

## Preface

Program development and programmer productivity are receiving much attention as a rapidly expanding technology dramatically increases available computing power. The challenge is to take advantage of this capacity with new applications, developed in a most efficient and effective manner. Thus, the work of the programmer is under close scrutiny.

In this issue, the Journal emphasizes the programmer and his work. Presented are a study of measurement standards for programs and programmers, a technique for evaluating program efficiency, and an essay on the work environment of the programmer. Also discussed are some developments that facilitate the use of mixed-integer programming in solving large management science problems.

The essay on the programmer's work environment is presented under a new format for the Journal. By definition, an essay is an analytical literary composition dealing with its subject from a personal point of view. Through this medium, the Journal will, from time to time, present what we believe to be thought-provoking topics of particular interest to the data processing community. It is our pleasure to have Gerald McCue initiate this new format. McCue, a principal in the San Francisco architectural firm of MBT Associates, was responsible for designing the IBM Santa Teresa Laboratory in San Jose, California, a facility planned specifically for the activities involved in program development. McCue was commissioned to create an environment that would meet the special needs of the programmer and enhance programmer productivity. In his essay, he discusses those needs and how they were used in the design process.

A programmer needs tools to measure the efficiency of a program; the paper by deFreitas and Lavelle discusses a technique for investigating the efficiency of compiled programs. A time analysis, by providing additional information to the compiled listing of a program, shows the programmer where efficiencies may be increased in a program. This technique can also aid the programmer in debugging.

Recently, a study group set up by the IBM San Jose Programming Center explored the interrelated topics of program quality,

programmer productivity, and the units of measure that could clearly display trends in these areas. Some of the findings of this group are discussed in the paper by Jones. For example, several commonly used units of measure can give paradoxical results: Units such as lines of code per programmer month and cost per defect tend to favor less efficient programs. Jones presents several other units of measure and looks at program quality with measures of defect removal efficiency and defect prevention. Programming productivity measures are broken into work and cost units. The author notes that while progress has been made in programming measurement, much remains to be done in this field.

Management science problems such as optimization have been resolved with only limited success because there have been too few easily used computational tools. To solve these problems, a new extended control language used with MPSX/370 facilitates the use of programming techniques such as mixed integer programming. The language permits easy access to the base elements of the problem and allows recursive processing, enabling the programmer to break the problem down to manageable size and solve it by iterative processes. The paper by Slate and Spielberg discusses this new control language and a number of base applications.

The final paper, by Chen, Crowder, and Johnson, illustrates a use of mixed-integer programming to solve the problem of installation scheduling. Large terminal network installations can be optimized for some measurement, such as capital investment, by using this approach.

We are grateful to Arthur J. Stein and Arthur Appel, members of the IBM Research staff at Yorktown Heights, whose expertise in computer graphics has been invaluable in developing covers for this and the previous issue of the Journal.

Connie A. Thiel  
Editor