EVALUATING THE GPSS SIMULATION MODEL FOR THE VIKING BATCH COMPUTER SYSTEM*

Jun-ji Lee**

Jet Propulsion Laboratory
Pasadena, California

ABSTRACT

In anticipation of extremely heavy loading requirements by the Viking mission during the post-landing periods, a GPSS model has been developed for the purpose of simulating these requirements on the Viking batch computer system. This paper presents the effort pursued in evaluating such a model and results thereby obtained. The evaluation effort consists of selecting the evaluation approach, collecting actual test run data, making comparisons and deriving conclusions.

INTRODUCTION

Two United States spacecrafts (Viking A and B) were launched to Mars on 20 August and 9 Sept. of this year. Each spacecraft has two major subsystems, i.e., lander and orbiter. Viking 1 was injected into Mars orbit on 19 June 76 and Viking 2 on 7 Aug. 76. Lander 1 landed on Mars 20 July, 1976; Lander 2 landed 3 Sept. 1976. The objective of the Viking Project is to "obtain scientific data which will significantly increase our knowledge of Mars, with particular emphasis on providing information relevant to life on the planet."

To pursue this goal both lander and orbiter are equipped with various sensitive instruments which will perform scientific experiments and engineering measurements on the planet, e.g., molecular analysis, soil physical and life detection. All data resulted from these activities from the spacecraft are received by the Deep Space Network (DSN) and processed by the Mission Control and Computing Center (MCCC) at JPL.

In anticipation of extremely heavy computer loading due to a high level of activity immediately after the landing on Mars, a simulation model (1) written in GPSS simulation language, was developed to simulate the operation of the overall Viking batch computer system. The model was used to forecast the Viking loading requirements during the post-landing periods. The model represents the data system capabilities of the Mission Control and Computing Facility and the General Purpose Computing Facility of the MCCC as they are utilized in processing data in a batch mode for the Viking Mission.

After this simulation model was developed, the Viking Project Office directed that a further task be undertaken to calibrate the fidelity of the model. In this paper we present our methodology, and the evaluation results and our experience obtained from our evaluation effort. It is our hope that our experience will prove beneficial to others who may wish to follow a similar course when they develop simulation models.

The paper is organized in the following manner. First, we briefly describe the Viking batch computer system and explain the simulation model (its purposes and main features). Next, we provide an explanation of our approach to the evaluation effort and how we have obtained our test data. We then present our conclusions regarding the goodness of our model, based on simulation results. Lastly, we present a discussion of the model sensitivity and some summary remarks.

---

[*] This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration. The Langley Research Center is Project Manager for the overall Viking Project and is responsible for flight operations. JPL is responsible for the Orbiter; MMA is responsible for the Lander.

[**] Member of the Advanced Engineering Group, Jet Propulsion Laboratory.

THE VIKING BATCH COMPUTER SYSTEM

The Viking batch computer system mainly consists of one IBM 360/75 computer (with 1 Mega-bytes of main core and 2 Mega-bytes of Large-Capacity Storage) and two UNIVAC 1108 computers located at JPL. The IBM computer is mainly for non-real-time computer processing functions, such as command generation, Viking Lander data processing and record production. The UNIVAC 1108 computers provide another non-real-time computing capability for analysis programs, including navigation, Viking Orbiter spacecraft performance analysis, Viking science data analysis and some data records production. Remote demand terminals are distributed between the two UNIVAC computers for use by project personnel in engineering analysis and program control tasks. Communication between the IBM computer and UNIVAC computers is accomplished by a high-speed electrical interface and by means of conventional magnetic tapes.

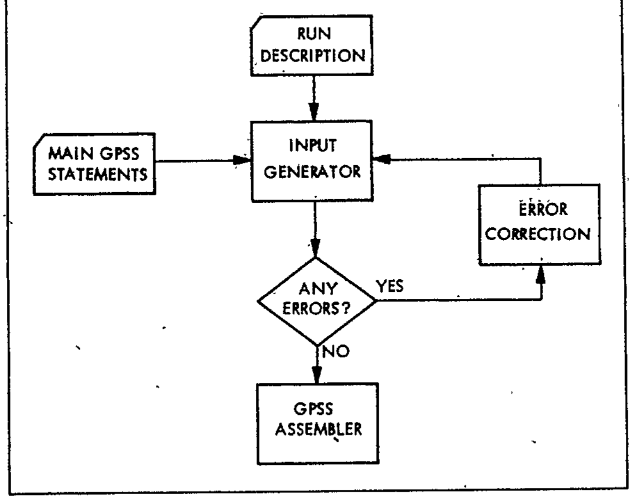
THE VIKING SIMULATION MODEL

In anticipation of heavy loading and use requirements after Viking's landings on Mars a model written in GPSS simulation language was developed with the intention the model could be used as a load forecasting and planning tool. For the convenience of following discussions only main features of this simulation model are given here.

1. The basic unit of time is one minute.
2. Input to the model are descriptions of resource requirements (e.g., CPU time, I/O time, and core size), representing a set of programs to be run.
3. A time line indicating the expected start times for all programs is required.
4. Any subset of programs can be arranged in sequential order by requiring a minimal elapsed time between the start time of one program and the end time of its preceding program. One advantage of this feature is that the electrical interface simulation becomes a trivial matter.
5. The delay created by an interactive demand terminal user is also simulated in the model.

Some of these features are made possible by utilizing the GPSS statement INITIAL. Since the model is to be used as a forecasting tool, it will be a burden to the user to set up manually a different set of INITIAL statements each time a new run is to be performed. This problem was solved by the development of a FORTRAN preprocessor, denoted as Input Generator. Its main purposes are to provide an automated generation of INITIAL statements required by each run and also to detect possible input errors. Combining with major GPSS statements, which simulate the functional operations of both the IBM and UNIVAC systems, the output of the Input Generator is a GPSS simulation model ready to be executed by the GPSS assembler. The overall system data flow is illustrated in Figure 1. For any model, once the model has been proven logically correct and bug-free, the question on validity must be considered. The Viking Project Office of NASA directed that a study task be created for the purpose of evaluating this model.

FIGURE 1.
Model Generation



EVALUATION APPROACH

As a first step in pursuing the study, a model evaluation plan was developed, which outlined our approach to the task. This approach consists primarily of the following steps:

1. To develop a 'yardstick' which will provide a quantitative measurement on the degree of model goodness.
2. To prepare and collect as many representative sets of real Viking data as possible from the scheduled Viking tests. These data should include individual and system statistics. (Since computer loads were already heavy, no special tests could be scheduled).
3. To run the simulation model with program descriptions representing programs executed in 2).
4. To compare results of 2) and 3).

5. To measure quantitatively the goodness of the model utilizing the 'yardstick' developed in 1).

For the rest of this paper, details of results obtained from the implementation of the above steps are described here.

MODEL EVALUATION ALGORITHM - Defining an evaluation yardstick.

No model can be treated with any credibility unless it has been shown that it reflects, within a given tolerance, the actual system.

There are mainly two basic ways of evaluating a given model as described in (2). One is to validate the assumptions on which the model is based and then to be sure that the model represents a proper implementation of these assumptions. The other is by comparing the observed actual system response to a set of inputs with the model response to similar inputs. It is the latter method which we have used in calibrating our model. In order to have a quantitative comparison, we have developed the 'yardstick', which is later used in evaluating the model goodness.

Some definitions of quantities and their notations are required before we present the 'yardstick'.

Given a set of sample values $X = \{ x_i, i = 1, 2, \ldots, n \}$, the variables $\bar{X}$, X_A and S are used to denote the mean, the absolute mean and the standard deviation of X. In our study, X represents either a set of observed data from an actual test or a set of data obtained from a simulation run.

In general, the evaluating yardstick can be described by Equations (1) and (2) below.

$$C = \sum_{j=1}^{M} w_j V_j \quad \text{with} \quad \sum_{j=1}^{M} w_j = 1 \qquad (1)$$

$$V_j = \left[\sum_{i=1}^{N_j} \left(x_{ji}^{G} - x_{ji}^{A} \right)^{p} \Bigg/ \sum_{i=1}^{N_j} {x_{ji}^{A}}^{p} \right]^{1/p} \qquad (2)$$

with p = any positive integer

where C = The model goodness
M = Number of evaluation parameters
V_j = The criterion variable for the j-th evaluation parameter
N_j = Number of variables accountable for the j-th criteria variable
x_{ji}^{A} = Observed test value for the i-th program or from a system parameter when $N_j = 1$
x_{ji}^{G} = Simulated value from a model run for the i-th program or from a model parameter when $N_j = 1$

As will be seen later, N = 1 when the performance of the total system for a specific aspect is to be evaluated. The model criterion function (Equation (1)) is useful because it measures the usefulness of the model rather than proving if it is true. It should be noted that Equation (1) was previously introduced in (3) as a criterion function and Equation (2) becomes a form of Theil's inequality (4) when p is set to 2. For our study, p has been assigned with only the two values, i.e., 1 and 2.

EVALUATION PARAMETERS

There are many data items available which can be utilized as evaluation parameters for a model. They may be either directly obtained or indirectly derived from obtained data. For example, job throughput-time (defined as the interval of time from initiation of a program job to the end of its execution) and system throughput-time can be obtained directly while system core utilization can only be derived indirectly.

In our study, it was decided that three parameters are more important than all others, based on the fact that the model is to be used for loading analysis purposes. These three parameters are job throughput-time, system throughput-time, and start-time delay (the difference between actual start-time and model start-time).

With this in mind, Equation (1) becomes

$$C = \sum_{j=1}^{3} w_j V_j \quad \text{with} \quad \sum_{j=1}^{3} w_j = 1 \qquad (1)'$$

while for the first two evaluation parameters, Equation (2)

becomes

$$V_1 = \left[\sum_{i=1}^{N_1} (G_i - A_i)^p \Bigg/ \sum_{i=1}^{N_1} A_i^p \right]^{1/p} \qquad (2a)$$

with p = 1 or 2

and

$$V_2 = (SysG - SysA) / SysA \qquad (2b)$$

where A_i = Job throughput-time for the i-th program obtained from an actual system run
G_i = Job throughput-time for the i-th program obtained from the corresponding model run
N_1 = Number of programs executed during a run
SysG = Model system throughput-time
SysA = Actual system throughput-time

For the purpose of conformity, the criterion value for the third evaluation parameter is defined by the following formula

$$V_3 = \sum_{i=1}^{N_1} D_i \Big/ \sum_{i=1}^{N_1} A_i \qquad (2c)$$

where D_i = start-time delay for i-th program.

It is interesting to note that Equation (2) provides a different interpretation with different values of p. For example, with p = 2 Equation (2a) places more emphasis on the effect from long programs than from short programs. With p = 1 all programs have an equal weighting in their effect on determining the closeness between the model and the actual system.

DATA COLLECTION

For the purpose of evaluating the simulation model, sets of actual run data from several representative Viking processing loading requirements have been successfully obtained. This effort was accomplished without difficulty. This is due to the fact that many Viking non-real-time ground data system tests had already been scheduled and data collection for the evaluation purpose became only a matter of including a statistics-gathering monitor in the system. The main purpose of these tests is to assess and measure the performance of Viking operational software as it operates in the IBM 360/75 batch computer and the UNIVAC 1108 computers in accordance with the Viking Flight Team time lines based upon mission operation strategy.

In less than two months we obtained 5 sets of test data from the IBM 360/75 system and 2 sets of test run data from the UNIVAC 1108 systems based on the previously scheduled Viking tests.

MODEL SIMULATION RESULTS

Based on these sets of test data, the simulation model was run separately for the IBM 360/75 part and for the UNIVAC 1108 part of the model. Plots showing the run activities in chronological order were generated for comparison purpose based on simulated data. While these plots provide visual comparison, they do not provide any quantitative measurement about the goodness of the model.

We assigned weights of 10/14, 3/14 and 1/14 to parameters 1, 2 and 3 in Equation (1)' to derive a quantitative measurement of the model. The weight assignment here is strictly subjective and is based on our belief that parameter 1 would be more important than the other two parameters in determining the model goodness for this particular effort.

Results from applying Equations (1)' and (2) are shown in Table 1 and 2. Table 1 contains 5 sets of comparison results for the IBM 360/75 system while Table 2 contains 2 sets of comparison results for the UNIVAC 1108 system.

CONCLUSIONS

While Table 1 and 2 provide detailed data regarding the degree of goodness of the simulation model, we have listed our conclusions for both systems as follows:

1. Using common forecasting practices, an estimate with 25% of an actual result is considered to be very good. In this context, since the model goodness values as shown in Table 1 are about 10% for all cases the IBM 360/75 part of the model is good. Furthermore. the test data sets represent a wide spectrum of run streams (from 28 jobs to 176 jobs in one run). It is also concluded that the IBM 360/75 part of this model is consistent.
2. Following the same argument as given in 1), since the model goodness values as shown in Table 2 are about 12% for both cases the UNIVAC 1108 part of the model is also valid. However, no conclusion can be made of its consistency due to the fact that only two sets of data were available and utilized for the evaluation.
3. Since the model goodness values obtained with p = 2 for parameter 1 are in general much less than those with p = 1, it is concluded that the model provides better loading analysis for runs which consist mainly of long jobs in both the IBM 360/75 part and the UNIVAC 1108 part of the model.
4. Test 4 data of the IBM 360/75 part of the test effort contains several subsets of jobs which were required to be executed in a specific order. This fact was reflected in the model simulation run. From the results given in Table 1, it is concluded that the model does simulate this sequential logic satisfactorily.

REMARKS

In conclusion, there are several additional remarks which deserve mention here. These have resulted from our observations

TABLE 1. Real vs Model Performance on the IBM 360/75

	TESTS					
	1	2	3	4	5	UNIT
A. NUMBER OF JOBS IN THE RUN	114	176	81	94	28	
B. JOB THROUGHPUT-TIME						
1. Mean for real system	5.6	9.8	14.9	23.2	25.3	min
2. Mean for actual system	6.2	10.2	16.0	24.4	28.0	min
C. JOB THROUGHPUT-DIFFERENCE						
1. Mean	0.7	0.4	1.0	1.3	2.7	min
2. Absolute mean	0.9	0.9	1.5	2.4	3.0	min
3. Standard deviation	1.1	2.2	4.2	5.9	7.2	min
D. COST VALUE						
1. Job throughput-time						
1.1 With p=1	15.6	9.7	10.1	10.4	12.0	%
1.2 With p=2	14.5	7.7	7.1	8.9	13.0	%
2. System throughput-time	0.1	1.7	0.5	0.1	0.1	%
3. Job-start delay time	2.1	10.8	6.6	0.7	3.6	%
E. MODEL GOODNESS VALUE						
1. With p=1	11.3	8.0	7.8	8.0	8.8	%
2. With p=2	10.6	6.6	5.6	6.9	9.5	%

TABLE 2. Real vs Model Performance on the UNIVAC 1108

	TESTS		
	1	2	UNIT
A. NUMBER OF JOBS IN THE RUN	127	46	
B. JOB THROUGHPUT-TIME			
1. Mean for real system	23.3	33.3	min
2. Mean for actual system	22.8	32.8	min
C. JOB THROUGHPUT-DIFFERENCE			
1. Mean	-0.5	-0.5	min
2. Absolute mean	3.4	4.8	min
3. Standard deviation	5.9	7.3	min
D. COST VALUE			
1. Job throughput-time			
1.1 With p=1	14.9	14.5	%
1.2 With p=2	7.1	13.0	%
2. System throughput-time	0.3	0.0	%
3. Job-start delay time	12.0	3.8	%
E. MODEL GOODNESS VALUE			
1. With p=1	11.6	11.1	%
2. With p-2	6.1	10.0	%

and experience in the course of developing the model and evaluating it. They are as follows:

1. While it is desirable to have all of the system features simulated in a model, it is often impractical to do so. Furthermore, if the purpose of developing a simulation model is for reasons other than system performance improvement and its basic unit of time is gross (one minute in our case), it will be like "using a sledgehammer to drive a tack" if the model is developed with all the system features implemented. Therefore, our model does not intend to simulate the I/O channel competition, file contention and system outage for both systems nor does it consider core swapping for the UNIVAC 1108 system.

2. The UNIVAC 1108 system is a time-sharing system which allow as many as 40 demand terminals to be active at the same time. It is therefore difficult to collect usable run data from the UNIVAC system unless it is operating under a strictly controlled environment. This explains why we have had only two sets of useful test data for this system.

3. In the course of our evaluation exercise, the simulation model took a set of data which represented the set of programs executed during a test and then compared its results with actual system run data. In other words, the evaluation of this model was performed by comparing its outputs with actual system outputs with both systems taking the similar set of inputs.

   However, in actually applying this model in the future, the input, describing jobs to be executed, may deviate from actual job characteristics. Therefore, its simulation results may be far from actual system results.

   This problem, a model sensitivity one, deserves much more attention, especially if a simulation model is to be used as a predictor. Currently, there are no plans to study this problem.

4. In developing the Viking simulation model, we made heavy use of the GPSS "HELP" block to interface the GPSS program to other FORTRAN subroutines, e.g., core-fragmentation routine and card-output routine. We believe this is one of the important advantages we derived from choosing the GPSS as the simulation language.

5. The real-time functions for the Viking mission support are performed by a dedicated IBM 360/75 computer and other smaller computers at JPL. This is the reason we did not simulate these functions.

ACKNOWLEDGMENTS

The author is grateful to M. Johnson of the Viking Project Office for his guidance on carrying out the validation task, to R. Miller, T. Smith and C. Vallado for their efforts in accomplishing the task and to D. Herrman for his comments and suggestions in preparing this paper.

REFERENCES

1. Lee, J., Miller, R. and Vallado, C. Viking 75 General Purpose Simulation System Model for MCCC Batch Computer Systems, JPL Internal Document No. 619-19, Jet Propulsion Laboratory, California, October, 1975.

2. McLeod, W. Social Simulation: Validation, Assumptions and Credibility, Simulation, Vol. 17, August, 1971

3. Schrank, W. and Holt, C. Critique of: "Verification of Computer Simulation Models" Management Science, Vol. 14, No. 2, October, 1967, pp B104-B106

4. Theil, H. Applied Economic Forecasting Rand McNally and Company, Chicago, 1966, p28