desired. The major potential use of this receival-point model is for detailed exploration of the effects of changing the tonnages of various grain segregations arriving at a receival node, and of changing the transport schedule from that node.

The simulation models comprise interconnected blocks. Some blocks are in the standard libraries supplied with Extend™, other more complicated ones have been purpose written. The blocks are written in the language ModL, which is a dialect of the computer language "C". Each block is coded as a set of subroutines, controlling the setting up of the block, initialisation at the beginning of a simulation, behaviour at each step of the simulation, and finishing off at the end of the simulation. Some blocks are hierarchical, being composed in turn of a set of more elemental blocks.

The design and component blocks of the Grain System Model and of the Receival Point Model will be described in the sections that follow.

2. GRAIN SYSTEM MODEL

The main window for the Grain System Model is shown in Figure 1, with the simulation order of each block highlighted, being numbered in order from block 1 through to block 7. Each block of the model will be discussed, being referred to by its simulation order number.

In operation, the model uses a simulation interval of one day for computations. Grain of various segregations are received at specified rates at the receival nodes, flow down to the ports using allocated transport, and are then shipped out to overseas customers.

The first three blocks enable control parameters to be entered; allocate available transport according to preset and dynamic priorities, and report accumulated costs of the system.

The fourth block is hierarchical, and can be opened up to display a group of up to ten receival nodes, all feeding grain down to the same port. This hierarchical block can be duplicated to cater for any number of receival nodes, up to ten for each hierarchical Group Node block.

The fifth block represents a port from which the grain is shipped out. Multiple ports can be modelled by using duplicate port blocks. Grain shipped from the ports feeds into the sixth block, representing the world market. To this block, as to any point in the system a "tonnage report" block can be attached to report accumulated tonnages of the various grain segregations shipped.

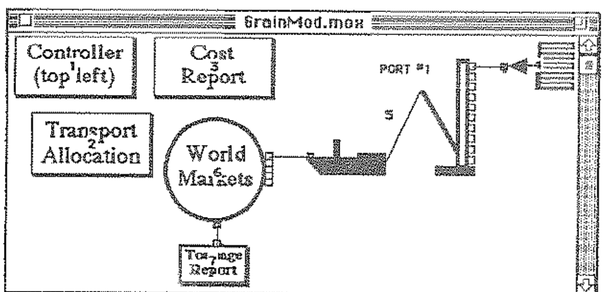

Figure 1: The Grain System Model

The model requires a text file of data (named "Grain.txt") to specify grain receivals at the nodes. Grain can be shipped out according to a preset schedule of shipments. If this option is chosen, then the shipping schedule for each port must be input through another text file of data for each port. The name of each shipping schedule file contains a single digit, identifying the port to which it refers. For example, the shipping schedule file for Port #1 would be named "Shiport1.txt".

2.1. The Controller Block

Figure 2 shows the controller block opened up (by double clicking) to reveal a simple control panel, enabling the receival node data to be revised by reading from the grain data file, Grain.txt. Revision is achieved by checking the "Refresh" box and closing the control panel, automatically causing the Grain.txt file to be read and its data used to update the information for each of the receival nodes.

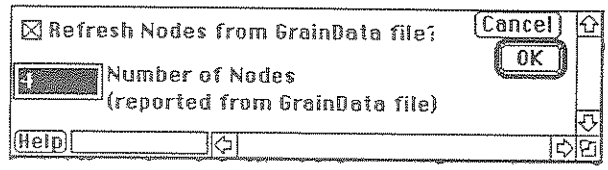

Figure 2: Dialogue Window of the Controller Block

The file Grain.txt is a text file, which can be created and edited using a spreadsheet package such as Excel. One record or line specifies parameters for each receival node. The 34 parameter fields are separated by tabs.

The first eight fields specify the receival node location, its destination port, its tonnage capacity, and cost parameters for operating (per day), handling and storage (per tonne).

The next eight fields specify transport parameters. The transport type determines whether grain is carried from the receival node to the port by standard or narrow gauge rail (0, 1) or by road (2) transport. Also specified is the distance and speed to port, the time spent loading, the transport capacity per trip, and the transport cost parameters.

Finally there are six sets of three parameters, one set for each grain segregation, specifying the total tonnage, starting day and receival rate (in tonnes per day) for that grain segregation arriving at the receival node.

2.2. Transport Allocation

The second simulation block, shown in Figure 3, allows specification of the total system transport capacity for each of the three transport methods. These tonnages are dynamically allocated to operate transport of grain along the receival node to shipping port routes. At any step in the simulation the unallocated transport tonnage is reported.

The six grain segregations can be given priorities in the allocation of transport, and the model allows for these priorities to be set in up to three ways. If there is a shipping schedule, then transport allocation is based upon the shipping schedule needs. The grain segregation or type priority is given next importance. Finally the receival node priority is taken into account.

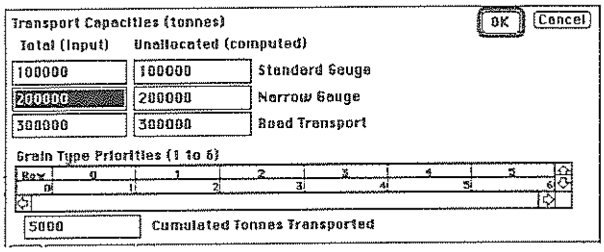

Figure 3: The Transport Allocation Block

2.3. Cost Report

As shown in Figure 4, the cost report block can be opened up at any stage in the simulation to see the cumulated site costs and transport costs. These costs are reported for each receival node and in aggregate for the whole system.

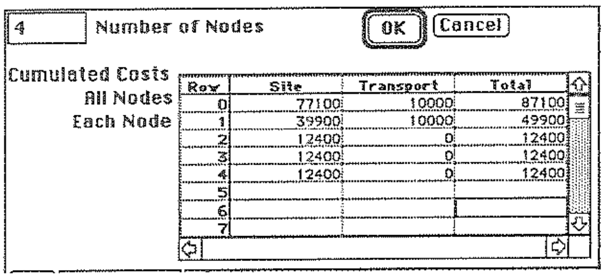

Figure 4: The Cost Report Block

2.4. Group Node

Receival nodes are clustered in hierarchical groups, to make the model less cluttered. Double clicking on the Group Node block opens up the dialogue showing the contained nodes, as shown in Figure 5.

Each hierarchical block can contain up to ten receival node blocks, which join together to feed into a single port. The block is automatically labelled with node and transport information by closing the dialogue.

Each receival node within the Group Node block can be opened by double clicking, to give a dialogue like the one shown in Figure 6. The node number is entered directly to the dialogue. All other data are updated from the "Grain.txt" file whenever the dialogue is closed, provided the "refresh data" box has been checked in the Controller block. If this is the case, the parameters for the receival node are then read from the corresponding single line of the "Grain.txt" file. Other figures on the dialogue are computed and reported as the simulation proceeds.

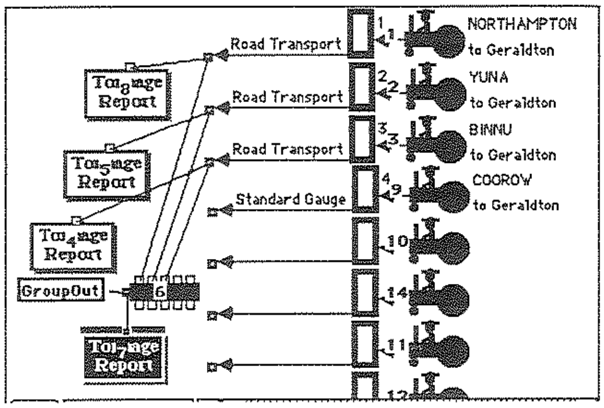

Figure 5: The Hierarchical Group Node Block

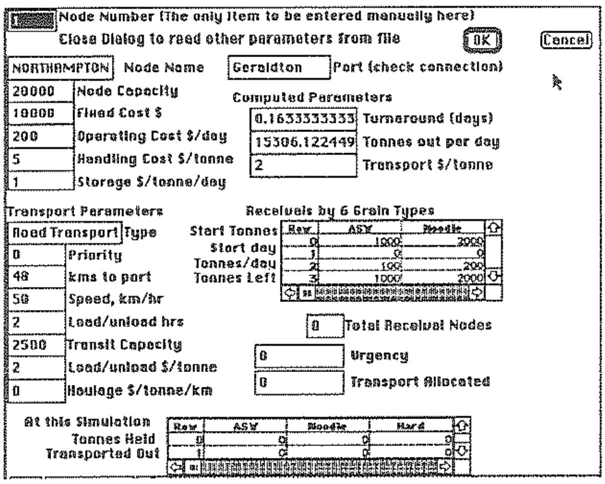

Figure 6: Dialogue Window of the Node Block

2.5. The Port Block

Although the model depicted has only a single port, further port blocks can be added and numbered sequentially. The port block dialogue opens as shown in Figure 7.

By default, grain is shipped out immediately it arrives at the port.

Alternatively, shipments can be controlled by a shipping schedule, and grain segregations are stored at the port until required by a scheduled shipment. If a shipping schedule is being used, each port requires a text file appropriately numbered. For example, Port #1 requires a text file named

"Shiport1.txt". In this file, each scheduled ship in sequence is specified by a single line, showing the day of departure and the required tonnages of each of the six segregations.

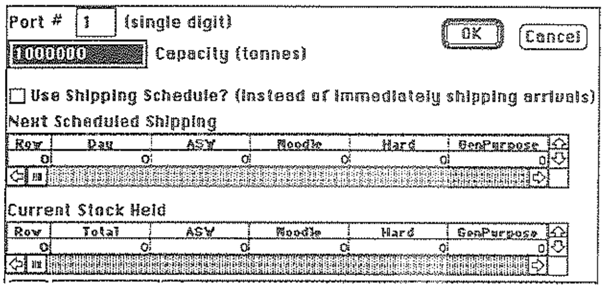

Figure 7: Dialogue Window of the Port Block

During simulation, the next scheduled ship specification is automatically loaded into the port dialogue, and is used to control the priority of grain segregations being brought to the port. If insufficient grain is available at the port when a shipment is made, this shortfall is flagged, a warning posted, and the simulation then proceeds.

After each ship time arrives, data are read in for the next scheduled ship and the process is repeated.

2.6. World Markets

The "World Market" block accumulates total tonnages of the six segregations as they are shipped out from the ports.

2.7. Tonnage Report

The standard plotting routines supplied in the Plotter Library of Extend™ report four channels. The "Tonnage Report" block, whose structure is shown in Figure 8, has been put together as a simple hierarchical block enabling tonnages for all six segregations to be reported. This block can be duplicated and attached to any output point of the model where it is desired to monitor tonnages. An example of this use occurred in the Group Node hierarchical block.

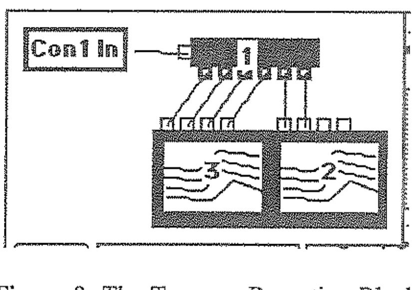

Figure 8: The Tonnage Reporting Block

3. RECEIVAL POINT MODEL

Unlike the Grain System Model, the Receival Point Model is designed to simulate in some detail the operation of just a single receival node. Up to five grain segregations or types are catered for, but the model could be easily modified for a greater number of segregations.

The Receival Point Model makes more use of the elemental building blocks supplied in the Extend™ libraries, and so makes less use of purpose-written ModL coding than does the Grain System Model. In order to achieve the desired operations, the Receival Point Model uses hierarchical blocks incorporating multiple elemental building blocks.

The Receival Point Model does not call upon any external data files to control its operation. All parameters are entered directly into the block dialogues. The model window is shown in Figure 9.

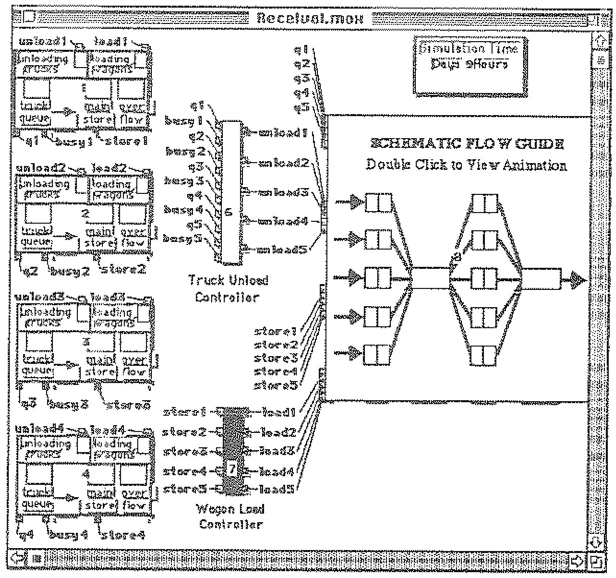

Figure 9: The Receival Point Model

The blocks are again numbered in the order in which they are computed at each step in the simulation. The first five blocks are hierarchical, each one modelling the arrival and storage of a single grain segregation.

Blocks 6 and 7 control the flow of grain. Block 6 supervises the operation of the single shared truck unloading bay. Block 7 loads up the wagons of a train for each scheduled departure time.

Block 8 summarises the other blocks, to give an animated view of the operations as the simulation progresses. Block 9 is just a clock, reporting the elapsed days and hours.

3.1...3.5. Segregation

The hierarchical segregation block is mainly made up of standard blocks from the provided Extend™ libraries. Its structure is shown in Figure 10.

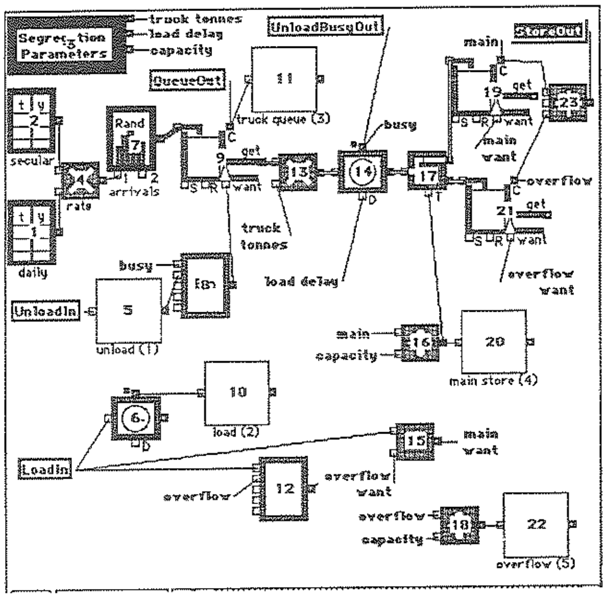

Figure 10: A Hierarchical Segregation Block

The daily (1) and secular (2) schedules control the Poisson arrivals of trucks, so that the arrival rate can change during the season, but with a daily operational schedule. Both these blocks are Input Data blocks from the Extend™ Generic Library (Diamond & Hoffman, pp.404-405). They enable the arrival rate to change over the simulation period.

Their dialogues can be opened by double clicking as in Figure 11. Multiplying the daily by the secular rate allows a seasonal variation modulated by a daily operational cycle.

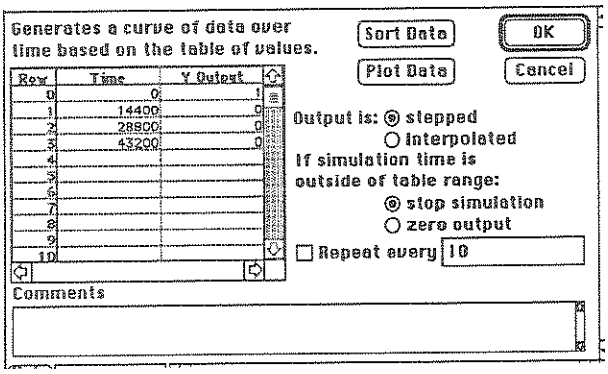

Figure 11: The Input Data Library Block Dialogue

On arrival, the trucks to be unloaded (9, 13, 14) queue for the single truck unloader bay, which is shared between all five segregations. Unloading takes a specified time, with some extra time whenever the segregation using the unloader changes.

Once grain has been unloaded, it is stored. Storage is preferably in the cheaper main storage (19), but if the main storage is full the grain goes to an overflow storage of unlimited capacity (21). When a train is preparing for departure, required amounts of grain are removed from the storage.

The plain blocks (5, 10, 11, 20 and 22) control animations of the simulation as it proceeds.

3.6. Truck Unload Controller

The Truck Unload Controller of Figure 12 receives information from all five segregations as to their trucks waiting to be unloaded. Whenever the Unloader falls vacant, it is assigned to whichever segregation has the longest queue. The cleaning time (7) causes appropriate delay (9) when the unloader is reassigned to a different segregation.

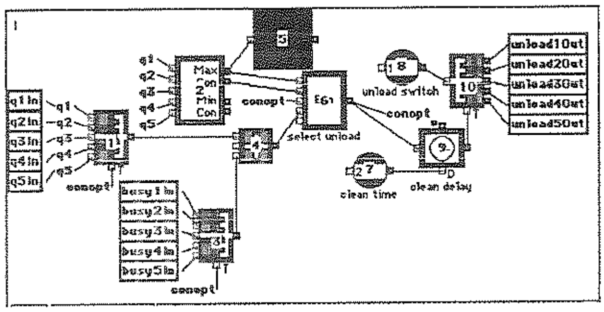

Figure 12: The Hierarchical Truck Unload Block

3.7. Rail Wagon Load Controller

The Rail Wagon Load Controller of Figure 13 loads the wagons of a train, according to the train schedule. This train schedule is controlled by specification of the first train time (6), the train interval (7) and the load duration (8). The train composition is specified by the number of wagons (16) and the tonnes per wagon (17).

Each wagon holds only a single grain segregation. Schedules of desired shipments are specified (1 through 5). These, combined with the storage quantities, are used to control the quantities of each segregation loaded.

Each time the train leaves, a whistle blows.

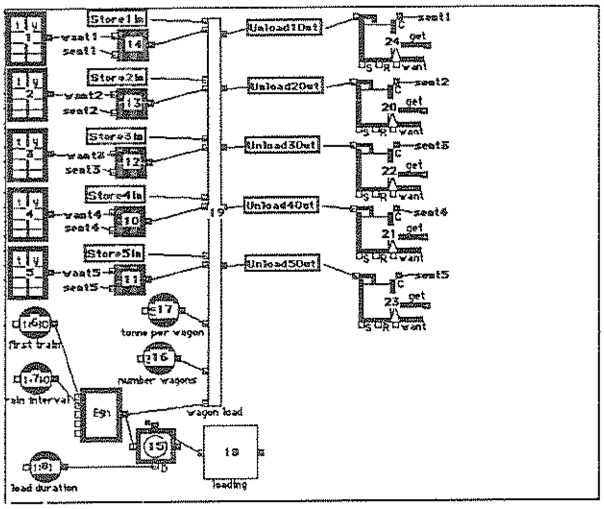

Figure 13: The Hierarchical Wagon Load Block

3.8. Schematic Flow Guide

The Schematic Flow Guide (Figure 14) gives an animated report summarising the operation of the system during simulation.

As the simulation progresses, the Flow Guide shows for each segregation whether trucks are waiting, and whether that segregation is currently being unloaded.

Animations are also used to show the storage status, and when the train is being loaded.

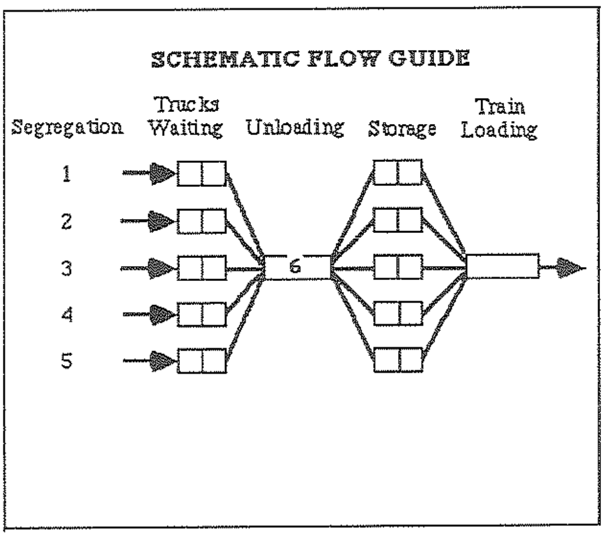

Figure 14: The Schematic Flow Guide Block

3.9. Simulation Time Clock

The simulation step interval for the model corresponds to one minute of elapsed time. The total simulation time can be of the order of multiple months. The simulation time clock (Figure 15) makes it easier to interpret the elapsed simulation time by converting the elapsed minutes into days and hours, which are shown in the animated display.

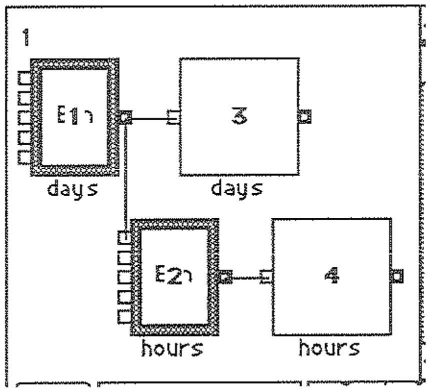

Figure 15: The Simulation Time Clock Block

4. DISCUSSION AND CONCLUSION

The models discussed have great potential use in evaluating changes to the grain handling system in Western Australia and elsewhere. As in many realistic situations, the nonlinearity and complexity of the system makes it unsuitable for analytical optimisation. Attempts to recast the problem for solution by analytical techniques such as linear programming require draconian simplifications or distortions. In a complex real world, it may be far more fruitful to use simulation to explore the possibilities of improvement, in preference to a Procustean forced fit to an optimisable analytical model.

However, a further advantage of building a simulation model is the opportunity it gives for various stakeholders and protagonists to compare and improve their conceptualisation of the system. This aspect of simulation modelling is especially powerful when a graphical dynamic package such as Extend™ is used to build the model in a form that can be meaningful to a wide range of stakeholders.

5. REFERENCE

Diamond, P. and Hoffman, P., *Extend™ Performance Modeling for Decision Support*, Imagine That, 510 pp., San Jose CA, 1995.