Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

A SIMULATION MODELER'S VIEW OF A CAROUSEL

Edward J. Williams

206-2 Engineering Computer Center
Mail Drop 3
Ford Motor Company
Dearborn, Michigan 48121-2053, U.S.A.

Kevin A. Fosso

Production Modeling Corporation
411 Sixth Street
Columbus, Indiana 47201, U.S.A.

ABSTRACT

We describe the concept of a carousel as seen from the business-process and the system-modeler viewpoints, compare and contrast various algorithms for picking from a carousel, and discuss the simulation modeling logic of one such algorithm in detail. The results and conclusions from simulations of two carousel systems are presented.

First, we present an introductory overview of a simulation study motivating a detailed examination of carousels. Next, we define a carousel and present an overview of the picking operation relative to a carousel. We then describe four algorithms for picking and discuss in detail the representation of these algorithms within simulation models, followed by a presentation of results from such simulations. In conclusion, we summarize carousel-configuration and picking algorithm conclusions.

1 INTRODUCTION

A supply company in the Detroit area wanted to improve its parts storage and order retrieval system. The existing system stored all items on shelves, from which orders were picked manually. Two proposed systems from two vendors replaced most of the shelves with flowracks, horizontal carousels, and vertical carousels, from which orders could be picked semiautomatically. Production Modeling Corporation wrote computer simulations of the two proposed systems in AutoMod II, so that the supply company could compare their performances.

When programming the simulation model, one goal was to determine realistic carousel picking times without increasing the simulation run time by scheduling every carousel move. An algorithm was developed that computes the picking time for multiple picks at one carousel bank, allowing one simulation event to be scheduled to simulate several carousel actions. We claim the improvement in simulation run time was significant.

To provide background, we describe the concept of a carousel as seen from the business-process and system-modeler viewpoints, and compare and contrast various algorithms for picking from a carousel. We then discuss the modeling logic of the algorithm for computing the time for multiple carousel picks. We conclude with some results and conclusions from the comparison of the two proposed order-picking systems.

2 DEFINITION OF A CAROUSEL

A carousel is a storage area where components for filling an order are stored in anticipation of that task. A rotating spice shelf ("lazy susan") in a kitchen serves as an example of a carousel. Every spice bottle in the shelf will be needed for some recipe; probably no particular spice bottle will be needed for all recipes; some spice bottles will be needed more often than others. Each of these considerations affects the efficient design and operation of a carousel.

In industrial practice, a carousel comprises multiple shelves, drawers, or bins containing parts or workpieces. These parts or workpieces will be needed to fill orders, just as the spices will be needed to follow a recipe. The orders to be filled may be external (destination a customer) or internal (destination a downstream process).

Industrial carousels characteristically implement semi-automated storage and retrieval. The carousel may be connected to a chain loop that moves the storage location to the carousel's exit point. Hence the implementation is neither completely manual (a worker goes to the shelves, drawers, or bins to select the needed parts) nor completely automatic (an automated storage/ retrieval [AS/RS] system). For visualization, consider the laundry shop's chain on which customers' clean laundry is sorted by last name; the clerk steps on a treadle, thereby revolving the chain and bringing the individual customer's laundry to the service counter.

3 OVERVIEW OF THE PICKING OPERATION

Picking is the operation of selecting needed items from the carousel and moving them from the carousel's exit point to an order container. Picking is typically a manual operation, such as the laundry clerk removing laundry from the chain loop and handing it to the customer. Picking is frequently controlled by a computer, which sorts the picks and controls movement of the carousel with the goal of making picks as fast as possible.

Optimization of the picking operation, as discussed below, may require multiple carousels, collectively called a "carousel bank." Multiple carousels allow the operator to make a pick from one carousel while the other carousel(s) are rotating to the next pick. By the time the operator completes a pick, another carousel will probably be ready for the operator to make the next pick. Therefore, operator waiting time is reduced and the number of picks per operator-hour is increased.

4 VARIOUS ALGORITHMS FOR PICKING

Algorithms for picking compete with one another relative to various objectives such as:

- minimizing capital investment for the carousel or carousel bank
- minimizing total motion of the carousel
- minimizing the average time to fill an order
- minimizing the worst-case time to fill an order
- maximizing the utilization of the picker
- maximizing the fraction of orders correctly filled.

These objectives and performance metrics are similar to those identified by Takakuwa (1991) for computer-aided cart systems and by Gunal, Grajo, and Blanck (1993) for automatic storage and retrieval systems.

Four commonly used algorithms are individual picking, cluster picking, rolling picking, and pick-to-light.

In individual picking, one order enters the carousel (or carousel bank), all picks required for its fulfillment are determined, the sequence in which these picks will be performed is determined, the picks are performed, and the order is moved from the carousel exit point to the order container. This process is then repeated for the next entering order.

In cluster picking, multiple orders enter the carousel bank simultaneously. The determination of the picks and their sequence is made, and the picks are performed. All orders then exit the carousel. This process is then repeated for the next entering group of orders.

In rolling picking, multiple orders may be in the carousel bank concurrently, but need not enter simultaneously. When a new order enters the carousel, its required picks are determined, their best sequence determined in view of the picks already scheduled, and this sequence merged timewise with those previously scheduled picks. Orders exit the carousel when filled, and the exit sequence of orders need not match their entry sequence.

Pick-to-light picking can be used in conjunction with any of the above picking algorithms. In pick-to-light, containers for orders are placed on a "sort bar." When an item is ready to be picked from the carousel, one light shows the exact location of the item within the carousel and a corresponding light shows which order contained on the sort bar will receive that item.

These competing picking algorithms have various advantages and disadvantages. For example, the pick-to-light method tends to maximize the fraction of orders filled correctly, but increases the capital expenditure.

5 SIMULATION MODELING OF PICKING ALGORITHMS

The valid, credible modeling of picking algorithms for carousels requires attention to several key issues.

First, realistic orders must be generated. In this context, the end user of the model will probably have available massive historical data gathered from actual orders. Statistical parameters needed in the model can then be conveniently obtained by loading these historical data into a spreadsheet, thereby advantageously using the large data capacities and statistical capabilities of current spreadsheets. In the simulation study that motivated this paper, 78,000 orders representing 230,000 picks were conveniently analyzed using the macro features and rich statistical function set of Microsoft Excel 4.0.

Second, especially if there are many carousels in the model, scheduling events for each pick will drastically increase simulation run time. This problem is especially severe when the picking algorithm under analysis is either cluster picking or rolling picking. Candidate solutions to this run-time problem are:

a) Develop a separate micromodel of the carousel; use the results of this micromodel in a macromodel of the entire order-picking or work-process simulation. This constraint of model scope is analogous to that advocated by Graehl (1992) in the context of power and free conveyors, which are likewise materialhandling systems of high complexity and potentially intricate detail.

b) Reduce the number of scheduled events by explicitly calculating the time required to pick an entire cluster of multiple orders. Performing this calculation usually works well for cluster picking, because the programming effort to perform the calculation saves the run time required for scheduling all the picks of the cluster. However, performing this calculation is less advantageous with rolling picking, because events must still be scheduled to check for newly arriving containers when space is available in the cluster and to release each container when it is finished picking.

For the simulations of the two proposed order picking systems, an algorithm was developed to implement alternative (b). The explanation and Figure 1 below describe how the algorithm works for a bank of three horizontal carousels with one picker. With minor modifications, the same algorithm was also used for vertical carousels and with carousel banks containing two carousels.

first pick, and to the previous pick for all other picks. In the diagram above this two-stage sort would produce the pick list A_2, A_1, B_1, C_1.

2) Calculate the move time for each carousel move. In the example above, the moves would be: current location of carousel A to position A_2, position A_2 to position A_1, current location of carousel B to position B_1, current location of carousel C to position C_1.

3) Set the cumulative order pick time to 0.

4) Determine which carousel has the minimum next-move-time. In the diagram above, that carousel is B because B_1 is closer to the operator than either A_2 or C_1. Add the minimum next-move-time to the cumulative order pick time.

5) Subtract the minimum next-move-time from the first move times for all carousels.

6) Subtract the time needed to pick the item out of the carousel (in this example, the B_1 pick time) from all **other** carousels (in this example, carousels A and C) because the other carousels are still moving while the item is being picked from carousel B. Add the pick time to the cumulative order pick time.

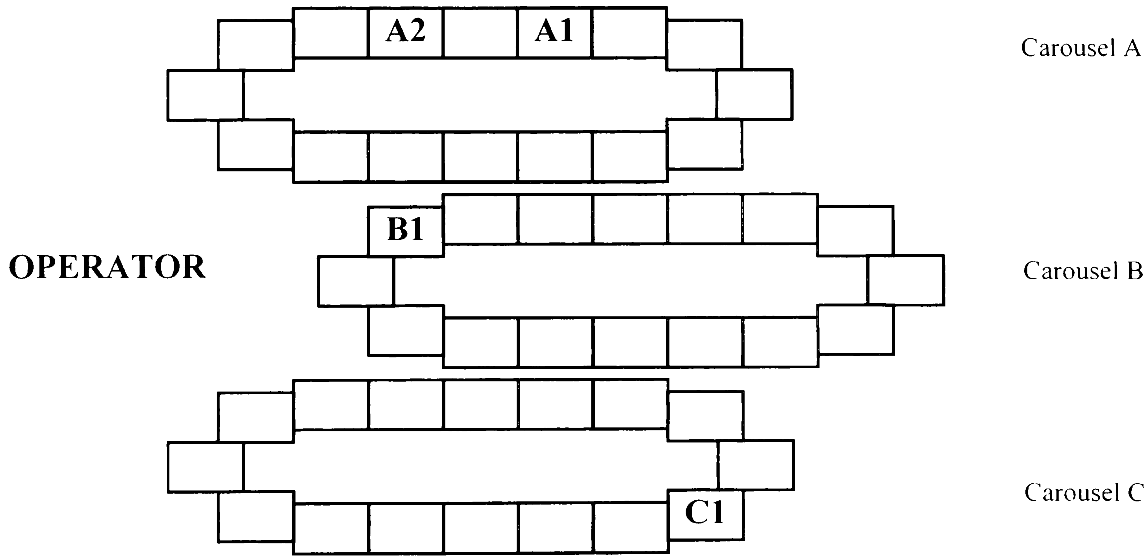

Figure 1: Bank of Three Carousels with One Picker

Suppose the following picks must be made to fulfill an order: A_1 from carousel A, B_1 from carousel B, C_1 from carousel C, and A_2 from carousel A. The modeling algorithm would then undertake the following steps:

1) Sort the pick locations by carousel. Within that sort, sort by closest location to the operator for the

7) Determine the next-move-time of the carousel from which the current pick is being made (in this example, carousel B).

8) Subtract the time required to place the picked item into the order container from the movetime of **all** carousels. In this example, subtract the time required to move pick B_1 to the carousel-bank exit point, since

all carousels A, B, and C will be advancing to their next pick positions while B_1 is being transported to the exit point. Add the place time to the cumulative order pick time.

9) If any carousel move time becomes less than zero as a result of this subtraction, set that move time to zero. If this reset occurs, it represents the situation wherein the carousel advancement time was less than the move-time + pick-time + place-time elapsed in steps (4), (6), and (8).

10) If the cluster pick is incomplete, return to step 4. If the cluster pick is complete, the calculated cumulative order pick time should then be used to schedule the time-of-completion event for this cluster pick.

6 SIMULATION RESULTS

The two systems were simulated multiple times with multiple replications to determine the maximum orders per day that could be processed in an average of seven hours. Processing time and picker utilization statistics for the two systems are presented below. The results summarized in Table 1 use rolling picking, whereas those summarized in Table 2 use cluster picking.

7 SUMMARY OF CAROUSEL-CONFIGURATION AND PICKING-ALGORITHM CONCLUSIONS

Based on the above results, the following conclusions were made regarding the carousel picking objectives and picking algorithms:

- In the carousel systems simulated, cluster picking yielded the highest number of picks per operator-hour.
- In the carousel systems simulated, rolling picking was less efficient than cluster picking. Two possible sources of inefficiency were discovered for rolling picking: (1) the extra operator work required to process individual containers may outweigh any carousel movement efficiencies gained from always having a large number of picks for the carousel to choose from; (2) with cluster picking, the fastest picking order usually causes the carousel to move in one direction for the entire cluster. With rolling picking, the continual introduction of new picks will sometimes cause the carousel to change direction, resulting in wasted carousel movement over areas of the carousel that were already recently picked.

Table 1: System #1. Rolling Picking at Carousels, two Carousels per Bank

Orders per day	1400	1500	1550	1600
Minimum Processing Time (hours)	6.09	6.57	6.86	6.97
Average Processing Time (hours)	6.20	6.79	7.11	7.39
Maximum Processing Time (hours)	6.42	6.94	7.40	7.77
Average Operator Utilization at:				
Vertical Carousels (%)	54.9	53.8	54.1	54.8
Manual Picking at Shelves (%)	64.0	73.1	75.3	77.9
Horizontal Carousels (%)	83.6	90.3	93.9	97.4
Flowracks (%)	53.4	62.3	64.1	66.0

Table 2: System #2. Cluster Picking at Carousels, two or three Carousels per Bank

Orders per day	2500	2600	2700
Minimum Processing Time (hours)	6.37	6.57	6.97
Average Processing Time (hours)	6.54	6.79	7.39
Maximum Processing Time (hours)	6.76	6.94	7.77
Average Operator Utilization at:			
Vertical Carousels (%)	89.5	91.4	96.7
Horizontal Carousels (%)	89.0	92.0	96.0
Flowracks (%)	86.7	88.3	93.9

- In the carousel systems simulated, three carousels in a carousel bank resulted in minimal operator waiting time for carousel movement.
- In a picking system with multiple picking locations or with multiple storage types (e.g., horizontal carousel, vertical carousel, flowrack), evenly distributing the picking load among the picking locations was extremely important for maximizing system throughput. In the above simulation results, system #1 is very unbalanced, whereas system #2 is very well balanced.
- In the cluster picking system, it is important to provide a nearby accumulation space for the next cluster to form while a cluster is being picked. Otherwise, the operator has to wait while the orders for the next cluster arrive.

ACKNOWLEDGMENTS

John Dennis, analytical software consultant, Quality & Product Information Systems, Ford Motor Company, Dr. Onur Ulgen, president of Production Modeling Corporation and professor of Industrial and Manufacturing Systems Engineering at the University of Michigan - Dearborn, and an anonymous referee all made valuable criticisms toward improving the clarity and organization of this paper.

APPENDIX: TRADEMARKS

AutoMod II is a trademark of AutoSimulations, Incorporated.
Microsoft Excel is a trademark of Microsoft Corporation.

REFERENCES

Graehl, D. W. 1992. Insights into Carrier Control: A Simulation of a Power and Free Conveyor through an Automotive Paint Shop. In *Proceedings of the 1992 Winter Simulation Conference*, eds. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson. 925-932.

Gunal, A., Eric S. Grajo, and David Blanck. 1993. Generalization of an AS/RS Model in SIMAN/CINEMA. In *Proceedings of the 1993 Winter Simulation Conference*, eds. G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles. 857-865.

Takakuwa, Soemon. 1991. Design and Analysis of Computer-Aided Cart Systems for Picking Discrete Types. In *Proceedings of the 1991 Winter Simulation Conference*, eds. Barry L. Nelson, W. David Kelton, and Gordon M. Clark. 365-374.

AUTHOR BIOGRAPHIES

EDWARD J. WILLIAMS holds bachelor's and master's degrees in mathematics (Michigan State University, 1967; University of Wisconsin, 1968). From 1969 to 1971, he did statistical programming and analysis of biomedical data at Walter Reed Army Hospital, Washington, D. C. He joined Ford Motor Company in 1972, where he works as a computer analyst supporting statistical and simulation software. Since 1980, he has taught evening classes at the University of Michigan's Ann Arbor and Dearborn campuses, including both undergraduate and graduate classes using GPSS/H, SLAM II, or SIMAN.

KEVIN A. FOSSO holds a bachelor's degree in mathematics (Wabash College, 1979) and master's degrees in computer science, business administration, and industrial engineering (Purdue University, 1987; Indiana University, 1991; and Purdue University, 1992). From 1980 to 1989, he did computer programming and product development at the Naval Avionics Center, Indianapolis, Indiana. He joined Production Modeling Corporation in 1992, where he works as a systems analyst developing computer simulations. Since 1994, he has been responsible for opening a new office for PMC in Columbus, Indiana.