Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

A MULTI-PURPOSE TOLL COLLECTION PLAZA MODEL

Andrew J. Junga

Applied Systems Modeling, Ltd.
P.O. Box 1239
Anderson, Indiana 46015-1239

ABSTRACT

A recent application of Automatic Vehicle Identification (AVI) technology enables traffic to flow through a toll collection plaza in a more expedient manner. Simulation is used by the AVI vendor as a means to better engineer controller equipment and to optimize the hardware/software combinations required for a given plaza. Simulation coupled with animation can be used by the AVI vendor as an instrument to show potential customers (various transportation authorities) advantages to using AVI technology on both existing and proposed toll collection plazas.

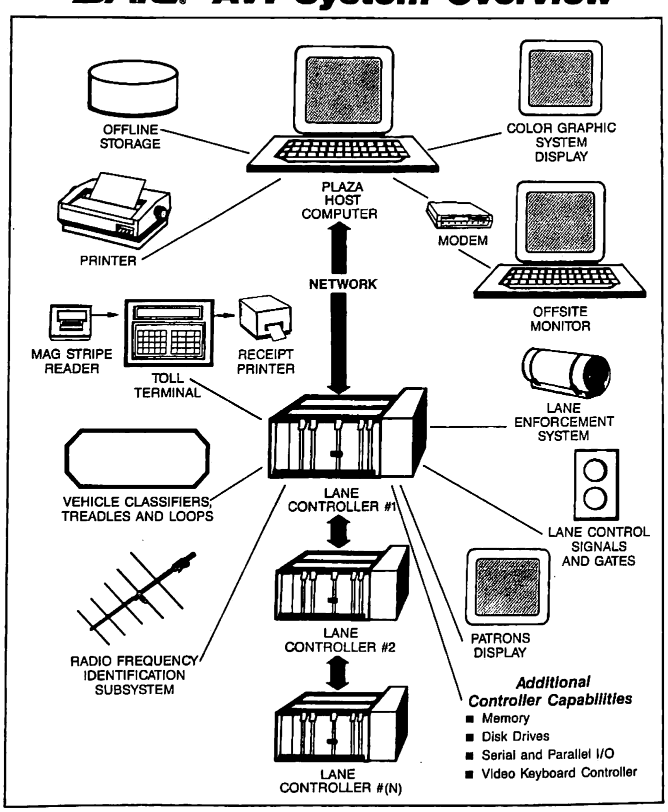
This paper will serve a two-fold purpose by showing the process involved in creating a generic toll plaza modeling package customized to meet the needs of the AVI engineering staff. It will also provide a basis for demonstrating the use of discrete-event simulation in a vehicular traffic environment as well as illustrate the capabilities to be obtained by integrating several software packages, such as a programming language, a spreadsheet, etc., into a dedicated single purpose modeling tool.

1. INTRODUCTION

Automatic Vehicle Identification (AVI) technology has recently been applied to the design of new and modification of existing toll collection plazas by Science Applications International Corporation (SAIC). AVI allows certain plaza lanes to be dedicated to AVI tagged vehicles. Transit through the plaza does not require slowing significantly or stopping for those vehicles so equipped, reducing or eliminating traffic queuing and the need for additional toll collection lanes to service an increased traffic volume. SAIC's AVI product brochure illustrates the equipment utilized (Figure 1) and describes its operation.

Goals of speeding traffic flow and reducing delays at toll plazas prompted the need for lane configuration experiments. Several approaches had to be tested to find optimum combinations for a given plaza and vehicle mix. Use of AVI in conjunction with manual collection, dedicated AVI lanes supplementing manual collection lanes, an AVI-only plaza, etc., are examples of these varying approaches.

SAIC AVI System Overview



The system is designed to remotely identify vehicles using small radio frequency transponders (tags) that have been mounted on the vehicles. As the vehicle passes by the monitoring equipment the tags are energized and a unique ID is captured and transmitted to the system controller where the account status is then verified.

The verification process involves insuring that the account is currently active and has an adequate balance. Approval by the control system will cause the account status to be displayed to the driver and the illumination of a signal light indicating that they may proceed. This entire process takes less than one second to occur. Disapproval by the control system will result in the notification of plaza personnel and manual collection of the toll.

Although the use of Radio Frequency Transponders is the key to remote vehicle identification there are a number of other elements that are equally critical. A sophisticated and flexible control architecture capable of reliably processing a variety of inputs and outputs in real time is essential.

Coupled with the software that enrolls users, prepares management reports, monitors lane activity and handles the financial elements, AVI affords toll organizations with the opportunity to reduce operational costs, increase throughput and minimize fare collection irregularities.

For further information on AVI please contact:

Bob Redding
Science Applications International Corporation
Automatic Identification Systems Division
4161 Campus Point Court
San Diego, CA 92121
(619) 458-5292

Figure 1. AVI System Description From SAIC Brochure

The AVI project manager had previous experience using discrete-event simulation to solve engineering problems and recognized its potential usefulness in this application. The expertise required to assemble an easy-to-use modeling package was not readily available within the company. The decision was reached to acquire outside expertise in order to meet internal deadlines and preserve already strained engineering resources.

A two phase approach was selected to better evaluate the capabilities that a simulation model could provide. The first phase was based on cursory specifications and was required to be completed quickly, producing a package to aid in the further refinement of additional specifications later. This was to allow an opportunity to examine 'hands-on' the merits of a simulation model of this application. Pending successful outcome of package testing by SAIC, a second phase would produce a more embellished product with greater capabilities.

The following sections describe the activities and status of each phase. A summary section presents overall project observations, problems encountered, and pitfalls to be avoided.

2. PHASE 1

The purpose of Phase 1 was to create and effectively analyze the usefulness of a simulation model in a toll-collection plaza application. A cost-effective package was required that would meet known specifications in a timely manner and allow AVI vendor testing.

2.1 Simulation Package Considerations

Discussions with the project manager led to the following requirements: The package had to include the use of a specific general purpose simulation language, a user interface, and the requirement that it all be capable of functioning on an IBM-PC compatible laptop computer. User data modification screens and customized model output reports were necessary. The model had to be generic in nature to allow for any number of plaza and AVI equipment operational parameters to be modified. The ability of the model to read external files containing traffic data was requested. The capability of the model to automatically open or close plaza lanes based on some predetermined algorithm was to be explored for feasibility. The eventual addition of animation to the package was a consideration from the onset to provide a means of showing AVI's advantages to a customer. It was decided, however, in the interest of time and for lack of a high-performance PC-based animation engine to investigate this further for Phase 2 implementation.

The use of GPSS/H was requested by the AVI project manager as a result of his past experience with the product and proved to be well suited to this application. The high compilation and execution performance afforded by GPSS/H produces excellent results when using the PC platform for simulation. GPSS/H for MS-DOS had just recently become available and offered a fully-functional implementation, with model size limited only by the well-known 640K memory restriction.

GPSS/H is rich in extended input/output (I/O) features that allowed data to be read from and written to files without the use of external routines in languages such as Fortran or C [Henriksen and Crain 1989]. GPSS/H, by its general purpose nature, allows modeling tangible entities (vehicles) and complex controller logic with one language.

The vendor provided traffic data for an existing toll collection plaza considering the installation of the AVI equipment. This approach would allow for comparisons of the model operating without AVI consideration to real-life experience as an aid in model validation.

While there was an enormous amount of data already collected and refined by toll agencies, a large portion of it was not in a useful form, requiring additional analysis and further data reduction or conversion. In most cases, this data was used to generate frequency distributions the model would eventually draw from to establish parameters (Vehicle Type, Method of Toll Payment, etc.). However, nearly all of the furnished data could be easily altered in the Phase 1 model, eliminating any need for additional collection or reduction of data at that time. More exacting data, better representative of a given traffic or plaza state, could be used in the model at a later date.

This wealth of information did provide the basis for establishing the level of detail to accommodate in the model's data-file structure. For example, Vehicle Inter-Arrival Times (IAT) were captured and expressed in fifteen minute increments, while Vehicle Type distributions were given as hourly statistics. Upon searching through all data variables, it was obvious that hourly intervals for establishing traffic and plaza parameters would provide enough resolution without drastically taxing the performance of the running model or adding to the amount of data necessary to make this package a useful tool.

The data-file structure maintains the necessary variables that are accessible to be changed by the user. This accommodated a method of allowing the user easy access to all parameters that can be changed to perform sensitivity experiments without requiring knowledge of a given simulation language or experience in general simulation principles.

During the beginning stages of model development a spreadsheet analysis is often helpful in data reduction. As a means to save considerable time developing a data-revision user interface, access to these variables was by way of a Lotus 1-2-3 spreadsheet that incorporated protected cells to prevent modifying the file layout. This also allowed a means to verify numeric values as needed and ensuring, for example, that the values in a given distributions added up to 100%. Simply writing the spreadsheet to an ASCII print-file permitted the model to read all of this data at run-time. As an alternative, an ASCII editor or word-processor could be used to change these files as well. The latter approach, however, was not recommended because of lack of error checking and possibility of file layout corruption.

2.2 Data Considerations - Traffic

Data included as model traffic parameters were:

1. Distribution of the total number of vehicles entering the plaza by hour.
2. Distribution of the number of occupants in each vehicle entering the plaza by hour. This is important in certain locales that promote car pooling by reducing or eliminating tolls and dedicating express lanes for high occupancy vehicles (HOV).
3. Up to fifteen different Vehicle Classes such as Auto, Auto with one, two, or three axle trailer, Motorcycle, Truck with differing numbers of axle trailers, Busses, etc. These classes were to be identified by name and number.
4. Distribution of Vehicle Class by hour of the day. This would allow for the distinction between rush-hour Auto traffic and Truck deliveries usually made mid-morning or later. It was determined that the model should be representative only of week-days, eliminating the need to consider a seven-day week.
5. Up to twenty different Toll Payment Classes. This would allow for not only AVI equipped vehicles, but a multitude of toll payment methods allowed at some plazas, including correct change, cash, asking booth attendants for directions, etc.
6. Comments specific to a given configuration or data that would also print with the customized reports. This proves useful when using multiple files to experiment with many different pre-established traffic scenarios and a given plaza configuration.

2.3 Data Considerations - Plaza

Data included as toll plaza parameters were:

1. The number of Toll Collection Method Classes available with this plaza configuration and whether or not each method required the services of the AVI controller equipment. In some cases, Automated Toll Machines (ATM) were tied to the controller to ensure proper payment and signal appropriately to the driver.
2. For each combination of Vehicle Class and Toll Collection Payment Class, the following:
   A. Distribution of Service Time at the booth itself.
   B. The Toll Collection Amount.

3. Whether or not a given Vehicle Class was eligible to use the HOV/Special Use lanes.

4. Information about relative times for this Vehicle Class to move through the queue to the booth and then move out of the collection zone after toll payment. This implies, to a degree, different vehicle lengths, velocities, and acceleration values for each vehicle type. A motorcycle is smaller and can move quickly to the booth and out of a collection zone relative to a much larger double-trailer truck.

5. Information specific (and proprietary) to the AVI controller itself, such as:

A. Number of records between upload/download to the plaza host computer.

B. Performance degradation during upload/download (and the consequent increase in service time), if any.

C. Distributions representing controller failure or malfunction and associated time until back on-line.

D. Other performance related criteria.

6. The number of lanes open for use in this plaza representation, up to a maximum of twenty.

7. For each lane, the following:

A. Lane number and queuing information.

B. Alternate controller to service this lane upon its own controller failure.

C. For each hour of the day and for every Toll Collection Payment Class, whether or not this lane will be open and accepting a given Toll Payment Method.

8. Comments specific to a given configuration or data that would also print with the customized reports. This proves useful when using multiple files to experiment with many different pre-established plaza configurations and a given traffic scenario.

The above proves to be a considerable amount of data for even a small to medium sized plaza. It affords, however, maximum flexibility in traffic or plaza configuration as a generic model. Again it should be noted that while there appears to be a great deal of information describing traffic and plaza parameters, the real emphasis of the model was in its simulation of the overall controlling mechanisms employed by the AVI system.

The capability of the model to read and process external traffic data files would allow existing plaza controllers already capable of sampling and collecting this information to 'feed' the plaza model. A 'without-AVI' and 'with-AVI' equipment comparison could be made using a customer's actual data to evaluate the benefits of the technology.

2.4 Package Implementation

Previous experience with the GPSS/H MS-DOS product demonstrated that only small to medium sized models would run due to the memory limitations of MS-DOS. For this reason, and to eliminate the replication of creating the same traffic stream entering the plaza on each run, traffic generation was created as a separate, smaller model to be run as demand required, writing the vehicle information to a file. This was in support of the plaza model reading traffic data from an external file and dramatically cut the plaza model's overall run time.

The nature of the data describing vehicle arrival times and parameters (all by hour of the day) was best suited to the use of GPSS Functions. Reading all model parameter data from an external file and using this data in a Function proves troublesome, however, so a 'GPSS code generator' was devised to read necessary data and write GPSS Function statements. The file created in this step would be merged with a baseline traffic generation model prior to run-time. While this seems complicated, it was easily implemented using a Basic program and an MS-DOS Batch command file.

It is also noteworthy that the traffic generation model was written entirely using GPSS/H Control Statements and not GPSS blocks that are normally thought of when discussing GPSS. Control Statements that typically serve to direct the operation of the model run serve as a programming language in their own right, having access to the wealth of variables and functions offered by GPSS/H.

MS-DOS Batch command files were also used to implement the primary user interface. From this, languages such as GPSS/H and GW-BASIC could be called to commence execution and DOS commands issued so that directories could be listed. Support programs, such as a share-ware file listing utility to view output and Lotus 1-2-3 or an ASCII editor could be called to change traffic or plaza parameter data. As a commercial product, this approach would not normally be recommended. However, to quickly put together an environment to 'test-drive' in lieu of a complete product specification, this approach serves very well, particularly for non-programmers, saving about three-fourths of the time (or more) normally required to code an elegant interface in a high-level language such as Basic, C, or Fortran.

Discussions and supporting data concerning driver lane selection logic indicated that there are almost as many discrete algorithms employed by drivers to select a lane as there are drivers. The data supported that a vehicle would, depending on its lane of origination, select a 'zone' (made up of usually two or three lanes) and then make a lane selection from a lane in that particular zone. This approach to lane selection was incorporated in the model, superseded only by the assignment of certain lanes dedicated to select Vehicle Types (eg., Auto-only lane, HOV/Special Use lane, etc.).

The original model opened and closed lanes hourly according to the schedule established in the plaza parameters file. A second plaza model was created that automatically opened and closed lanes based on traffic volume and queuing. This was accomplished by changing about ten lines of code in the original plaza model to implement the open/close algorithm as specified by the AVI vendor. The MS-DOS memory limitation required this multiple-model approach.

The automatic lane scheduling model presented some additional problems. All lanes were assumed open from the start and one lane at a time was closed based on the number of queued vehicles. Damping was required in this logic to prevent a succession of open then closed states when traffic volume reached a critical point. As this was an interesting exercise, its usefulness lies only in the controller's ability to open or close those fully automated lanes where no booth operator would be affected.

Information used by Toll-Authorities had to be extracted from the simulation runs. Select data would be reformatted and written to an output file for later viewing and analysis, saving the user from learning to read and interpret lengthy simulation output statistics [Schriber 1974]. This was easily implemented via GPSS/H external file I/O capability and allows for easy modification at a later time.

The Traffic Generation model was written to create a traffic stream (and associated parameters) for any single or multiple one-hour periods during the course of the day. A one-hour period of traffic was generated automatically to serve as plaza model warm-up and stabilization. Random Number Stream offsets were user-selectable, providing a means to create different traffic streams for the same time of day, saving the results in various Traffic IAT files.

The Plaza model was written to read the Traffic IAT files and emulate the AVI controller logic. About half of the lines of GPSS/H code for this model involved controller logic implementation. This model also provided all necessary customized output reports. The Plaza model, as the Traffic Generation Model above, was written to incorporate the use of Random Number Stream offsets, allowing for multiple experiments with a single plaza configuration and traffic stream.

Documentation was created that described the use of the package and the logic included in the GPSS/H models. Everything was then shipped to SAIC for installation on an IBM-PC compatible laptop and subsequent testing.

3. PHASE 2

The purpose of Phase 2 was to allow for testing of the entire package for feasibility as both an in-house engineering tool and for external use as a sales tool to illustrate AVI concepts and advantages to customers. Several approaches to marketing this package as a stand-alone product were also explored. Overall, the package was received favorably and served the purpose of a research platform.

Activity in the area of toll plazas has heightened with the realization that a 'pay-for-use' approach to new road construction may soon become more common-place. Several similar and dissimilar technologies have emerged that have created a renewed interest in toll plaza technology from a design standpoint.

Adding animation and a better defined and implemented user-interface were the two areas to focus on for future development. The intent suddenly shifted from an in-house engineering tool with sales applications to a turn-key package to be sold to or accessed on a time-sharing basis by other Toll Authorities.

Possibilities examined as feasible solutions will be explored and commented upon in the following:

3.1 Improve The Package And Run On An IBM-PC Compatible Or Equivalent Laptop:

Adding animation is now practical because of new software available today. PROOF Simulation/Animation/Presentation software (Wolverine Software Corp.) is as of this writing in a Beta-test stage. This particular toll-plaza model has been animated using PROOF and the results appear to be very promising. Using PROOF for animation only requires the insertion of a few statements at several strategic locations in the model to write commands to an external file that are later read by the animator [Brunner 1990]. PROOF offers high performance animation at a relatively low price and the product is based on IBM-PC Enhanced Graphics Adapter (EGA) or Video Graphics Array (VGA) hardware.

Also in a Beta-test stage as of this writing is GPSS/H 386 simulation software. This is a fully-functional version of GPSS/H that runs on the IBM-PC compatibles (utilizing the 80386 processor) under MS-DOS. GPSS/H 386 virtually eliminates the barriers presented by the 640K memory limitation of the MS-DOS version. The typical performance increase and elimination of model size impediment this version affords would allow both the traffic generation and the plaza models to be combined into a single model, executing more quickly than either one individually does at present.

Third party software that facilitates the creation of a user-interface has eliminated much of the programming required for this task. Routinely, front-ends are created for models that present options as drop-down menus, a custom data editor with error checking, context-sensitive help screens, etc. A sample user-interface screen is shown in Figure 2.

This option is the overall most cost-effective approach. Not only have hardware and software prices decreased, the accompanying performance improvements make the overall utility of the IBM-PC compatible platform a hands-down choice for this application.

3.2 Improve The Package To Run On A Workstation Class Computer To Allow Dial-up Access:

This approach was investigated to determine the feasibility of allowing engineers the capability to time-share and run the model remotely. While the GPSS/H models would require no conversion to run on the workstation, all other components of the user-interface and options would require change. Hardware and software costs would both be significantly higher for only a marginal performance improvement.

Comparable animation software, user-interface development software, etc., is not nearly so prevalent in this class of machine as it is on the IBM-PC. Workstations and the UNIX environment provide an attractive development arena and excellent performance in execution. However, the use of workstations is not as 'mainstream' today as is the use of PCs.

While this option is attractive for certain other reasons to the vendor, it would not provide the cost/benefit ratio as a turnkey modeling package compared to option 1 above.

3.3 A Hybrid Of The Two Options Above:

This alternative considered using a workstation as a host computer to be dialed into by modem in order to change data and run simulation experiments. The calling party would have access

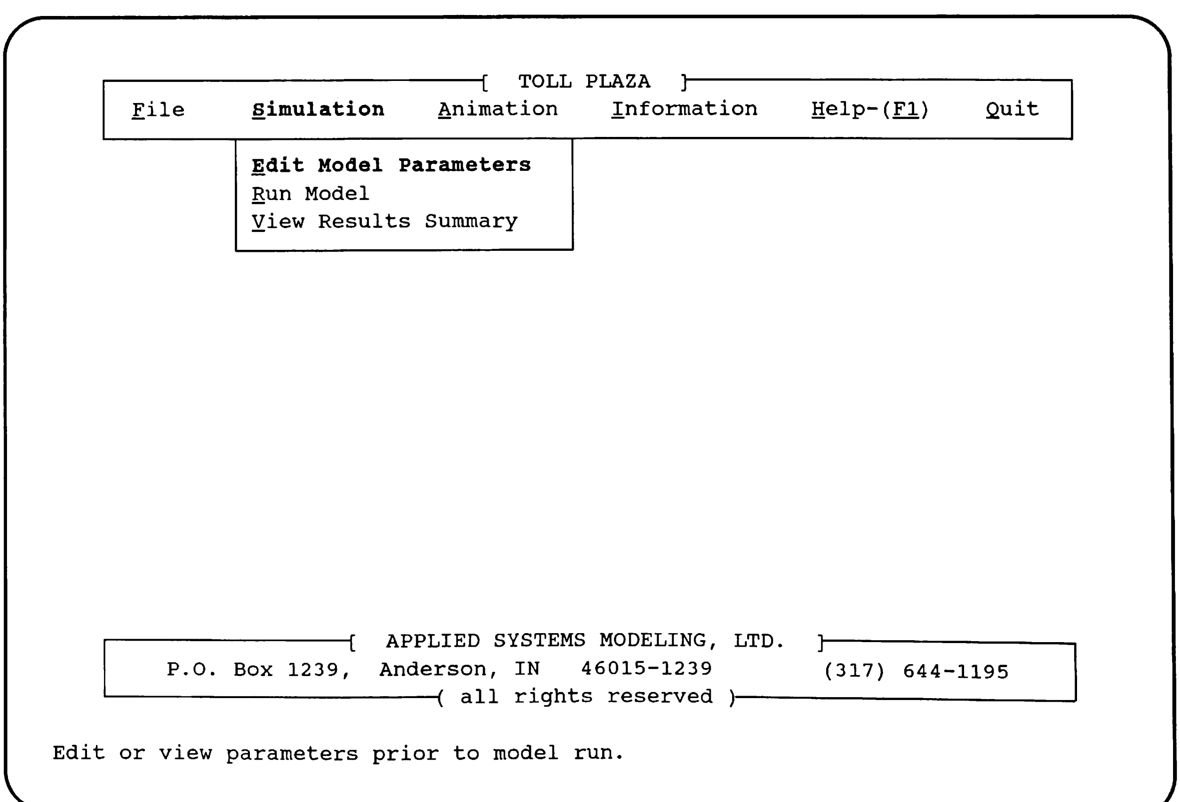


Figure 2. Sample of User-Interface Screen

via phone-line to all aspects of the package with exception of viewing animation while on-line.

To accommodate animation, the user would first run the model and create the necessary animation files. These files would be either mailed on floppy-diskette or down-loaded over the phone line. The user would have PROOF animation software on a local PC to run the animation and view the results.

The benefits to this approach were the AVI vendor would have the sole copy of the simulation models and could easily update the logic as the product developed. The user would use the product on a pay-for-use basis. The simulation would run quickly with the processing power of the workstation and the animation could be locally viewed by the user.

Drawbacks to this approach are the communication expense and speed, the delay in viewing the animation, and possible access delays to the host workstation. Software expense is again relatively high and there were concerns regarding the implementation of the interface for any number of terminal types that could potentially access the workstation.

At the time of this writing, testing is continuing with the Phase 1 package and specifications are being refined.

4. SUMMARY

This section will present overall project observations, problems encountered, and pitfalls to be avoided. This is included to facilitate the novice to simulation or the experienced attempting to assemble a similar type of package.

There are several simulation packages on the market today that offer some, if not all, of the features included in this project. Icon based model builders, graphics based model/animation builders, etc., all have a place as analytical tools. Many are geared towards modeling in a given segment of engineering, such as manufacturing/material handling, communication networks, etc.

While these packages may perform well for many instances of modeling, the user is left with the burden to examine the fundamental operation of their constructs, such as a conveyor, AGV, etc. In some cases, writing external routines in a high-level language is necessary to perform tasks not within the confines of these specific purpose packages. Complex, customized control logic is one of these areas.

On the other extreme, general purpose simulation languages carry the stigma of simulation products available many years ago. Some are perceived as difficult to use, requiring more time to build and debug a model than necessary with today's specific purpose packages. Most will afford the ability to model at any level of detail at the expense of longer development times and the need for experienced, well trained modelers.

The estimated total cost of all the software packages required to do this project with the approach taken is considerably less than if a special purpose simulation package had been used, especially when animation is included. Most of the software used, such as GW-Basic, Lotus 1-2-3, etc., is familiar to most PC users, should any problems be encountered.

An on-going process of development occur's in the area of the user-interface, or the so called 'front-end'. This is a time-consuming part of the overall development process that requires certain programming skills. The final product, tailored to the customer's requirements, is the result of many hours of work and testing.

Specifications should be established early enough to allow ample time to develop a data file structure. This is an area that is affected by both the user-interface (data editor) and the simulation model itself. A simple change here could propagate many changes elsewhere in the product.

This approach to modeling seems to be an ideal compromise between the power of the high-performance simulation language and the user friendly menu system. It could be applied to virtually any simulation language or package. The importance is in selecting the best tool for the task, and creating a useful, well-behaved product to get the expected results.

As always, decisions made by users not experienced in the use of simulation could have more disastrous results than if no simulation was used at all. It is imperative that such a product be delivered with precautions and instruction on its use. Areas of requiring instruction include the process of experimenting with simulation and the interpretation of its results.

ACKNOWLEDGMENT

The writer wishes to acknowledge Mr. Robert T. Redding, Science Applications International Corporation, for granting permission to use the SAIC Toll Plaza Simulator as the subject of this paper and for his comments and suggestions on its content.

REFERENCES

Brunner, D.T. (1990), *PROOF Documentation*, Final Alpha 5.3, Wolverine Software Corporation, Annandale, VA.

Heriksen, J.O., and R.C. Crain (1989), *GPSS/H Reference Manual*, Third Edition, Wolverine Software Corporation, Annandale, VA.

Schriber, T.J. (1974), *Simulation Using GPSS*, J.Wiley & Sons, New York, NY.