Listed are abstracts from recent papers and books by IBM authors. Inquiries should be directed to the publications and publishers cited.

Abstracts

An introduction to database systems, C. J. Date (IBM UK Lab, England), Addison-Wesley, Reading, Massachusetts (1975), 366 pp. This introductory text and reference is divided into five major parts: data base system architecture, the relational approach, the hierarchical approach, the network approach, and security and integrity. Part one provides a general introduction to the concepts of a data base system. The relational approach is primarily devoted to an exposition of the ideas of E. F. Codd. Despite the theoretical advantages of the relational approach, and its potential for efficient implementation, it has not yet been demonstrated in a full-scale system. The sections on the hierarchical approach are based entirely in IBM's Information Management System (IMS), because "it is a good example of the hierarchical approach and an important system in its own right." The network approach is based on the proposals of the Data Base Task Group of the CODASYL Programming Language Committee. Security and integrity are covered in two concluding chapters. Extensive references and bibliographies.

Automatic structuring of programs, G. Urschler (SDD Vienna, Austria), *IBM Journal of Research and Development* **19,** No. 2, 181–194 (March 1975). Described is a method that allows the translation of a traditionally written (unstructured) program into a set of top-down structured, semantically founded, GOTO-free modules. The method reveals not only the logic of a given program in a most natural way, but it also reduces code duplication to a minimum.

Material requirements planning, J. A. Orlicky (DPD White Plains, NY), McGraw-Hill, New York, New York (1975), 292 pp. New systems for inventory control and management allow users to reduce inventories while improving service. Formerly, the inventory control problem was perceived as being essentially mathematical, rather than one of massive data handling and data manipulation. Ordering average amounts, however, did not fit particular situations. The fact that the chronic problems of manufacturing inventory management are now being solved is due not to better mathematics but to better data processing. The three key functions of a material requirements planning (MRP) system are inventory control, priority planning, and providing a basis for capacity requirements planning. All MRP systems have a common objective, which is to determine (gross and net) requirements, i.e., discrete period-demands for each item of inventory, so as to be able to generate information needed for correct inventory order action. This action pertains to procurement (purchase orders) and to production (shop orders). MRP systems meet their objective by computing net requirements for each inventory item, time-phasing them, and determining their proper coverage. In addition to planned orders, the MRP system also reevaluates the timing of open orders relative to (possible changed) net requirements in the near future, and it signals the need for rescheduling these orders, forward or backward in time, as required, in order to realign coverage with net requirements.

Reliable software through composite design, G. J. Myers (SRI New York), Petrocelli Charter, New York, New York (1975), 156 pp. Composite design is a technique which reduces complexity in the design of highly modular programs. The author chose the term composite design to distinguish between this work and earlier efforts generically known as modular design.

The benefits of composite design result from its achieving high module independence. The resulting reduction in complexity has a positive effect on a program's quality, particularly in terms of reliability (number and significance of bugs), maintainability (cost of fixing a bug), and extensibility (cost of adding new function). Composite design also appears to reduce the cost of producing a program and the elapsed time of the project.

A highly modular design is achieved by maximizing the relationships between the elements of a module and minimizing the relationships between modules. Module coupling is used to minimize the relationships between modules. Module strength is used to maximize the relationships between elements in individual modules. The scale for coupling is inverse to the scale for strength.

A method called composite analysis is described, which turns a problem description into a program structure. The basis of composite analysis is that the structure of the program should resemble the structure of the problem.

Other topics include a chapter on management of design, including brief sections on structured programming, top-down development, and HIPO. Another chapter, on modularity and virtual storage, describes paging considerations and module packaging.

The new math of computer programming, H. D. Mills, (FSD Gaithersburg, MD), *Communications of the ACM* **18,** No. 1, 43–48 (January 1975). Structured programming has proved to be an important methodology for systematic program design and development. The author discusses structured programming in mathematical form to illustrate the relevance and power of classical mathematical concepts to simplify and describe programming objects and processes.