USING DATA DRIVEN SIMULATION TO BUILD INVENTORY MODEL

Minghui Yang

Boeing Demanding Forecast IMM
19300 International Blvd.
Seattle, WA 98124, USA

ABSTRACT

Many general-purpose simulation languages (such as ARENA, SLAM II, GPSS/H, SIMAN etc) have been the major simulation tools to simulate the demand-supply processes. They have made great contributions to the decision making. But in the recent years, followed by the fast development of WINDOW applications, powerful PC hardware and software, numerous applications have used different approaches to develop simulation applications. One rapidly developing area in simulation is dynamic data-driven simulation (DDDS) by using data manipulation and analysis packages. SAS is a powerful tool for data analysis and data manipulation. It can also be used to build simulation models in data driven applications. This article presents our research development in this area for demand forecast applications.

1 INTRODUCTION

Some traditional simulation languages (for example SLAM II, ARENA, GPSS/H, and SIMAN) are specially designed for simulation. In the 90's, they were the major tools to simulate complicated processes. Even today, many of them still provide very useful results for decision making. Numerous papers and books in their applications have been published. Courses on general-purpose simulation languages have been offered in numbers of higher education institutes. When a general-purpose simulation language is used to construct a simulation model, a discrete event simulation model (including NETWORK model and animation) is often generated.

Compared with other commonly used software such as JAVA, C++ and C#, the market share of the general-purpose simulation language has not reached to a certain scale yet. In fact, some popular used packages such as SLAM II or SIMAN in early 90's are no longer popular as before because many new vendors with newer software, products and methods have appeared on the simulation market. Although this is the nature of the software

industry, there are several observations that the author would like to make some comments.

It is obvious the rapid development of the WINDOW visual programming and computer games in the last 10 year have big impact on the directions in the discrete event simulation. If any traditional general-purpose simulation language fails to follow the trend, it is likely the software will be out of the competition. In the recent years, followed by the fast development of WINDOW applications, powerful PC hardware and software, numerous applications have used different approaches to develop simulation applications. One rapidly developing area in simulation is dynamic data-driven simulation (DDDS) by using data manipulation and analysis packages.

The term DDDS comes from the dynamic data-driven application system (DDDAS) which was formalized by Darema (2000) around the time of a National Science Foundation (NSF) workshop in March of 2000 and is a new and fairly active area of research. The data in a DDDS is fed into an executing simulation model either as the data is collected or from a data archive. The dynamic portion of the DDDS implies that the input data will be used to influence the measurements for additional data it may require. Current research for these systems focuses on simulations of physical/artificial/social entities. Many of these simulations are inspired by older models used to make predictions about physical systems (i.e. weather forecasting, oil exploration, etc). The simulation can then make predictions about the entity regarding how it will change and what its future state will be. The simulation is then continuously adjusted with data gathered from the entity. The predictions made by the simulation can then influence how and where future data will be gathered from the entity, in order to focus on areas of uncertainty.

There are several issues and factors which influence the fast development of the data driven simulation modeling technology.

The technical support and software cost is a serious concern to many companies which do not purchase certain software unless the software can be widely used in

the company with low software cost per machine and with the vendors' technical support to all the update versions. The high cost of software is forcing many companies to keep only certain numbers of software for their operations. At the Boeing Company, the IT department plans to reduce the number of software licensed at Boeing from several hundreds to less than one hundred. Even lots of simulation software are popular and powerful with good functionalities, they may still be removed from the IT support list, or the number of licenses is significantly reduced due to the high cost. So many analysts have to use the existing software to construct simulation models. DDDS is a method which can utilize the existing software well.

How to maintain/allocate the adequate simulation resources efficiently within a company organization is also an issue due to the limitation of the simulation usage. It is often takes fair amount of time to learn a general-purpose simulation language then to develop a model. The development process includes the model construction, network flow, program compile, model validation, model output development and analysis. After model constructed, the language itself may not be used frequently. To use the general-purpose language frequently, the best way is to have a centralized simulation team within a company. Their major task is to build different simulation models. This may not be the situation for many companies including large companies. Many departments at Boeing do need to use simulation as a tool for decision making. To build a useful simulation model, it does not only require one to know the language itself, but also to know the real system very well and can transform the real system by an adequate simulation model. The centralized simulation team knows the language itself very well but perhaps not the real system. Vice versa, those experts know the system very well. But they may not have enough training in modeling and simulation.

At the Boeing Integrate Material management division (IMM), the Demand Forecast group has great background in math., statistics, operations research, economics, social studies and other related areas. But very few people in this group had special training in the general-purpose simulation languages. The discrete event simulation model construction requires "art" as well as the creative thinking. It also requires generating the input data with certain distributions to the model. To obtain the distributions of the actual data, it involves a quite work for the data processing and analysts. The data processing process implies a lot of extra work.

On other hands, massive input data and software to collect and generate the data are already available in many companies. Under the DDDS modeling development, modelers are required to make decisions with regard to which data to absorb, when it should be absorbed, and how it should be absorbed. Our Boeing IMM Demand Forecast group uses SAS dealing with large amount of data daily. SAS has very strong data processing ability and is the software for our group to build the data-warehouse and perform forecast. Instead of using or purchasing the general-purpose simulation language, we use the existing data and SAS to develop data driven simulation models which become great tools in many decision making projects for Boeing IMM.

2   THE INVENTORY SIMULATION MODEL

Keeping adequate inventory is a very important factor for airlines to have desired service level. When the maintenance division requires certain quantity of a part and the part is available at the desired warehouse, a success occurs. Otherwise, a failure occurs. Most airlines require very high service level from 97% to 99%. Many parts are expensive with high holding costs. It is a challenge to manage the inventory level to achieve the service level with the minimum costs. How to set up certain reorder point (ROP) and reorder quantity (ROQ) is a difficult task. Hillier and Lieberman (1980) provided some basic and useful formulas in their famous book "Introduction to Operations Research".
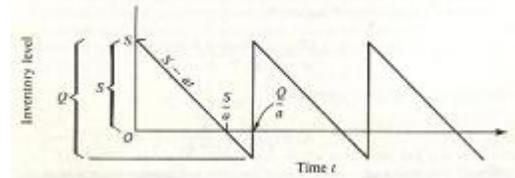

Figure 1: An inventory model with failure allowed

Following data are assumed available at the given airlines:

- The snapshot of the airline inventory level at the given time is known.
- The issue and return quantity of all parts of the maintenance division of the given airline during the given time period.
- The lead time of each part after order is placed is given and unchanged during the study period. That is, the number of days from the part ordered to the part arrived to the maintenance facility is considered a constant
- The price (or cost) of each part is given and unchanged during the study period.

Of course, other data would be great help such as maintenance schedule data, maintenance type data (A check, C check or D check), airplane flight data (flight hours and cycles) and airplane model type. But in many cases, some or all the desired data might be difficult to be provided by the airlines.

If an optimization model based on the parts historical consumption data has been built and all the ROP and ROQ have been generated for the airlines maintenance group, then it is important to check if the ROP and ROQ generated from the model can achieve the desired service level. The best tool is to use simulation to verify the ROP and ROQ.

3 IF A DISCRETE EVENT SIMULATION MODEL APPLIED

We may use any general-purpose simulation language to build the discrete event simulation inventory model. It is important to define all events and nodes associated with the model. Following events and nodes might be needed (SLAM II terminologies are used here as an illustration):

- Arrival event: the customer demand arrival distributions which may or may not be exponential distributed.
- Server: if one warehouse is used, one server system is applied. The service rate may or may not be exponential distributed.
- Failure rate: since the desired service level may not be 100%, certain percent of failures are allowed
- AWAIT node is needed when requested quantity is not satisfied.
- When parts are not needed and returned to the warehouse, the inventory level changed. This is another arrival process. The return interval and its distributions also needed for the model
- When a demand arrives, the inventory level changes. An ASSIGN node is required
- When the inventory level is below the ROP, an order is placed. The model needs to schedule ordered quantity to arrive to the system within certain time period (ROLT).
- Number of demand and number of failures need to be counted to calculate the service level.
- The output reports for decision makers - those output reports include monthly service level report, average daily inventory level report and parts return rate statistics report.

Boeing IMM division did not purchase any general-purpose simulation software and cannot show the detailed model in this paper.

However, the following SLAM II NETWORK chart could be an illustration of the model presented to the management.

Figure 2: An illustration of a SLAM II network model

Unfortunately the above model is only for ONE part. Many airlines have more than 50,000 parts active (with issue quantity and/or return quantity) at a given time period to achieve the desired overall service level. It is obvious that the above discrete event simulation model is not suitable to obtain the final results for all 50,000 parts at ones. All parts may have to be divided by 5 to 100 categories based on part price, part arrival time (lead time) and number of demands during the study period. If over 100 combined categories for the 50,000 parts are defined, the event interval distributions analysis for each category is a quite heavy task. It is difficult to apply the above model for Boeing IMM division to perform the decision analysis.

4 DATA DRIVEN METHOD FOR THE INVENTORY SIMULATION MODEL

An alternative way has been applied to build the simulation model by using the SAS data steps to simulate the inventory process. The methods present here is general and can be applied to other areas such as Airport Traffic control, correction center capacity analysis and other industrial and social studies.

The major business rules and inventory policies can be formulated by the following math formulas:

$$H_i = H_{i-1} - D_i + I_i + R_i$$
$$D_i > 0 \text{ if } H_i \leq ROP + DQ_i$$

where ROP is the reorder quantity, DQ_i is the total DUEIN quantity at day i, and H_i, D_i, I_i and R_i are the inventory level, demand quantity, receiving quantity and returned quantity at day i respectively.

When an order is placed, the order quantity satisfies:

$$D_i = ROP - H_i + ROQ - DQ_i$$

where ROQ is the plan reorder quantity. For simplification, the ordered quantity D_i will arrive to the warehouse after ROLT days (the lead time).

To build above simulation model by data driven method, data is crucial point since no discrete events are needed. The major task is to design a data structure to pass the above recursive formulas to generate the output data. Then SAS is applied to use the output data to generate all output reports. The following designs and steps illustrate how the model is being built.

- The plan data for each day in the simulation is needed. That is, ROP, ROQ and ROLT should be available for each day during the simulation time

frame. And the initial inventory quantity H_1 at day one should also be available. To eliminate the impact of the initial inventory, H_1 can be set as ROP+ROQ so the simulation can be in the stable condition after simulation starts.

- The actual transaction data (ISSUES and RETURNS quantity) is required for each part at each day. If there is no ISUUES or RETURNS at that day, the quantity should be zero.
- The suitable SAS procedures should be selected. PROC IML becomes a candidate first since it deals with matrix data very well. The days in the simulation become a $n \times 1$ matrix where n is the number of days. This is similar to the clock in the discrete event simulation.
- In contract to the discrete event simulation, it is concentrated on the data passing to the model. That is, what types of data should be constructed.

Lead by Dr. Mike Huffman, a data driven simulation model has been built by SAS PROC IML to use the recursive relationship between H_i and H_{i-1}. Several very useful output variables for the inventory plan such as service level, fill rate, return rate are also calculated. After simulation ends, all inventory and service failure data have been generated by the IML matrix. So the output reports are also generated by SAS.

One issue by using PROC IML is the matrix size might be huge since no blanks (missing data) are allowed in the matrix. So all quantity at no transaction days have to be set as zero. If the part population is 50,000 and six months of operation is simulated, several matrices in the simulation will have the size 50000 x 181. When fast PC is not available, the simulation run might take several hours. Now with fast PC with more memories, it takes much less time to complete the simulation. But it may still take couple of hours to complete the simulation to compare two or more plans. Improving the simulation running speed becomes our next development.

5 FURTHER DEVELOPMENT OF THE INVENTORY SIMULATION MODEL BY USING HYBIRD METHOD

The PROC IML runs slow when the matrix is huge. In our inventory model, there are lots no-activity days during the simulation period. All issue or return quantities are zero in those days. But it consumes the majority of computation resources to go through the no-activity days in the simulation.

A new "hybrid" method has been created by the author to re-construct the model. The new method combines the concept of discrete "events" and SAS data steps. The advantage of using PROC IML to build the model is the coding and design is easier since the output data is generated at each time point during the simulation period. So if the output data is generated only at the time when an "event" occurs, the running speed will be much faster.

The following events are created by the hybrid method: "INITIAL" event, "ISSUES" event, and "RETURN" event. For each "INITIAL" event and "ISSUES" event, an artificial "RECEIVE" event is created first. The key design idea is that each "RECEIVE" event must be paired with an "INITIAL" event or an "ISSUES" event. When no order is placed at the "INITIAL" event or the "ISSUES" event, the corresponding "RECEIVE" event data should be removed from the simulation output data. It means, the data needs to be visited twice during the simulation. The first round is to build the structure and the second round is to fill the data which includes deleting unnecessary "RECEIVE" events from the data.

Because each "RECEIVE" event is paired with an "INITIAL" event or an "ISSUES" event and the lead-time is a constant during the simulation for each part involved in the simulation, the "RECEIVE" event date will be R days later after an "INITIAL" event or an "ISSUE" event where R is ROLT (lead time) days. This assumes that each "INITIAL" event as well the "ISSUES" event accommodated with an order activity. We know this definitely not true since at the "ISSUES" event and "INITIAL" event, an order activity will occur only when in INBIN quantity is below the ROP. Since we have the artificial "RECEIVE" event with certain date, then the whole data can be sorted chronologically according to the event order. This is very similar to the SAS PROC IML except IML output occurs at all time point and the hybrid method outputs data only when it is needed.

The challenge now becomes how to eliminate some "RECEIVE" events since their paired "INITIAL" event or "ISSUES" event may not have an order activity. By using the RETAIN statement at the SAS data steps, we are able to tracking the RECEIVE events to decide which one should be eliminated. When no order occurs, the corresponding "RECEIVE" event will be eliminated from the SAS data. The SAS data is sorted by part number, event date (several events could happen at the same day) and event for processing. In order to achieve more efficient calculation, the part population is classified as several groups by number of "ISSUES" event since each "ISSUES" event requires a "storage" place for the possible ordered quantity. After a particular part processed, the storage space (memory space) for the part will be release and reset for the next part. All the above events are done by using SAS steps and other SAS procedures. Compared with using the PROC IML, the coding is more complicated and difficult.

The detailed codes of this method will not be presented here. The final results show that the simulation runs very fast. It takes about one minute to simulate 50,000 parts inventory modeling for one year operation that is almost 100 times faster than using the PROC IML method. Since each event time point can be stored, the time interval between events can be calculated. Thus all reports available by PROC IML method can also be generated by using the hybrid method.

The hybrid method is data driven and the actual data is manipulated and passed to the model. Similar to the PROC IML method, the data has to be formulated as certain ways with certain variables to run the simulation.

6   COMMENTS

The discrete event simulation model puts more emphasis on building the model network flow and on the event construction. The data is used to support the event construction as parameters. When the data driven simulation model is applied, the model designer must pay very close attention to the data and data structure. The designer also needs to manipulate the data. In many cases, no distribution functions about the input data are assumed.

Data should be integral part of the simulation. They are not only some parameters of the model. When data is changing dynamically, the model will provide "if then" results. In fact, changed data may change the distributions of the data itself; the simulated results will reflect the results under different distributions.

All large companies have huge amount data and they already use some data processing software to deal with the data. If the software has a strong statistical analysis ability, it can be used as the data driven modeling tool to answer "if then" type of questions since the data can be manipulated easily through the data processing software. This type of data driven simulation is dynamic and can use the actual data instead of assuming data distributions to generate some random numbers.

It has a great advantage to use SAS for data driven simulation since not only SAS has very strong data processing ability, but also SAS has many statistical analysis packages to perform the simulation. The output data from the simulation is much easier to be processed and analyzed by using SAS.

The discrete event simulation network flow model generally is suitable to simulate more complicated model while data driven simulation is suitable for larger data. Since the hardware cost is much lower than decades ago, the data driven simulation model becomes much common now. But it cannot replace the discrete event simulation model since they have different emphasis and different view of the process to be simulated.

REFERENCES

Darema, F. 2004. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In *Proceeding of the 2004 International Conference on Computational Science*, 662-669.

Kennedy, C., and G. Theodoropoulos. 2005. *Technical Report CSR-05-9: Towards Intelligent Data-Driven Simulation for Policy Decision Support in the Social Sciences*. University of Birmingham, School of Computer Science. Available via <ftp://ftp.cs.bham.ac.uk/pub/tech-reports/2005/CSR-05-9.ps.gz>

Hillier, F., G. J. Lieberman. 1980. *Introduction to Operations Research*. 3rd ed. San Francisco, California: Prentice-Hall, Inc.

AUTHOR BIOGRAPHY

MINGHUI YANG, a math modeler and statistician at the Boeing Integrated Materials Management, has worked at decision analysis and statistical analysis for over 25 years. He received his Ph.D in statistics from Oregon State University in 1988, an MS in Operations Research, an MS in Systems Engineering and a BS in Electrical Engineering. He was a visiting scholar at Stanford University Operations Research Department in 1982. Having over 20 publications in Markov decision analysis, simulation and SAS data warehouse applications, Minghui is a certified SAS programmer. He has presented his research findings in many regional, national and international conferences. He has taught statistics and computer science classes at Saint Martin University, Oregon Institute of Technology and MBA program at CKGSB. He is honored by four universities in China as full professor at business school. At spare time, Minghui enjoys church services, community activities, traveling, bridge, "GO" game, skiing and golf.