Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Précis—User-Oriented Computer Modeling Environments

Christopher E. Nugent, Thomas E. Vollmann, A. Harold Howell, Robert L. Schuldenfrei, James R. Westfall,

MANAGEMENT SCIENCE
Vol. 17, No. 5, January, 1971
Printed in U.S.A.

Précis

CAPITAL BUDGETING/PROJECT SELECTION BY MATHEMATICAL PROGRAMMING: AN ANNOTATED BIBLIOGRAPHY

ROY B. LARSON

Case Western Reserve University

A comprehensive bibliography on the application of mathematical programming to the capital budgeting/project selection problem is presented. Over 150 items are cited with a brief summary given of the relevant contents of each item. Critical comments concerning problem formulation, conclusions, and/or computational feasibility are offered for many entries. Where alternate sources represent substantially identical work, an attempt has been made to cite the most widely available source as the primary entry with alternate references indicated in footnotes to the review of the primary entry.

The 116 primary entries are arranged chronologically from 1955 through 1969 with alphabetically sequenced author and title cross indices given as appendices. The potential value of the bibliography to research workers in the field is enhanced further by the presence for each primary entry of a list of all other entries of the bibliography which are referenced by the primary entry at hand. Further, each primary entry is also accompanied by a citation index which indicates all other entries of the bibliography which refer to the primary entry at hand or to one of its listed alternates.

Readers interested in obtaining a copy of the complete paper should request a copy of Technical Memorandum No. 173 from:

Roy. B. Larson
Operations Research Department
Case Western Reserve University
University Circle
Cleveland, Ohio 44106

USER-ORIENTED COMPUTER MODELING ENVIRONMENTS

CHRISTOPHER E. NUGENT, THOMAS E. VOLLMANN, A. HAROLD HOWELL, ROBERT L. SCHULDENFREI AND JAMES R. WESTFALL

Evolution in models for problem solving is inevitable and desirable; as goals are achieved (or problems solved) new, more fundamental or telling, problems are identified leading to model revisions and better problem solutions. The speed of the evolution is dependent on two major factors: the inherent flexibility of the model and relationship of the model user to the model designer. This paper focuses on these two factors in the modeling environment. It is critical to note that the direction or course of the evolution can only be determined by users since they have the relevant goals, objectives, or criteria; expert model builders are much more capable of problem solution than of problem formulation or identification.

In considering computer models, the existence of a gap of understanding, appreciation and knowledge between model designers and model users in traditional computer system design is drawn and the costs of that gap are discussed. The authors believe that modern computer technology can make possible a shift in the role of designers and users in which the designer's task is to create the proper computer environment for the user to design iteratively his own individualized computer models.

One might reasonably argue that such models could be much more easily implemented by a model building specialist. Such an argument would be valid if related to only one particular change; but we are not focussing our attention on only one change, rather we are concerned with the process of change or model evolution that may in fact be virtually endless. The process of change can only be directed by the user (who has the criteria); and the major goal is to speed the change process.

For practitioners unskilled in computer technology to use a computer effectively, a man-machine communication system is required more geared to the user's everyday communication skills than are most computer systems today. Key to implementation of the user-oriented concept is the ability of nonprofessionals to make major changes in their computer models.

A highly flexible user-oriented computer language allowing users to assume more of the system design function and obtain individualized massive computation power with a minimal learning investment has been developed. The language is oriented toward data-based applications and achieves its user-oriented flexibility by being written in a higher level language. (Versions exist in BASIC and FORTRAN.)

A preliminary computer system was designed for use in medical research by doctors at a college health clinic. For data, 26 facts for each of 50 mononucleosis patients were stored. The facts stored included the day, month and year of a mononucleosis patient's admittance to the hospital, his doctor, age, sex, race, *etc.*, as well as results of medical tests reporting white blood count, percentage of lymphocytes in blood test, heterophile dilution, beef absorption, *etc.* Each fact was given an alphabetic code (*e.g.*, WBC for white blood count, AGE for age, BEEF for beef absorption), each patient a numeric code. Nine commands were implemented: one for retrieval (RET), four for limited manipulation (ADD, SUB, MPY, DIV), one for output (PRT), one for editing (LIM), one for grouping (EQU), and one for terminating execution. Use of the system at the beginner's level requires very little computer sophistication. In about one hour physicians with no prior computer experience have been taught the elementary commands of the language and have used it to assess existing data bases. Use takes a conversational form between man and machine in which the man presents one command at a time for execution and the system indicates its readiness to accept the next command by printing a question mark.

The language has been written in a highly modular fashion and provision has been made for easy user revision of certain modules likely to change, particularly the data base itself and the parameters that describe the contents of the data base. The user at this level does not need to know the systems programming language. Physicians with no programming knowledge of Dartmouth BASIC have been taught to change the environmental and data sections of the program in about 2 hours.

The next level of use requires some knowledge of the systems programming language. But with this knowledge the user can if desired completely redesign the system to fit his own particular needs. This takes longer to learn, the actual time being a function of the user's programming ability and the magnitude of the change that he wants to make. A good BASIC programmer might expect to take 5 to 10 hours to make a significant change in the program, a beginning programmer 10 to 20 hours. Ease of changing the computer model is enhanced by its modular construction. Any module can be altered fairly easily, and new modules, perhaps with calls on existing modules, can be added readily.

Use of the language by the doctors led to the generation and testing of many highly individualized hypotheses with resultant changes in the treatment of the disease, the data to be collected on patients afflicted with the disease, and in the procedures used for collecting data. In addition, hypotheses were generated that were interesting enough to call for more extensive data collection in other student health centers; it was expected that these hypotheses and subsequent tests might lead to publication in a medical journal.

Use of the language on other data bases has also been accomplished. Community planning (data base consisting of such facts as population, public dollars spent for education per school child, public dollars spent for welfare, automobile taxes, *etc.*), banking (data base comprised of daily bank financial statements), sociology (prototype system based on 1/10,000 sample of the 1960 population census), financial (Standard & Poor's Compustat Data), and political demographic data (Russett data bank of political and social indicators) represent some of the other applications; new ones are being continually added.

The paper discusses all of these applications and the computer modeling environment which fosters evolution in these systems.

Copies may be obtained by writing:

Thomas E. Vollmann
Amos Tuck School
Dartmouth College
Hanover, New Hampshire 03755