

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

Volume 6 | Number 2

Article 1

3-1969

Letters

Richard J. Pegis

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



Part of the [Accounting Commons](#)

Recommended Citation

Pegis, Richard J. (1969) "Letters," *Management Services: A Magazine of Planning, Systems, and Controls*: Vol. 6: No. 2, Article 1.

Available at: <https://egrove.olemiss.edu/mgmtservices/vol6/iss2/1>

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.



Crucial points

I have read both R. L. Mathews' article ("A Computer Programming Approach to the Design of Accounting Systems," M/S July-August '68, p. 32) and G. M. Levinson's answering letter (M/S November-December '68, p. 2). Their discussion raises two crucial points.

The first deals with cost/effectiveness of different programming techniques. The argument is not really between COBOL and FORTRAN but between programs written in higher-level (compiler) languages and the far more machine-oriented assembly languages which bear essentially a one-to-one relation with the machine instruction set.

Certainly, to use a program on a production basis for a large part of available computer time, running-time efficiency must be the major concern. Running-time efficiency can be heavily hardware-dependent, as in situations where inadequate core storage is available and

excessive disk access must be used instead. In these situations the heavy investment in programmer time and debug time is justified from a cost/effectiveness point of view.

However, most programs do not fall into this category. Very few programs start in this category. Those programs that do fall into this category usually do so as they evolve, growing slowly from experimental programs. For this type of program the compilers, even if they generate less efficient codes (i.e., more instructions*) than a good programmer working in assembly language would, would still be cost/effective because the programs can be written and debugged much faster by less skilled programmers.

The proper time to write the machine code is when the program is fully debugged and its form frozen. Then the program should be carefully analyzed for time consumption and the necessary portions of the program time-optimized. Thus, in the early stages of program trial and design, ease of manipulation and analysis far outweigh running time and core

*Mr. Levinson might give the wrong impression when he states that "compilers utilize a great deal of storage in themselves." This storage is used only during the actual compile, not during program execution.

storage considerations from the overall point of view of cost/effectiveness, and in this stage the matrix approach may permit great savings of time and money.

The second point raised by the article and its discussion deals with the overall approach to computers. To date, the vast preponderance of commercial computer utilization has been for doing what we've always done, only faster and, we hope, at a lower cost. However, with the availability of third generation hardware and the increasingly easy access to time sharing services, it is vital that we move beyond this stage.

The time has come when it is no longer sufficient to ask, "Can the computer do this job better?" If we are to stay competitive and use all of the management tools available to us, we must start asking, "What problems can the computer solve for us which we have been unable to solve before?"

We must consider not just what functions the computer can take over but what problems the computer can solve—in forecasting, marketing, decision making, product design, etc. It is only with this approach that management can fully utilize the economic benefits inherent in computer technology.

Richard J. Pegis
Optimization Associates, Inc.
Rochester, New York