SIMULATION AS AN AID IN PRODUCT DEVELOPMENT

Paul J. Kosteva

International Business Machines Corporation

Vestal, New York

SUMMARY

The development of new computer hardware poses problems that can be answered by simulation. The sequential nature of the problem permits the use of job shop type simulating techniques. A GPSS simulator was structured with a program format to aid in problem association while simultaneously providing some functional, modular and compact organization. Essential knowledge of the development task was generated by structuring as well as operating the simulator. The lack of certain information necessary for constructing a truly representative simulator was especially helpful in determining a data characterization for future development endeavor. The simulator was also helpful in the structuring of a pilot line simulator and a linear optimization model.

PROBLEM DESCRIPTION

When industrial, ecological, and other environmental problems are not defined (because of their enormity or their strangeness), simulators are especially useful. The inclusive variables may be of such complexity and numbers that they defy description and association by other means.

The simulator relegates information into a form that is meaningful to the human because the simulator takes the form of the problem and provides output in a systematic and associative manner. Its concepting or building aids in understanding the problem. This familiarization is rather detailed depending on the level of detail considered for the simulator. In fact, the learning process is enhanced further by selecting the level of detail. If one is to select a level of detail, the investigation into the detail must be deeper than that which is selected.

With the preceding information in mind, it is logical to assume that large and small industrial development problems can be solved or eased by simulation.

The development of new computer hardware poses problems that can be answered by simulation. Simulation is especially useful because the problem is one of sequencing a part or parts through a labrynth of paths. Many variables are not related or even familiar. Regression techniques do not relate to the problems or are too gross to provide any worth because the situation is not known enough to provide more detailed analysis. Modeling is a phase that follows simulation for the same reasoning, that is, the problems are too grossly defined.

SIMULATOR DESCRIPTION

This simulator provided an analysis of a technology development line. Technology development is considered as a line here in that the product development is dependent on preceding stages. The line is highly sequential in nature permitting the use of job shop type modeling procedures. The simulator has sufficient detail to interact with individual processes and to provide sufficient real world features of the existing line. GPSS was selected because it enabled sufficient detail to be incorporated into the simulator while simultaneously providing a relative ease of program formatting.

Emphasis was placed on modularity, functionality, ease of alteration and compaction as basic properties of the simulator. An interplay of emphasis on properties was necessary to obtain a better program format. (Extreme compaction was sacrificed for modularity and problem association. Problem association permits the results of a simulation to be better related to the real world situations.)

The simulator employs a main sequence function, time functions, branching functions, and a main program loop consisting mainly of seven blocks: QUEUE, ENTER, DEPART, ADVANCE, LEAVE, TEST and LOOP in that particular order. The TEST block provided an exit from the normal sequence to blocks that address the branching function. The branching functions are especially useful for a development situation in handling variable criteria such as yields for rework and scrappage and also for changes in sequences for rework. The branching function relates main process sequence steps to TRANSFER blocks in the program format. Each TRANSFER block is unique containing individual process yield figures for rework and scrappage. These figures change rapidly from week to week and often from day to day requiring a program format in which yield figures are easily altered. Since rework usually entails variations in transactions retracing to subsequent steps, ASSIGN P-,N statements together with TRANSFER blocks serve as effective combinations. If a deviation occurs in the retracing as occurred in this simulation, another standard processing loop can be used. Usually the deviation is purposely kept small and consistent to enhance the line when and if it becomes a pilot and/or manufacturing line.

Compaction of the simulator was sacrificed to some extent by the use of many TRANSFER blocks. Functions could have been used in place of these blocks; however, fast change and process association would have been destroyed with a resultant reduced turnover in simulator operation (see Figure 1).

The small number of transactions or parts dealt with in a manual line allow a compromise in constructing the simulator for computational speed. Time and memory requirements are small for small numbers of transactions. The branching function may slow the simulator because of the indexing necessary; however, the advantages offered by the branching function definitely outweigh the small increases (if any) in computer time.

The simulator was operated from a 2741 telemetry terminal to a remote entry 360-65 computer. The output was standard; however, selective printing was performed from the terminal. Usually less than one thousand lines of print were needed to obtain the results of a run. The program, block count, queue and storage (facility) printout and histograms comprised the printout.

FIGURE 1

SYMBOL	BLOCK	FIELD	COMMENTS
AAA	ASSIGN TRANSFER	1-,5	Process 58
BBB	ASSIGN TRANSFER	1-,3 ,LOOP	Process 24,28,133
CCC	ASSIGN TRANSFER	1-,11 ,LOOP	Process 15
DDD	TRANSFER	.200,NEXT,ELMTE	Process 20
EEE	TRANSFER	.200,NEXT,LOOPN	Process 101
FFF	TRANSFER	.200,NEXT,LOOPN	Process 17
GGG	TRANSFER	.200,NEXT,LOOPN	Process 10

SIMULATION RESULTS

The information that is provided now became known by formulating the simulator. Much of it would not have come to mind until the project was complete if the simulator were not constructed. Some would not be known at all.

Since processes change to the extent that they are eliminated and replaced, the sequence is heavily impacted and also changes rapidly. Rework sequences are a special problem. Reworks are necessary to conserve development gains. Changing processes and main sequences cause a heightened sensitivity to rework activity, thus rework sequences are prominent and highly variable to the extent that every process may require its own.

Scrap yields are difficult to determine during the initial development stages. The line development usually progresses from front to back. Overlapping the sequence (using the same process station for different phases of development or using a similar technique that exists at the beginning of the line) provides some quickening (and economy) of the development line structuring. Yields for the front of the line may or may not provide any insight as to yields at the end of the line. At any rate, the yield figures become more accurate as more parts are introduced into the line and as the development line progresses in becoming successful. Successful yields (completed parts) do not occur until late in the development period.

Rework yields for individual processes are maximized at the beginning of the development task because parts have to be reworked to acquire some knowledge and economy for the developing processes. The front end of the line develops first in a complex sequential process primarily because yields are not sufficient for a time to provide a testing and sampling of parts near the middle or end of the line. Simulated parts may be utilized to acquaint personnel with procedures and to solve the more apparent problems; however, the parts are usually of such complexity that hardware simulation leaves many problems unsolved. Rework yields are, therefore, as unstable as scrap yields. Figures become progressively accurate as the development task becomes successful; knowledge of yields progresses from the beginning to the end of the line.

Because emphasis is placed upon providing successful processing, record keeping may be unintentionally neglected especially if developmental time restrictions are severe. Records are usually kept but they are non-systematic, incomplete, difficult to understand and in such large amounts as to defy association and assimilation for total system study.

Process equipment is scattered through various laboratories. Little is known about batching capabilities if batching is at all possible. The batching may be variable. Some equipment is unique while other equipment is duplicated. Some equipment is temporary. Personnel and equipment are not necessarily correlated. Any of the development personnel may use any equipment. The equipment is usually operated manually to some degree and limited personnel may be available. All these variables pose queueing problems when simulating.

Other information for constructing the simulator was somewhat inadequate. Scrap yields were available only a few months prior to simulating whereas the development line was in existence for a much longer period of time. The scrap yields were not nicely defined and consisted of different sets totaled together. One set followed one sequence, etc. Rework sequences were non-existent. Main sequences were defined for the near immediate situation. Changes were not transmitted rapidly to paper for easy access.

Rework sequence records were non-existent and were acquired from individuals who remembered such events. Much of this information was very prone to errors. Rework yield figures were not available.

Process time periods were estimations given by process engineers and were not explicitly defined to eliminate misunderstanding. Batch processing versus serial processing was largely a mystery because detailed time records were not kept.

Records of equipment usage were not available. The correlation of parts processed versus equipment was non-existent. Batching was performed but was not recorded. Personnel correlation and equipment usage was not available.

Well documented input rates were not available.

The preceding answers became known by constructing the simulator. Answers are now given as a result of operating the simulator.

The simulator was run initially with seventy-five parts spaced at eighty minute intervals. Scrappage was total as was anticipated. One hundred and fifty parts were cycled into the simulator and final scrappage occurred relatively close to the same process and/ or equipment where final scrappage previously occurred. An unlimited number of parts were introduced into the simulator to determine the length of time for a part to be completed.

Although ficticious yields (20%) were utilized for reworking, a good correlation existed between simulator scrappage and actual scrappage. Values agreed within 20%, sometimes within 10%.

Credence on the length of time for a completed part was tempered by the fact that scrap yields for the final half of the line were unknown. The time period is probably longer than the simulator predicts.

The simulator is structured to enhance the study of hardware developments. This situation implies operation of the simulator at successive time intervals of development. Since emphasis is not placed on rates of completed parts but on development periods preceding completion, the structure places emphasis on individual developments.

The simulator can determine if developmental situations are uniform or nonuniform in time with different people participating. If uniformity prevails in the development line (from station to station), better predictions of time can be made for planning purposes especially if part of the total product development has been accomplished.

For example, assume that many separate product development phases are required for a total product development. Assume also that a separate phase is a function of several variables (manpower, rework, and scrappage). An argument may be advanced that each separate development phase has its own unique problems and unique manpower such that no uniformity of time exists from situation to situation.

A counter argument may be advanced that all developmental personnel are generally equally educated (i.e., each individual has as much knowledge of his specialty as another has in his different specialty) and the state of the art of each specialty is generally equal (otherwise it would not be selected or more personnel be assigned initially, etc.) such that time uniformity between phases does exist.

The simulator was used to examine some processes in front of the manual development line and evidence of uniformity from station to station was present. A histogram of scrappage versus simulator containment time produced a relatively flat distribution. This fact enhances the supposition that development time periods (for success) for individual processes is uniform such that accurate predictions can be made by simulating the line prior to completing that line (see Figure 2).

The preceding information became known by operating the simulator. It is now in order to describe the manner in which this knowledge is to be utilized for future product development.

First, records should be kept for rework to enable the determination of yields. Because rework is so prominent in a line of this nature, the records must be systematic in efficient form (or coding) such that logical information can be rapidly acquired from the large amounts of data. It is advisable, because of the availability of time shared systems at this locale, to enter the data into the terminal system because it forces a certain amount of efficient organization. However, designed efficiency is also necessary.

The number of reworks for a particular process station, if recorded, provides an indication of more troublesome processes and may provide indications that the process is evolutionary or revolutionary. Decisions on terminating or continuing pursuit can be made.

Rework yield figures are necessary for proper simulation. Proper simulation in turn determines accurate durations for sequencing. Accurate yield estimations can be made for the untried development processes. Management decisions to proceed with the same developments, manpower and particular emphasis will be afforded a bias by the simulator.

The batch sizes and time to process each batch, number of personnel handling the equipment, and the amount of usage of each equipment should be logged as well as the part number for which it is used. To accomplish this, the equipment should be coded by laboratory number and/or code defining each device. The nature of the equipment and the manner of usage by personnel (i.e., Equipment is not necessarily related to one specific process) require that a characteristic marking that is brief but associative, be used. New equipment should be related to equipment it has replaced and the date of installation should be marked on the device. Efficient records should be kept of replacement or elimination of all equipment.

The dating information may be absolute or relative. Relative dating (related to beginning of development) is more compact and systematic but confusing to those accustomed to calendar dating. Calendar dating can be performed with a conversion ability to or inclusion of relative dating. A common point of time reference aids in associating and assimilating data. The dating method should include a means for handling overtime and weekend labor.

A history of sequences should be maintained and dated. A simple number-letter code is desirable if practical. Differences between successive sequences should be recorded with dates. Reasons for altering the sequence should be filed. Reasons should be of a general nature if possible by using multiple choice selection by the process engineer. Loss of preciseness is offset by the fact that many unique reasons although logged deter and frustrate assimilation and association because of the large amounts of data. The process engineer can keep unique records if desired.

A process should be coded as to its entry and exit into the development sequences. Its counterpart, if any, should be logged.

Part entries into process stations should be recorded. Scrap parts and rework parts might be individually coded in the same entry as total parts.

Many of the preceding codes can be numbers with many digits in which different digit positions signify different data. The information can be placed in matrix form for data manipulation.

CONCLUSION

To summarize, simulation of the manual development line, although incomplete, provided knowledge about attacking the problem of computer technology development. It helps provide an association and assimilation of development data. It also aids in characterizing data and attests to the lack of some essential data. Operation of the simulator provided some confidence that it is useful for predictive purposes if more complete input data is available for simulation.

Another simulator was developed for a pilot manufacturing line for the same computer technology. The development period of this simulator overlapped that of the simulator described in this report. Both developments were performed by the author, and therefore, utilized an interchange of program format. Many ideas derived at in building the manual development line simulator were included as part of the pilot line simulator. Much information from the pilot line simulator, in turn, was instrumental in the production of a linear programming optimization model by the author. The cost objective function formatting and much of the constraint equation formatting was made possible by the information generated in constructing and operating the two simulators.

SYMBOL	BLOCK	FIELD	COMMENTS
AAA	ASSIGN TRANSFER	1-,5 ,LOOP	Process 58
BBB	ASSIGN TRANSFER	1-,3 , LOOP	Process 24, 28, 133
CCC	ASSIGN TRANSFER	1-,11 , LOOP	Process 15
DDD	TRANSFER	.200, NEXT, ELMTE	Process 20
EEE	TRANSFER	.200, NEXT, LOOPN	Process 101
FFF	TRANSFER	.200, NEXT, LOOPN	Process 17
GGG	TRANSFER	.200, NEXT, LOOPN	Process 10

Fig. 1

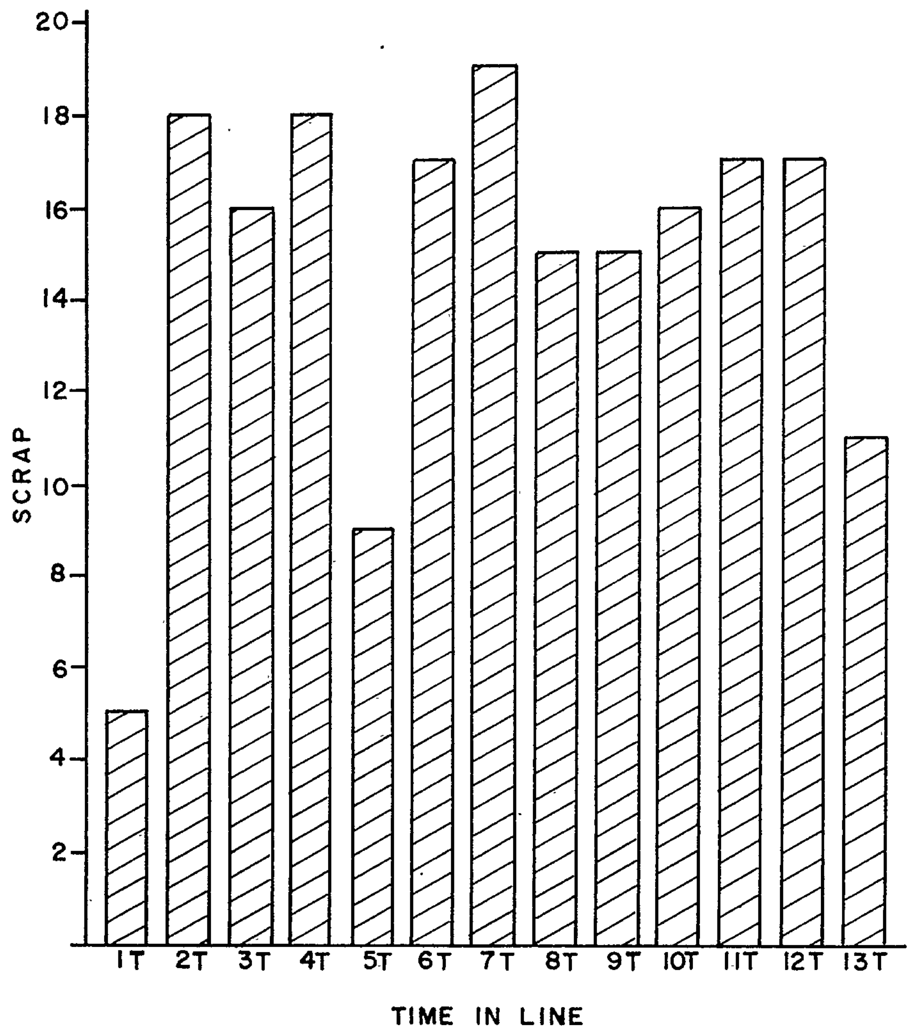

Fig. 2