digital

DECsystem-10

DATA BASE MANAGEMENT SYSTEM

-DBMS-10-

Another step

DIGITAL now offers a software system to provide the data structures, access methods and program control features associated with advanced data management techniques. This system is another step toward giving DECsystem-10 users the software building blocks to generalize on-line oriented applications development and processing.

DATA BASE APPLICATIONS

Certain collections of data such as those occurring in commercial accounting, production inventory control and university administrative systems are utilized in computer applications which have functional relationships and common processing requirements with other applications. Many times, data file organizations are defined in one application or individual program process that effectively causes restructuring, redundant appearance and repetitive processing of the same data in another application.

"On-line" processes such as customer order entry, shipment planning and student information retrieval systems tend to utilize different data structures and access methods from the processes used to create and maintain primary data files. As new applications for existing data are determined and additional data items are defined, program development personnel are required to alter both existing data file forms and programs or create and maintain redundant copies of previously recorded data. To enable DECsystem-10 users to organize and maintain data in forms more suitable to the integration of a number of related but separate processes and applications, DIGITAL offers a Data Base Management System called DBMS-10. The DBMS-10 software is intended for use with applications systems where centralized definition and control of data processing and program development functions require data structures and processing techniques not satisfied by traditional data management facilities.

DBMS-10 FEATURES

The DECsystem-10 Data Base Management System is based directly upon the proposals of the CODASYL Data Base Task Group (DBTG) appearing in their report of April, 1971. DIGITAL's goal for this data base software is to provide the features which will assist users in obtaining the most significant objectives stated in the CODASYL DBTG report.

These features are summarized below:

- Hierarchical Data Structures
  In addition to sequential structures, both simple tree and more complex network structures can be created and maintained. Data items can be related within and between various levels of the structure established. Figure 1 illustrates these basic structures.
- Non-redundant Data Occurrence
  Data items may appear in a number of different structural relationships without requiring multiple copies of the data. Data structures may be established and modified in a manner most suitable to a given application without altering the occurrence of the data in structures maintained for other applications.
- Variety of Access and Search Strategies
  Data sets may be maintained in either chains or pointer arrays, as illustrated in Figure 2. Access may be through DIRECT, CALCULATED or VIA set location modes. Search keys in addition to the normal storage key may be defined.
- Concurrent Access
  Multiple run units (or programs) using the same reentrant code module further supplement the DECsystem-10 monitor's extensive centralized file handling capabilities. Any number of concurrent retrievals to the same data areas can be handled. Concurrent updates below the area level can be processed by a single program module referenced by other programs.
- Protection and Centralized Control
  Usage of a common data definition language establishes data base structures maintained on direct access

storage devices that are selectively referenced by individual application programs through privacy lock and key mechanisms. Physical placement of data is specified by a centrally controlled data definition processor.

- **Device Independence**
  The common input/output, buffering and control conventions of the DECsystem-10 monitor provides basic device independence. Applications programs deal with logical areas rather than physical devices. Data base areas may reside on the same or different direct access storage devices as non-data base files.
- **Program Independence of Data**
  Significant steps toward this goal are achieved by using the SCHEMA and SUB-SCHEMA concepts of data definition. Individual programs reference only user selected data elements rather than entire record formats. Program alterations due to adding new data and relationships are minimized. Alterations of the original form (i.e., binary, display) and element sizes still require program recompilations.
- **User Interaction Without Structural Maintenance Responsibilities**
  Individual users, applications and programs may access data structures without the responsibility of maintaining detailed linkage mechanisms that are internal to the data base software. Common and centralized authorization, error recovery and building techniques reduce individual activity to maintain data structures.
- **Multiple Language Usage**
  The DBMS-10 data manipulation language and pre-processor is oriented to the COBOL host language concept as described in CODASYL DBTG report. However, other procedure level language usages may be realistically considered with the later development of other pre-processors and compilers.

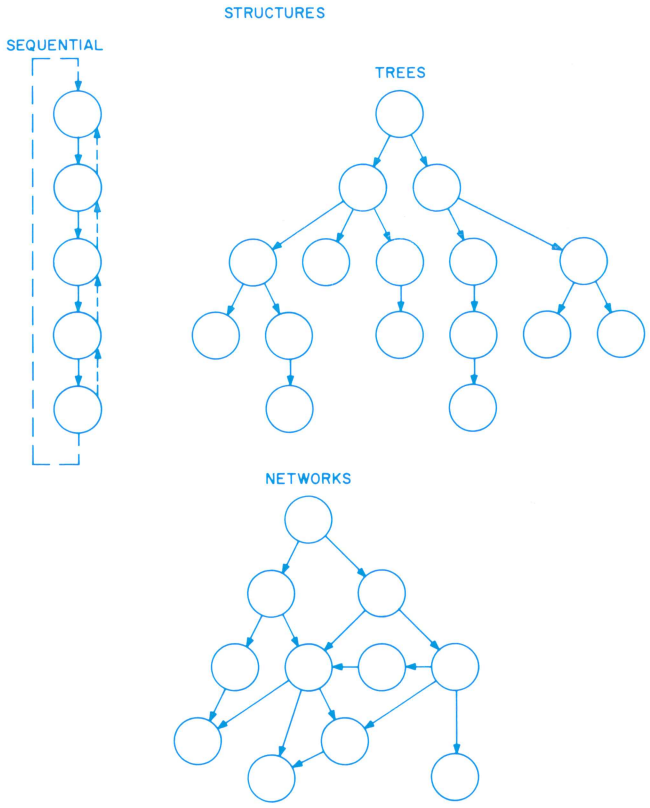

Figure 1

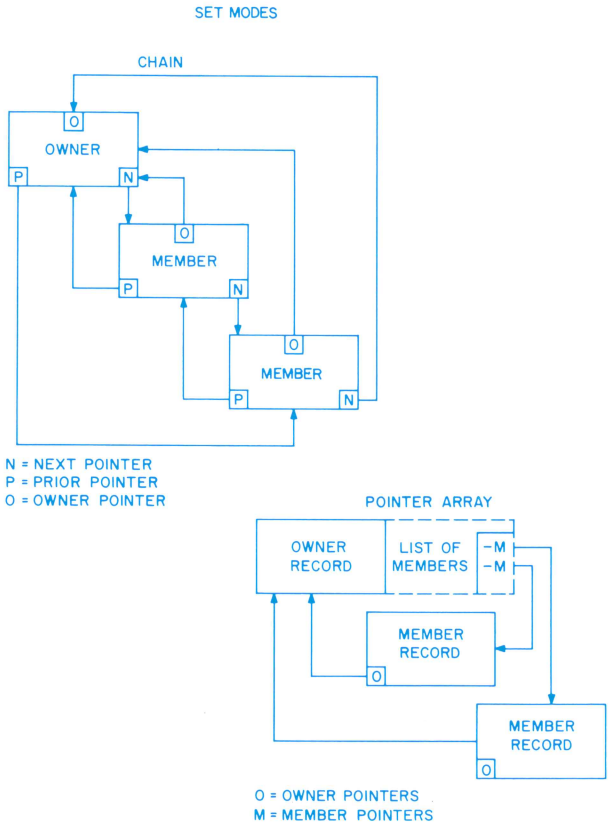

Figure 2

DBMS-10 SOFTWARE MODULES

Consistent with the CODASYL Data Base Task Group report, the body of DECsystem-10 data base management software includes:

- DDL—data description language and its processor
- DML—data manipulation language and COBOL pre-processor
- DBM—data base manager module of reentrant run time routines
- DBU—group of data base system support utilities

DATA DEFINITION PROCESS

Figure 3 illustrates the basic relationship of the data definition software module and its functions. The DDL processor includes the capabilities of a device media control language to accept and process statements allocating storage space, establishing page buffer sizes, record densities and maximum entry parameters of the data base. An installation may define any number of data bases according to processing relationships, storage requirements and scheduling considerations. A SCHEMA defines a data base. After the allocation parameters are established, statements giving a complete description of all the AREAS, RECORD descriptions SET relationships appearing within the data base are input to form a SCHEMA recorded on direct access storage. The purpose of the SCHEMA is to provide a single, centrally maintained description of the data to be referenced by any authorized application program. Once constructed, the description of the data need not be recreated by individual programs. The SCHEMA is expected to be built and maintained by the person functioning as the data base administrator. The DDL processor can accept placement control, privacy keys, record selection expressions, ordering schemes and search keys. Sequential, tree and network structures can be established with OWNER and MEMBER relationships maintained in either CHAIN or POINTER ARRAY form. Access of DIRECT, CALCULATION or VIA set location modes are included.

The last step in data base definition process is to develop a SUB-SCHEMA for the use of a specific application or system of programs. A SUB-SCHEMA enables an individual program to extract definitions of specific areas, records, sets and data elements from those maintained within a given SCHEMA. The authorization to access specific data base segments is recorded on direct access storage while SUB-SCHEMA statements are being processed. Any number of SUB-SCHEMAS may be defined within a SCHEMA such that only the particular items of interest to a given program need be explicitly referenced in the program itself. Basic independence of record item placement is obtained for the programs as data base elements, record types and set relationships are added to the data base for other SUB-SCHEMAS. Access authorization is maintained by privacy keys and external controls applied by the data base administrator.

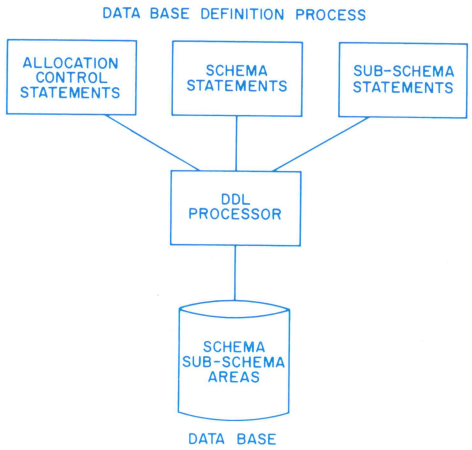

Figure 3

DATA MANIPULATION PROCEDURES

Data manipulation occurs through the use of DML commands embedded within a COBOL source program. Applications utilize descriptions and structural relationships already recorded in the data definition process to retrieve, alter, add and delete data and data relationships within a data base. Physical input/output and linkage aspects are handled by the run time data base manager module. The association of data areas and run units are established by DML OPEN and CLOSE commands. Temporary grant and release of restricted or exclusive access and update privileges occur through options with an open and close of data areas. STORE and DELETE commands are used to add and withdraw actual data into and from the data base. Set relationships are established and cleared with INSERT and REMOVE commands. FIND commands locate specific record occurrences, while GET commands fetch data from a current record into the program Users Work Area. Data is changed using the MODIFY command. IF and MOVE commands are used to test conditions unique to sets and to save current status indicators. In general, an application program concentrates on the processes to be performed rather than the establishment of unique file structures, input/output techniques and access modes.

The integrity of data relationships is maintained and enforced by internal software. For example, a single command such as REMOVE for a record within an extended hierarchy might cause multiple clearings of set relationships. Before any original data relationships are altered for such a REMOVE, the data manager module checks to prevent any improper operation. An access conflict anywhere in the structure prevents all modifications for this command and returns an appropriate error condition. Command subroutines can be invoked to deal with specific error status conditions by referencing USE clause procedures.

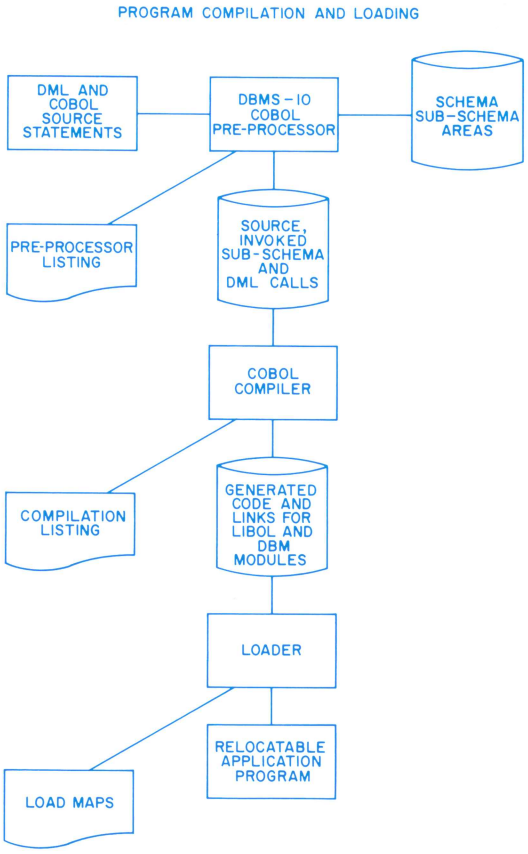

Figure 4

PROGRAM DEVELOPMENT PROCESS

Figure 4 illustrates the software modules involved in program compilation and loading. A complete source program containing both standard COBOL and DML statements is input to the DBMS-10 COBOL pre-processor. An INVOKE clause in the SCHEMA Section of DATA Division of the application source program causes the COBOL pre-processor to communicate with the appropriate SUB-SCHEMA. Elements appearing in the SUB-SCHEMA are then inserted into the program's User Working Area and made available for reference by DML commands. The DML commands them-

selves are translated by the pre-processor to form statements embedded in the modified COBOL source program output by the pre-processor. The COBOL compiler and Loader steps are completed to form a runable version of the program with appropriate linkages to execution time modules.

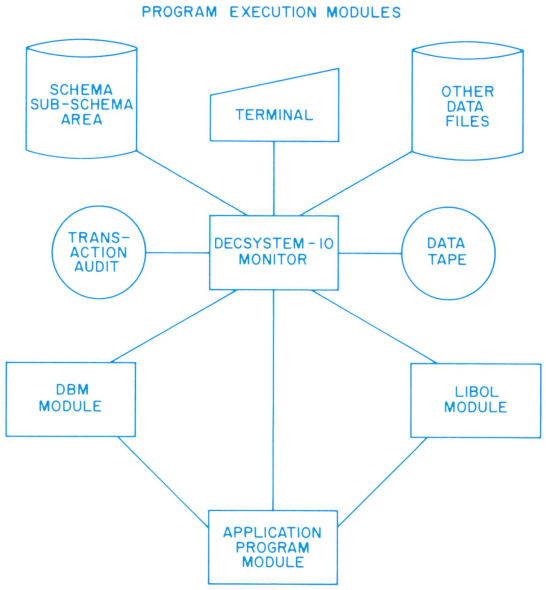

Figure 5

EXECUTION TIME PROCEDURES

Figure 5 illustrates the basic relationships between an application program, data storage and execution time modules. Access to all processor and storage facilities are routed through the DECsystem-10 monitor. Input/Output requests for non-data base direct access storage, magnetic tape peripherals and on-line terminals are processed through the COBOL run time module, LIBOL. The DBM, Data Base Manager is the run time module that links an application program through the monitor to the data base segments on direct access storage devices. Physical storage for data base areas and other types of data files may utilize either the same or different devices, at the administrator's option. Control and allocation of resources for multiple, concurrent applications using the various hardware devices are the functions of the reentrant run time modules and monitor. The file management, multiprogrammed run control and input/output features of the DECsystem-10 monitor are utilized in an unaltered manner for data base processes.

The extensive protection mechanisms, mass storage throughput optimization and physical units can further supplement the efforts of the design analyst and system administrator to implement efficient, secure application systems.

DBMS-10 SCOPE

The DECsystem-10 data base incorporates a significant portion of the features proposed in the CODASYL Data Base Task Group report. As a host language form, DBMS-10 is oriented to procedure level usage from programs as opposed to self-contained inquiry/response and report generation systems. Transaction processing and message control are related functions, but are not within the scope of the DBMS-10 software or the CODASYL DBTG report. Such functions remain within the province of applications programs or special processing modules. The form and major features of the data description language and the SCHEMA/SUB-SCHEMA concepts of the DBTG report have been incorporated in DBMS-10. The primary data definition features not currently specified in DBMS-10 are related to run time transformations of data items where item form and characteristics differ in SCHEMA and SUB-SCHEMA definitions. The form and functions of the primary DBTG data manipulation language are included in the DBMS-10 implementation of the DML commands. Only minor differences exist in optional formats and error code values associated with the DML commands. Refer to the DBMS-10 Functional Specification for detailed documentation of all data base features.

The DBMS software can be considered a body of generalized processing techniques around which specific data base oriented applications can be oriented. The form of this implementation allows analysts and programmers to express design parameters and data structures in the standard terminology of the CODASYL Data Base Task Group report. The usage of a common data definition language, processing module and overall data administration should enhance communications among installation personnel serving different applications areas. Longer term benefits may be derived from the study of the form, usage, and relatability of data among its various collectors, organizers, and users. The range and generality of the DBMS-10 features allow installations to select and analyze the processing techniques utilized in successive additions of applications and data modules to an increasingly refined data base. DBMS-10 should be viewed as a building block for more advanced implementations of a variety of application systems.

IMPLEMENTATION AND SUPPORT

DBMS-10 software in the initial release level includes the data definition language (DDL) processor, the COBOL preprocessor and the data base manager (DBM) module to support the data management language (DML) commands. All structural definitions are included except the pointer array set mode and search key features. Data base support utilities include initialization, print, data base clean-up and transaction audit routines. A second level release will support pointer arrays and search keys. Utilities to facilitate recovery processes and to analyze statistics gathered during DBMS-10 activities will also be added with the second release level.

The DBMS-10 software operates on standard release level of the DECsystem-10 monitor. No additional hardware facilities above those recommended for installations using COBOL language processing are required for the data base software.

Documentation includes a DBMS-10 Functional Specification and DBMS-10 User and Administrator Guides. Formalized training in Data Base Concepts, DBMS-10 Usage and DBMS-10 Administrator Techniques are provided at customer or DIGITAL sites. Software support is available on either a scheduled or user request basis. For additional information about DECsystem-10 data base management software, availability dates, lease prices and software support arrangements, contact the nearest DIGITAL Sales Office.

DBMS-10 HIGHLIGHTS

- CODASYL DBTG form of data definition and data manipulation languages
- Sequential, tree and complex networks supported
- Relative placement control for clustering records accessed in groups
- Reentrant data base manager run time module
- SCHEMA and SUB-SCHEMA definitions supported
- Both chain and pointer array set modes
- Temporary areas for developmental accessing without altering production areas
- Secondary search keys can be defined for specialized retrieval
- Residence of both data base areas and other files on same storage devices
- Removable areas and structures within a given data base
- Usage of both data base and traditional I/O operations in same program
- Variety of access modes including DIRECT, CALCULATION and VIA sets
- Variable number of record types, sets, areas and SUB-SCHEMAS within a SCHEMA
- Data bases may be restructured and expanded without altering programs
- Security to prevent inappropriate or unauthorized access to data base segments
- Operates in either batch or on-line terminal oriented mode

digital

DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts, Telephone: (617) 897-5111 • ARIZONA, Phoenix • CALIFORNIA, Sunnyvale, Santa Ana, Los Angeles, Oakland, San Diego and San Francisco (Mountain View) • COLORADO, Denver • CONNECTICUT, Meriden • DISTRICT OF COLUMBIA, Washington (Riverdale, Md.) • FLORIDA, Orlando • GEORGIA, Atlanta • ILLINOIS, Chicago • INDIANA, Indianapolis • LOUISIANA, New Orleans • MASSACHUSETTS, Cambridge and Waltham • MICHIGAN, Ann Arbor and Detroit (Southfield) • MINNESOTA, Minneapolis • MISSOURI, St. Louis • NEW JERSEY, Englewood, Metuchen, Parsippany and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Centereach (L.I.), Manhattan, Syracuse and Rochester • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland and Dayton • OKLAHOMA, Tulsa • OREGON, Portland • PENNSYLVANIA, Philadelphia and Pittsburgh • TENNESSEE, Knoxville • TEXAS, Dallas and Houston • UTAH, Salt Lake City • WASHINGTON, Seattle • WISCONSIN, Milwaukee • ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Melbourne, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BRAZIL, Rio de Janeiro, São Paulo and Porto Alegre • CANADA, Calgary, Alberta; Vancouver, British Columbia; Ottawa and Toronto, Ontario; and Montreal, Quebec • CHILE, Santiago • DENMARK, Copenhagen • FRANCE, Grenoble and Paris • GERMANY, Cologne, Hannover, Frankfurt, Munich and Stuttgart • INDIA, Bombay • ITALY, Milan • JAPAN, Tokyo • MEXICO, Mexico City • NETHERLANDS, The Hague • NEW ZEALAND, Auckland • NORWAY, Oslo • PHILIPPINES, Manila • PUERTO RICO, Miramar • SPAIN, Barcelona and Madrid • SWEDEN, Stockholm • SWITZERLAND, Geneva and Zurich • UNITED KINGDOM, Birmingham, Edinburgh, London, Manchester and Reading • VENEZUELA, Caracas

Printed in U.S.A. EDU 01172 2600 / R 03 05