Management Services: A Magazine of Planning, Systems, and Controls

Volume 5 | Number 6 Article 1

11-1968

Letters

Gerald M. Levinson

William S. Kamin

Howard M. Carlisle

Benny R. Copeland

Follow this and additional works at: https://egrove.olemiss.edu/mgmtservices

Part of the Accounting Commons

Recommended Citation

Levinson, Gerald M.; Kamin, William S.; Carlisle, Howard M.; and Copeland, Benny R. (1968) "Letters," *Management Services: A Magazine of Planning, Systems, and Controls*: Vol. 5: No. 6, Article 1.
Available at: https://egrove.olemiss.edu/mgmtservices/vol5/iss6/1

This Article is brought to you for free and open access by the Archival Digital Accounting Collection at eGrove. It has been accepted for inclusion in Management Services: A Magazine of Planning, Systems, and Controls by an authorized editor of eGrove. For more information, please contact egrove@olemiss.edu.

COMMENTS LETTERS CRITICISMS

Matrix accounting

I have read MANAGEMENT SERVICES' recent article by R. L. Mathews ("A Computer Programming Approach to the Design of Accounting Systems," July-August '68, p. 32) with a great deal of interest . . . I believe that a short dissertation should be made regarding high-level languages such as FORTRAN or COBOL.

These languages are problem-oriented languages; that is, they are designed to accept terms of a problem or routine expressed in English and mathematical syntax.

In order to produce such a language a voluminous software package must be programed. This is known as a compiler. In a compiler one COBOL statement is equal to many basic machine-language instructions, and therein lies the basic argument against the author's method.

The matrix, as described by the author, is accomplished by FORTRAN through the use of the dimension instruction set. This can also be accomplished in COBOL by use of the subscripting set.

However, in a production environment this is not the pragmatic approach for a highly complex financial system. Firstly, the COBOL and FORTRAN compilers utilize a great deal of storage in themselves; secondly, the two or three necessary high-level instruction sets mentioned above, which look innocent and simple to the uninitiated, in reality generate numerous machine-language instructions which in turn create excessive use of the computer's central processor; and, finally, there is no guarantee that all the boxes (used to accumulate data) in the matrix will be utilized.

The use of FORTRAN in a commercial system is also somewhat questionable. FORTRAN was designed and is used primarily for scientific and mathematical problems. COBOL, on the other hand, was designed for commercial purposes specifically. Recent surveys . . . indicate that almost 70 per cent of applications written are in COBOL and that the rate is increasing each year. This is because of the ease with which it can be learned (approximately two weeks in most instances) and its support by national (CODYSYL and USASII) and international (ECMA) committees and the various user exchanges.

I would also like to point out that in the author's array he is providing for 88 (11 columns x 8 rows) accumulators. By inspection of those accumulators used, we can see that only 25 per cent are utilized. Therefore, if we lose 75 per cent utilization of eight accounts, how much would we lose if the theory were applied to a chart of accounts containing hundreds or thousands of accounts such as many businesses and corporations employ today?

A basic procedural design of the traditional double entry logic calls for only two table look-ups (a debit record file and a credit record file) with storage facilities. This method encompasses 16 accumulators which can be utilized almost 100 per cent of the time.

With regard to optimization of computer equipment, I would like to point out that the subject FORTRAN program was written in 1966 on second generation hardware and software, when separate although related accounting routines were necessary. Since then there have been many developments, such as implementation of the integrated data base concept, modular memory-utilizing drums and disks, faster instruction and data-handling speeds, and advanced buffering and sorting techniques which optimize core and peripheral storage. Therefore, with today's trend toward mass-storage techniques the implementation of the traditional double entry logic, though still somewhat restricted, is more easily . . . handled.

In conclusion, "Matrix Accounting" certainly makes an effective presentation—especially to nonaccountants. It would be particularly powerful if applied to the traditional application of funds or any report showing cause and effect, i.e., the structure of the American economy.

Since the apparent attributes are worth pursuing, I would like to take exception to the method described in the article and suggest another path.

To utilize subscripting and/or dimension subset for processing data and displaying the results leaves a great deal to be desired, especially in processing time. It is suggested that the program for creating a matrix display should be a separate COBOL display or print routine not utilizing the subscripting technique. This print program should then interface with the financial data base and utilize its information for inclusion in the matrix.

In addition, practitioners concerned with comprehensive information systems should be aware of the software state of the art and standardize the language vehicle before making any basic policy decisions concerning programing procedure.

Gerald M. Levinson, CPA
Manager, Information Processing
UNIVAC *Division of Sperry Rand Corporation*
Philadelphia, Pennsylvania

At a loss

In the article by Mr. Carlisle in your July-August, 1967, issue ("Systems Approach to Integrating Cost and Technical Data," p. 34), the author states on page 38, with reference to Exhibit 3 on page 41, "It will be noted that for all four motors this effort was completed approximately ten days later than the original plan. This gives a good indication of the technical progress under the program as of this date."

Unfortunately, I can find nothing in Exhibit 3 which indicates actual technical progress to date, except the circled numbers indicating rocket motors completed. Is that all there is to it? How does he know they were completed ten days late? Since the utilization of this report seems to be at the heart of the IMPACT system, I'm at a loss to understand how to use it.

William S. Kamin, Controller
Standard Fruit Company
La Ceiba, Honduras

Several methods

In relation to the question . . . raised regarding the determination of technical progress or physical completion, there are several methods for doing this. The method I utilized here was to list below the planned technical progress line the actual technical progress in the same manner that costs are treated on the lower half of the chart. Thus, opposite the subprogram "design" there are two sets of 2's, the lower set being encircled. The circles indicate that this portion of the work has been accomplished. Relating the circled 2's to those indicating the planned schedule shows that each set slipped schedule by about ten days during January and during February.

Another method can be demonstrated on Exhibit 1 by utilizing the Key Program Milestone line. Under this method, as milestones are accomplished, this is indicated by showing an inverted triangle below the line at the date the milestone was completed. The difference in spacing on the line between the planned date for completion of the milestone (triangle above the line) and the actual completion date (inverted triangle below the line) indicates whether the project is ahead of or behind schedule.

Another common method of portraying technical completion is to weight milestones or tasks within a project and, based on this weighting, determine what per cent of the program each increment represents. Based on the planned completion of tasks and milestones, lines are presented on the program chart indicating the planned schedule. As tasks are actually accomplished, the percentage completion is determined from the weightings, and parallel colored lines are shown below the planned lines on the program schedule in Exhibit 3. As an example, if the reporting date is as of April, 1966, it is possible to determined how far ahead or behind schedule you are depending on where the colored "actual" completion line stops on the chart.

Howard M. Carlisle, Head
Department of Business Administration
Utah State University
Logan, Utah

Correction

[In my article, "Statistical Decision Theory" (M/S May-June '68, p. 45)] the very first matrix (p. 50) has a typographical error. The Demand column should read:

Demand (D)	corrected
0	
1	corrected
2	
3	

Benny R. Copeland, CPA
Associate Professor of Business Administration
North Texas State University
Denton, Texas