Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

EMULATION OF A MATERIAL DELIVERY SYSTEM

Todd LeBaron
AutoSimulations Inc.
655 Medical Drive
Bountiful, UT 84010 U.S.A.

Kelly Thompson
Rapistan Systems
507 Plymouth Avenue, N.E.
Grand Rapids, MI 49505-6098

ABSTRACT

Emulation is the process of exactly imitating a real system. Recent advances in simulation technology make it possible to emulate real world control systems by using a system's control logic to interact with a simulation model. Routing logic, PLC or PC control software, sequencing algorithms, and more can be integrated, tested, and debugged within a simulation environment. Simulation models communicate with control software and provide animation and statistical output for evaluating control logic and material handling systems.

Traditionally, simulation models have been a good tool to test and refine system algorithms and control logic. However, once refined, the algorithms and logic must be re-implemented in the real system. One of the main benefits of emulation is that it eliminates the need to re-implement code. Because the actual control system is used to develop, test, and refine algorithms and logic, it exists as developed in the real system. This eliminates re-implementation errors and provides greater confidence in the emulation results.

Emulation has been used for a Rapistan Systems project to test, debug, and optimize complex algorithms and control logic. Emulation of the complex pick and pack conveyor system will be presented. The emulation approach and benefits that have been achieved will be discussed.

1 SYSTEM DESCRIPTION

A Rapistan Systems client site uses a material handling system to move picked items from storage to packing stations. The material handling system consists of conveyor sections which continuously move carriers around a closed loop that connects all pick and pack stations. There are 55 possible pick induction locations (larger boxes) and 120 possible pack locations (diagonal sections). The loop conveys carriers in a counter-clockwise direction. A graphical representation of the pick and pack conveyor system is shown in Figure 1.

Orders for small components must be picked and packed for shipping. Hundreds of orders are processed each day. Orders consist of a wide variety of component types and quantities. An order may contain one or more component types picked from the same or different pick stations. All of the components for a particular order are assigned and routed to a specific pack station. The order is processed when all of the components for the order have been picked and routed to the assigned pack station.

While an order is being processed (packed), another order can be assigned to the same pack station to start the picking and routing operations. Components of this second order are not processed until all components have arrived at the pack station and the previous order has completed packing.

2 EMULATION

Emulation of the Rapistan control system for this project integrates a simulation model with the actual control system. The simulation model provides the output for evaluating control logic and algorithms. The simulation model also provides real time 3-D graphical animation for improved visibility and confidence.

The simulation model has a built-in message handler that receives and sends messages through a standard network interface (TCP/IP). The simulation message handler pulls message information from the server at predefined time intervals and acts on these messages. In addition, the simulation model sends messages to the server when certain events have occurred within the simulation model. This process is illustrated in Figure 2.

A main benefit of emulation over pure simulation is it eliminates the need to re-implement code. By developing and refining the control logic and algorithms in the control software, it exists as developed. Code developed and refined in traditional simulation models must be re-implemented into the actual control software if it is to be used. This creates the possibility of communication and re-implementation errors.

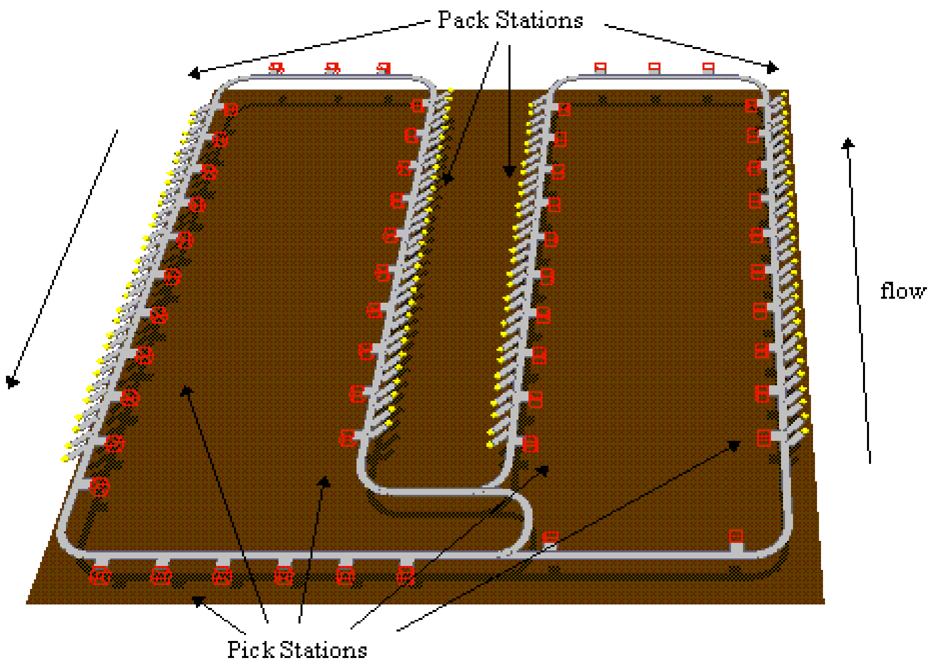

Figure 1: A Pick and Pack Conveyor System

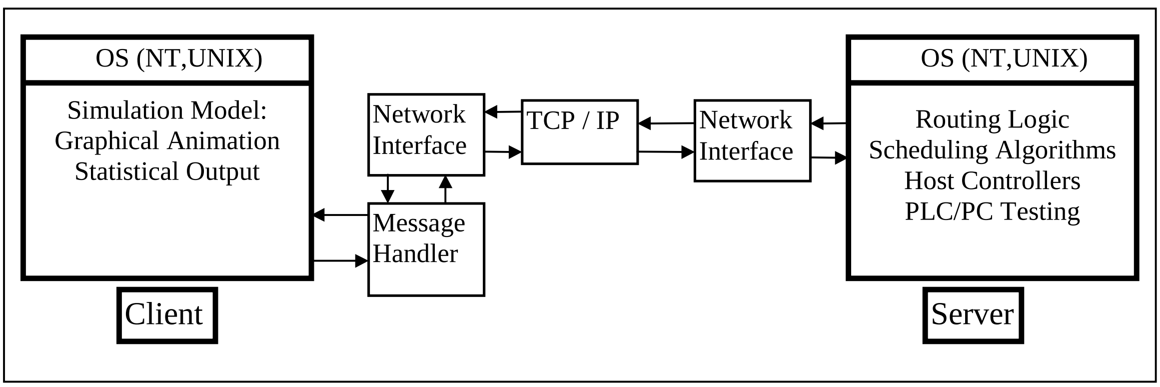

Figure 2: Typical Emulation Structure

2.1 Time Scale Considerations

Time scale is the ability to run emulation models faster than real time. For example, a time scale of 10 would emulate 10 minutes of events in 1 minute of real time. Time scale is one advantage of emulation, but caution must be taken. Because emulation uses information from a real time system, it is not possible to turn the animation off to quickly process the events as is possible in traditional simulation models. To do this in emulation introduces errors into the emulation results. This is caused by communication delays between the controller or server and the system (simulation model). Consider an emulation with a communication delay of 0.1 second. With animation on and running at real time, there are no errors because the emulation and animation are running real time and the simulation model cannot process events faster than real time. However, with the animation turned off, the simulation model processes events as fast as it can and many events can be processed in 0.1 seconds. This introduces sequencing errors into the emulation, which undoubtedly leads to erroneous results. For this reason, there are limits to the time scales at which emulation models can run.

3 PROBLEM DESCRIPTION

At the beginning of each day, the number of orders and all components in each order for that day are known. For example, the first three orders in an order profile (there are hundreds of orders) may look as shown in Table 1. In this example, order 1 requires two picks to be made at pick station 45. Both of these picks (component types) will be assigned and routed to the same pack station. Likewise, order 2 will require a total of five different component types be picked from three different stations. One pick will be made from station 12, One from station 18, and three from station 25. All five of the picks will be assigned and routed to the same pack station.

Table 1: Part of a Daily Orders Profile

Order #	Item #	Pick Station	# Pick Types
1	1	45	2
2	1	12	1
	2	18	1
	3	25	3
3	1	4	1
	2	6	2

etc.

Historical data was used to generate daily order profiles (as in Table 1). This file was used to test the order release and pack station assignment algorithms. Different order release and pack station assignment algorithms were developed and refined using emulation to optimize the system. The goal in developing algorithms was to process the required number of orders per day within the planned facility schedule. Fully utilizing the pack stations is key in accomplishing this goal. If pack stations spend a lot of time waiting for the components within an order to be picked and routed, the overall utilization of the system and the total number of orders completed is reduced. Emulation provided a valuable tool in testing order release and pack station assignment algorithms.

4 ALGORITHM DEVELOPMENT

The final system algorithm and logic used in the control system are proprietary. However, during the refinement process, two initial algorithms were developed and compared. These two algorithms are called the FIFO and RULE1.

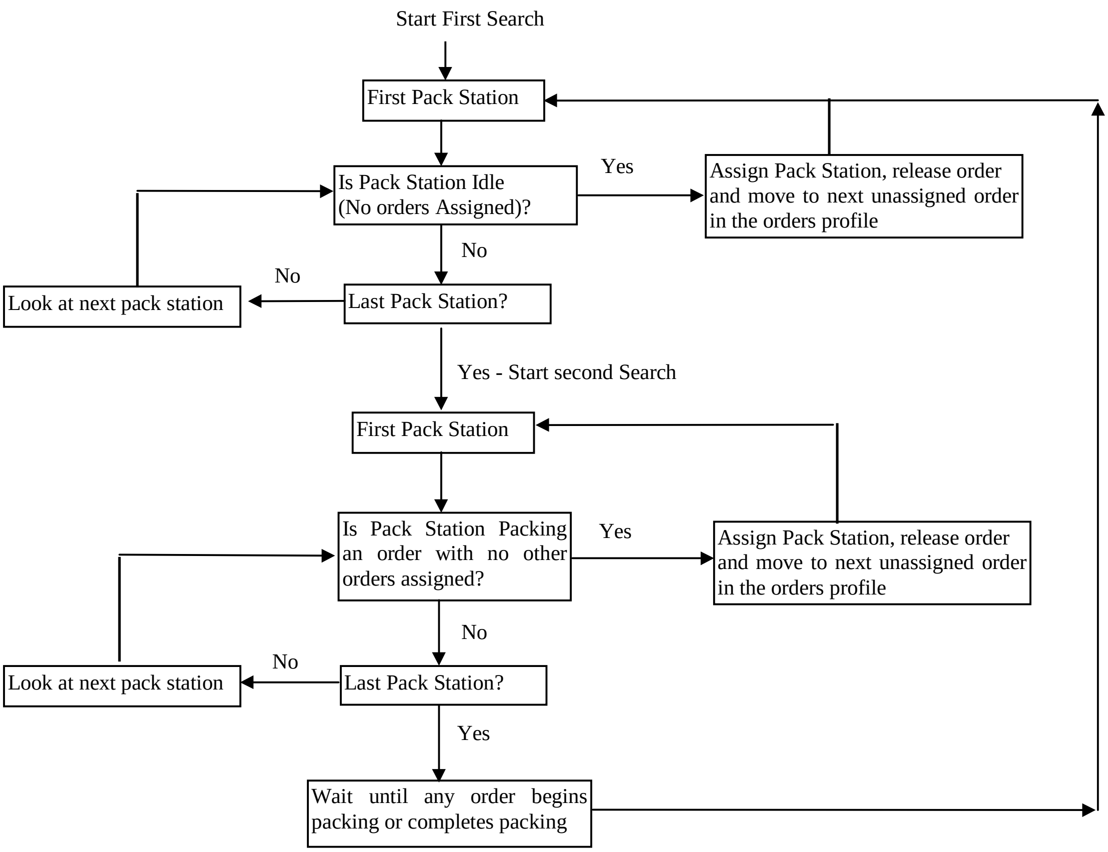

Figure 3: FIFO Pack Station Assignment Algorithm

4.1 The FIFO Algorithm

The FIFO algorithm implements a simple order release sequence based on entry in the orders profile. Orders are released based on entry in the orders profile, top to bottom. Two searches are made to assign each order a packing station. Figure 3 illustrates the FIFO pack station assignment algorithm.

The first search loops through all pack stations beginning with the first pack station. If the pack station is idle and has no orders assigned to it, the controller assigns and releases the order and moves to the next unassigned order in the orders profile.

The second search loops through all pack stations beginning with the first pack station. If the pack station is currently working on one order, and no other orders have been assigned to it, the controller assigns and releases the order and moves to the next unassigned order in the orders profile.

If the controller fails to find a pack station after going through the first two searches, it waits until it receives a signal from any pack station indicating it has either started or finished an order. At this time, the controller repeats the order releasing sequence beginning at the first loop and attempts to assign a pack station for the next unassigned order in the orders profile.

4.2 The RULE1 Algorithm

RULE1 was developed to improve the FIFO algorithm. RULE1 optimizes the pack assignment and order release sequence by selecting pack stations nearest the picking requirements. RULE1 prioritizes orders based on the number of picks for the order and releases them accordingly. For example, orders that have only one pick attempt to be released before orders having two picks. Orders with two picks attempt to be released before orders

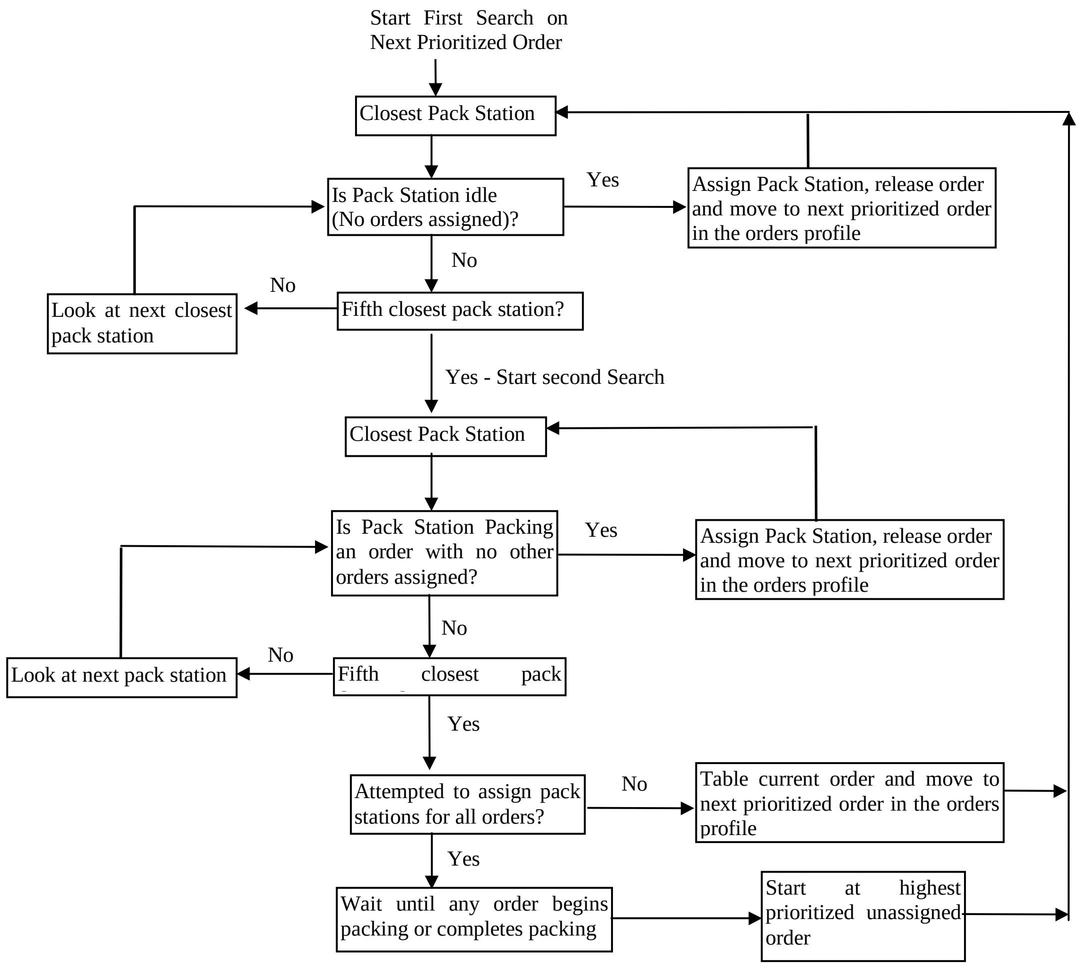

Figure 4: RULE1 Pack Station Assignment Algorithm

that have three picks, etc. In addition, the pack station is assigned based on its travel distance from the most downstream pick station in the order. A pack station must be one of the five closest pack stations to the most downstream pick station in the order to be assigned. A possibility of two searches are still made to assign each order a packing station.

The first search loops through five of the pack stations beginning with the closest pack station to the most downstream pick station in the order. If the pack station is idle and has no orders assigned to it, the controller assigns and releases the order and moves to the next unassigned prioritized order.

The second search loops through five of the pack stations beginning with the closest pack station to the most downstream pick station in the order. If the pack station is currently working on one order, and no other orders have been assigned to it, the controller assigns and releases the order.

If the controller fails to find a pack station after going through the first two searches, it tables the order and moves on to the next unassigned order. Once it has attempted to scheduled all unscheduled orders, it waits until a pack station has completed an order, or is ready to begin packing an order before looping through all of the unscheduled orders again.

Figure 4 illustrates the RULE1 pack station algorithm.

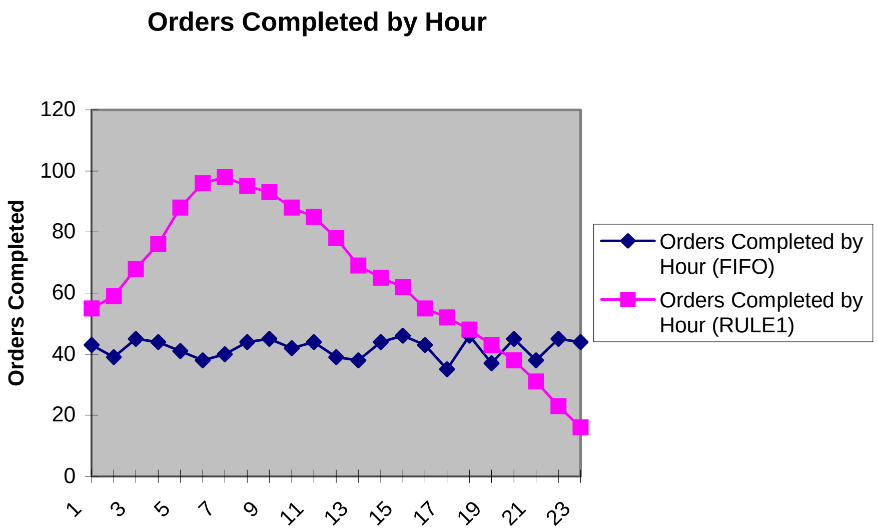

Figure 5: Orders completed by hour for two algorithms

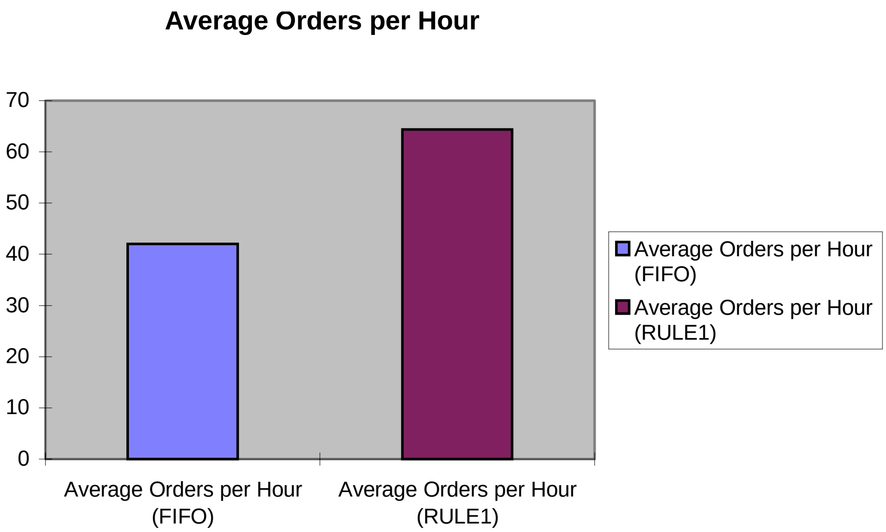

Figure 6: Average orders per hour by algorithm

5 ANALYSIS

The hour-by-hour orders completed over the 23-hour period is shown in Figure 5. Using the FIFO algorithm, the orders completed by hour remains somewhat constant. This is expected as the orders are scheduled as they appear on the orders profile and no regard is given to relative pack station location. Using RULE1, the orders completed peaks at hour 7. This is because RULE1 prioritizes orders based on the number of items in the order. Early in the day, orders with fewer picks are scheduled first. Orders with fewer picks (items) can be processed faster than orders with more picks. As the day goes on, the orders with more picks dominate the unscheduled orders. Since these orders require more time to pick and route to the assigned pack station, the orders completed per hour declines. RULE1 also schedules orders based on pack station location which reduces delivery times and increases the pack station utilizations.

The emulation results indicate that using the correct order scheduling and pack assignment algorithm is key to improving pack station utilization and system throughput. Using the default FIFO algorithm, the average orders complete per hour is 42 (965 orders completed per work day). Using RULE1, the average orders complete per hour jumps to 64 (1472 orders completed per work day), an increase of 52%. This is illustrated in figure 6.

6 SUMMARY

Emulation is a valuable tool used in developing control algorithms and scheduling rules. Emulation provides the graphical and statistical output needed to accurately evaluate different algorithms and control logic. Emulation can be done off line, avoiding the costs of debugging real systems. The emulation used at Rapistan Systems was able to prove that the system could handle the projected growth in daily orders. In addition, emulation provided the tool to refine the scheduling logic used by the control system, providing great confidence that the system could meet the daily order requirements.

REFERENCES

Rohrer, Matthew W., "AutoMod Tutorial," *Proceedings of the 1997 Winter Simulation Conference*, 1997.

Law, Averill M., Kelton, W. David, "*Simulation Modeling and Analysis,*" second edition, 1991.

AutoSimulations, Inc. 1997, *AutoMod User's Manual*.

AutoSimulations, Inc. 1997, *AutoSched User's Manual.*

AUTHOR BIOGRAPHY

H. TODD LEBARON has worked for AutoSimulations since 1990 as a Simulation Analyst, conducting numerous simulation studies over the past eight years in a variety of applications. He was recently made manager of the West Coast consulting group. He also teaches AutoMod training courses and provides consulting support. Mr. LeBaron received a B.S. in Manufacturing Engineering from Brigham Young University in 1988.

KELLY THOMPSON has worked for Rapistan Systems since 1995 as a Simulation Consultant analyzing and optimizing a wide range of Rapistan Systems. He was recently promoted to Senior Project Engineer for the Software Products Engineering Group where he manages large scale software projects. Mr. Thompson received a B.A. in Technical Communications and an A.A. in Computer Programming from Boise State University.