ON-LINE SYSTEM SIMULATION

R. J. Ceci and G. W. Dangel
The National Cash Register Company
Hawthorne, California

SUMMARY

The objectives and description of a simulation model in the evaluation of On-Line Systems are discussed. In addition, the design and implementation of the model is considered. Finally, simulation results obtained from the model are presented.

INTRODUCTION

This paper discusses a discrete stochastic SIMSCRIPT simulation model developed by NCR to be used as a tool in the evaluation of On-Line Systems. The On-Line System (Figure 1) consists of remote terminals, line concentrators, communication multiplexors and a central computer system. The remote terminals communicate with a communication multiplexor via the line concentrators. The communication multiplexor contains its own local processor. This local processor is connected via an intercoupler to a central processor. Other similar local processor configurations may also be connected to the central processor system. The simulation model, however, consists of a single local processor configuration connected to a central processing system.

OBJECTIVES AND GENERAL DESCRIPTION

The simulation model was developed to evaluate the performance of both the software and hardware for proposed On-Line Systems. The model was designed so that a variety of systems configurations and user applications could be simulated without recoding major portions of the model. Such model flexibility was accomplished by modularizing and parameterizing the design of the simulation model.

The parameters which can be varied easily include the following:

- Application parameters: e.g., number of different transactions, transaction logic, user program times, etc.
- Hardware parameters: e.g., line speeds, scanning times, cycle stealing rates, etc.
- Configuration parameters: e.g., number of remote terminals assigned to each line concentrator, number of line concentrators connected to the communication multiplexor, etc.
- Software parameters: e.g., interrupt service times, verify times, etc.
- Traffic parameters: e.g., customer arrival rate, distribution of different transaction types, load distribution for different branches, etc.
- Number and types of different simulation reports.

The simulation process generates customers (transactions) with a specified Poisson arrival rate and assigns them to a remote terminal according to load distribution tables. The transactions then follow the sequence of events from remote terminals to multiplexor to central computer and back to the terminal depending on the type of transaction created. The model computes variable transmission times depending on the number of characters transmitted and accounts for variable interrupt service, verify, disc operation, and user times.

MODEL DESIGN

The model was designed to simulate a general On-Line System, although it was first applied in an on-line retail environment. Therefore, the following model design details are described with respect to a retail application.

The retail environment may be envisioned as a typical supermarket (Figure 2). The simulation model is concerned with the arrival of customers at the checkout counters and the procedures involved in servicing the customers. The cashier uses a remote terminal to record and control (credit authorization, department and class validation, etc.) customer sales. The terminal sends all pertinent sales information to a computer system where inventory control, sales statistics, etc., are maintained on a real-time basis.

Each retail transaction was segmented into a sequential string of basic messages. The transaction type depends upon the media of exchange (cash, charge), the number of items purchased, and whether or not a discount will be given. An example of a retail transaction would be a charge, two or more items, no discount transaction (Figure 3). The number of input and output characters shown in Figure 3 are used in the computation of the transmission times (between a remote terminal and a communication multiplexor) for each step in the transaction flow.

Each step in the transaction flow is represented by a basic message type. Each message type consists of a sequential string of basic hardware functions. An example of a message type is the sequence of hardware and software functions necessary to verify an account number. A graph with respect to time of the hardware and software functions is shown in Figure 4. The SIMSCRIPT flow chart of the same message is shown in Figure 5.

Hardware Functions

The model executes the message and function strings in an interpretive fashion, i.e., each element of the string is analyzed sequentially to determine the transaction flow and/or hardware function. This processing technique is the key in achieving the desired flexibility and is schematically shown in Figure 6.
The message strings are comprised of the following hardware functions:

A - Line concentrator turn on
   The 'line concentrator turn on' function controls the requests for the line usage of all remote terminals and initiates the message processing.

B - Terminal-to-multiplexor transmission and vice versa
   The model accounts for cycle stealing during terminal-to-multiplexor transmissions.

C - Local processor usage
   Requests for local processor usage are processed according to a priority scheme; application processing is generally interruptable.

D - Local disc usage
   The 'local disc usage' function controls and services the disc requests. The disc serves only one transaction at a time.

E - Local processor-to-central computer transmission and vice versa
   Again, the model accounts for cycle stealing during processor-to-central computer transmission.

F - End of message function
   The 'end of message' function releases the line concentrator and initiates the scanning mechanism which detects line requests from remote terminals that are assigned to the released concentrator.

Depending upon the type of transaction, each basic message consists of a specific mix of the hardware functions noted in steps B through E.

The initiation and completion of a hardware function may necessitate a request for executive software service. The executive service is performed by software functions which are independent of the message routines and their hardware functions. The executive software controls and initiates future hardware functions of the active system messages.

A simplified representation of the hardware functions B, C and D are shown in Figures 7, 8 and 9, respectively to illustrate some stages of the transaction process.

MODEL IMPLEMENTATION

The implementation of the model from its inception was accomplished in less than four months. The use of generalized NCR-developed SIMSCRIPT routines made this short implementation time possible. Some examples of these routines include the following:

- A debugging-validation package was inserted in the model allowing the user to check the internal model logic without performing extensive hand calculations. This validation aid is extremely valuable for a complex model for which there are no comparable analytical results obtainable.

- A generalized queuing routine was inserted in the model to allow the programmer to perform any standard queuing operation on all sets within the model by a single CALL statement.

- Finally, standardized data gathering and reporting routines were included in the model to provide a variety of simulation reports for the various interested parties. Furthermore, additional reporting requests can be included in the model with minimal programming effort.

SIMULATION RESULTS

The simulation model produces the following types of reports which the user can request at any desired time during the simulation:

- The systems performance characteristics are summarized in the system utilization report (Figure 10).

- System configuration and traffic data are illustrated in Figure 11.

- A number of different queuing and timing data are produced by the simulation model. An example of a timing report is given in Figure 12.

CONCLUSION

The on-line system model designed by NCR proves that a specified hardware configuration and its software could be simulated for a specific application by use of a generalized model. Thus, it is possible to use the described model to simulate many different On-Line Systems. At the same time, the model can be used for both system design and system analysis of hardware and software.
TABLE 3  TRANSACTION 2 (CASH OR MORE ITEMS, NO DISCOUNT) SERVICE TIME

<table>
<thead>
<tr>
<th>LOWER CLASS LIMIT</th>
<th>UPPER CLASS LIMIT</th>
<th>OBSERVED FREQUENCY</th>
<th>PER CENT OF TOTAL</th>
<th>CUMULATIVE PER CENT</th>
<th>REMAINING PER CENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.169466</td>
<td>55.252922</td>
<td>26</td>
<td>55.9%</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Fig. 12