



TEST GENERATION USING THE COBOL DATA BASE MANAGEMENT SYSTEM

by: Lucille H. McHenry
University of Mississippi
Student Chapter
Association for Computing Machinery

Test Generation Using the COBOL Data Base Management System

This project in data base management and design has two purposes. The first, and most important purpose, is to learn the fundamentals of data base development from the standpoint of the Data Base Administrator and those of data base usage from the programmer's viewpoint by using the COBOL language and the COBOL Data Base Management System. The second purpose is to develop a system to be used by the Computer Science Department of The University of Mississippi for computer-aided instruction.

Data Base Definition

The COBOL Data Base Management System (DBMS-10) allows the Data Base Administrator to structure and link data which are related to permit ease of accessing and referencing by the programmer without regard to the physical means of storage. DBMS-10 generally allows data to be structured in a manner most suited to each application, even when other applications may use some or all of the same data, and in such a way that data redundancy is not required. It allows retrieval of data by one program, or run-unit, while another is updating the data base. A variety of search strategies are provided and permitted for use against the entire data base or a part of it. DBMS-10 provides protection from unauthorized access of data and from the destructive interaction of programs. It allows device independence and the declaration of a variety of data structures ranging from those with no relationships between data-items to complicated networks. Finally, it allows the user, or programmer, to interact with the data without having to worry about maintaining the structural associations that have been defined.

DBMS-10 provides three languages for the design and usage of the data base. These are the Device Media Control Language, the Data Description Languages, and the Data Manipulation Language. Their functions include data base description, space allocation, and data transfer.

The Device Media Control Language (DMCL) entries are those which specify the maximum number of records to be stored per page, make area assignments to files, and establish the size allocations for these files. It was determined that 4 records would be placed on a page with each page having a size specification of 256 words. These specifications are based on the fact that 292 characters are needed to store just the data portion of the question records. In addition, each page header requires 4 words, each line header requires one word, and each set link associated with a particular record occurrence requires a pointer 1 word in length. The single area, QUESTION-ANSWER-AREA, was assigned to AREA1 and was allocated 300 pages of storage space for the storage of 1000 question records, 4 level records, and for the owner record of the set in which this entire data base is incorporated as well as for the DBMS-10 bookkeeping involved.

The DMCL is not a self-contained language but is an extension of the schema DDL, which will be discussed in the next section. The DMCL entries must pass through the DDL processor at the same time as the schema DDL.

The Data Description Languages (DDLs) are used to describe the logical and physical mappings of data in the data base. These mappings are listed in the file FORTST.DDL. This DDL describes the data mappings within the schema TESTER and the sub-schema TESQUS. There can only be 1 schema defined per data base. However, as many as 36 sub-schemas can be defined in the same data base. These mappings, or descriptions, define data-items, data-aggregates, records, sets and areas included in the data base. They also provide the relationships that must exist and be maintained between these items.

A data-item is the smallest named unit of data in the data base. Q-PER-LEVEL and LEVEL-NO are examples from this DDL.

Data-aggregates are named collections of data items within given records. A data-aggregate can be a one-dimensional vector or a repeating group which occurs an arbitrary number of times within a record. No data-aggregates were defined for this data base's DDL.

A named collection of data-items or data-aggregates is called a record. This is also the definition of a record type. In this data base there are three record types defined:

1. AREA-OWNER - This is the owner record of the set containing the owner records for all other records in the data base. There is 1 occurrence of this record type.
2. Q-LEVEL - There are 4 occurrences of this record type. Each serves as the owner record of the sets containing the different levels of questions.
3. Q-A-RECORD - These records contain the data used in test generation. The DMCL entries have allocated enough space for 1000 occurrences of this record type.

The location mode of each record type defined is CALC based on a particular field in each of the record types Q-LEVEL and Q-A-RECORD, using LEVEL-KEY and QUESTION-KEY, respectively. The location mode for AREA-OWNER is also defined to be CALC based on A-KEY. However, since there is only 1 occurrence of this record type, a search for the first occurrence of this record type will make it the current record in the area. The CALC keys for the other record types are used as sort keys within the sets in which they are defined to be members. The usage of these sort keys will assure the user that the desired record will be returned. DBMS-10 will find the nth occurrence of a record based on the CALC key.

Sets are named collections of record types which describe the logical relationships which exist between them. There is an owner record declared for each set defined. Each set type can have one or more member record types. The ordering of the record occurrences within each set occurrence is independent

of their physical storage locations.

There are 2 sets defined within sub-schema TESQUS. These are AREA-SET and Q-A-SET. AREA-SET, with owner record AREA-OWNER, sorts the member record types of Q-LEVEL in ascending order based on LEVEL-KEY. There is 1 occurrence of this set type. Within this set, the set Q-A-SET is defined. The record type Q-A-RECORD is defined as the member record type for this set. The occurrences of these record types are also sorted within each occurrence of this set type on the basis of QUESTION-KEY. They are also sorted in ascending order. Each occurrence of Q-LEVEL serves as an owner record for each occurrence of this set type. Each occurrence of a record in a set is linked to the next record in the same set occurrence and to the owner record of that set occurrence.

There is only 1 area defined within the sub-schema TESQUS, QUESTION-ANSWER-AREA. However, more may be defined within a data base if needed for the applications for which that data base is designed. The area contains occurrences of set and record types. How the area is opened determines whether or not other run-units can open the data base for RETRIEVAL or for PROTECTED-UPDATE. If opened for PROTECTED-UPDATE, other users can have access to the data base for RETRIEVAL only. If the area is open for RETRIEVAL, then other run-units can open the data base for RETRIEVAL or for PROTECTED-UPDATE. However, if the area is opened for EXCLUSIVE-UPDATE, no others can access the area in any way. The differences in EXCLUSIVE-UPDATE and PROTECTED-UPDATE is that during EXCLUSIVE-UPDATE, changes are made directly to the data base (.DBS file) whereas in PROTECTED-UPDATE, modifications are stored in a recovery file (.RCV) and are later incorporated into the data base when it is opened by another run-unit.

The schema consists of all the DDL entries which describe the data-items, data-aggregates, record types, set types, and areas. Sub-schemas select the areas, set types, record types, data-items, and data-aggregates from the schema. Schemas and sub-schemas are not data, but they describe data. Passing the schema through the DDL processor produced the files TESTER.SCH and AREA1.DBS. If any errors are encountered in the processing, a DDL error code will be given. These errors must be corrected and the DDL reprocessed before the .SCH and .DBS files can be used. The record types in the data base are described in Figure-1, and the set relationships are shown in Figure-2.

The sub-schema has a privacy lock defined for it. The correct key must be specified in an application program to match this lock before the schema and sub-schema can be invoked. This is to prevent unauthorized access of the data stored there.

The Data Manipulation Language (DML) of DBMS-10 is the language used in the run-units to access the data base. COBOL is the host language for the DML since COBOL allows the DML statements to coexist freely in the application programs. This produces a unified language. The DML statements are those which actually allow the user to have interface with the data base. DML statements are used to invoke the schema and the desired

Figure-1 Record Types

AREA-OWNER record

A-KEY

where A-KEY is used in storing the record

Q-LEVEL record

LEVEL-NO
Q-PER-LEVEL

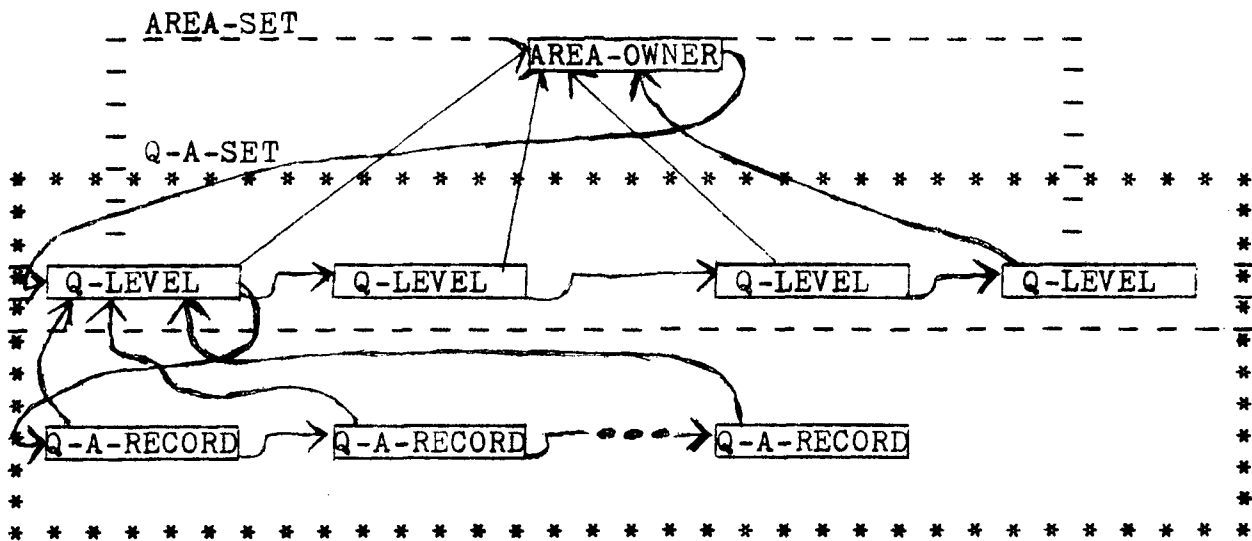
where LEVEL-NO gives the number of this record occurrence and Q-PER-LEVEL indicates the number of questions stored in this level

Q-A-RECORD

QUESTION-KEY
ANSWER
QUESTION

where QUESTION-KEY gives the occurrence number of this record type, ANSWER is the answer to the question stored here, and QUESTION contains the question

Figure-2 Set Relationships



sub-schema and then to open the desired area in the designated USAGE-MODE. There are many options available within the DML which are available to the user. The programs in the test generating system use versions of the DML FIND, STORE, GET, MODIFY, OPEN, and CLOSE commands. The DELETE command was not used since all the area which a record occupies is not released when a record is deleted. Due to the static nature of the space defined and the permanency of the data, the non-recovery of the total storage area could cause problems. However, provisions have been made to replace incorrect data with the corrected version of that data or with new data. This is done using the MODIFY command and is done without any loss of space.

The Operational Architecture of COBOL DBMS-10 is divided into the Data Base Control System, the User Working Area, and the run-unit. These concepts are basic to all DBMS-10 data bases.

The Data Base Control System (DBCS) is the object-time module for the DBMS-10. It controls data placement and access within the data base. It also provides the communication link between the data base and the application program through the association of monitor relative address blocks within the data base media with the data base page locations.

The DBCS has a set of error messages associated with it. If any of these errors are encountered, they are fatal and result from serious failure of DBMS-10. Correction of these errors can be as simple as adding more space to the data base or as complicated as a monitor I/O error.

The User Working Area (UWA) is the area where all data which the DBCS provides in response to a request is delivered and the area where all data that the DBCS will pick up for storage must be placed. Every program can have a UWA that contains a declaration of each record of the sub-schema currently invoked. The data in the program UWAs is not disturbed except when a DDL command is executed or by statements from the host language. The UWA locations are not necessarily contiguous.

The DBCS sets up the UWA on the basis of the particular sub-schema invoked. Each of the data-items are assigned a location here and can be referenced by its name. Data-items in the data base but not in the sub-schema invoked cannot be referenced. Since only 1 sub-schema has been defined for this data base, all the data-items are available for access each time the schema is invoked.

There are a number of locations which the DBCS provides in the UWA. These are Systems Communications Locations (SCL) and are used to provide interaction between the run-unit and DBMS-10. There are seven of these locations provided in which are stored the current record and area names, the error status, the set, area, and record names where the error occurred, and the error count.

The error status location, a field defined as ERROR-STATUS, of the SCL is the one most likely to be encountered by the programmer using data base. The ERROR-STATUS conditions are given

as a combination of a major and minor code in the format 9999 where the major code is the first 2 digits. If the leading digit of the major code is 0, it will be omitted.

The run-unit is simply the execution of a program. One or more program-units makes up a program. A program-unit is the smallest grouping of source language statements that can be compiled. Each of the programs used in the test generating system are composed of only 1 program-unit, the main program.

Data Base Usage in Test Generation

This data base has been developed specifically for the storage of questions, answers, and programming type problems concerning the Fortran programming language to be used in administering tests to students who are enrolled in this course. There are three COBOL programs in this test generating system: QSTOR, UPDATE, and WRITER. Each contains DML commands for use with the data base.

The program QSTOR was developed to load the data base. In it, the area AREA1 is opened for PROTECTED-UPDATE. The single occurrence of AREA-OWNER was stored during the initial execution. On each subsequent execution, this is bypassed on the basis of the user's responses to program prompts.

To store the initial questions in each level, the particular Q-LEVEL record occurrence had to be stored first. After this, the user could input the number of questions to be stored at that time in the level. After this the user could input the QUESTION-KEY, ANSWER, and QUESTION as they were requested. This storing routine is performed as many times as the number of questions he stated he wished to store. The field Q-PER-LEVEL in the Q-LEVEL record type keeps track of the number of questions currently stored in the level.

When the user wishes to re-enter a particular question level, he searches for the particular owner record desired. This record is displayed showing the LEVEL-NO and Q-PER-LEVEL. The number of questions the user wishes to store is requested as before, and the same storing routine is performed. Checks are made to assure that the 250 question per level limit is not exceeded.

The program UPDATE was developed for the user to use in checking specific questions in the data base and for replacing an incorrect or bad answer and/or question. Initially, the program searches for the occurrence of AREA-OWNER. Then the user responds to prompts with the level and question numbers desired. These are displayed to the user upon a correct FIND. Then the answer and question are displayed. The user can then indicate whether he wishes to continue or to alter the answer and/or question. If he desires to do modification, he then indicates whether he wishes to replace the answer or question or both. He will replace these as a response to a prompt. When the user has scanned as many questions as he desires, he indicates he wishes to exit the program and closes the data base.

The program WRITER in the test generating system is used

to retrieve questions from the different levels and to print them with the answer, the question number within the level, and the level number. Once again AREA1 is opened, and a search is made for the AREA-OWNER record. The user then inputs the level number and the number of questions desired from that level. The questions are selected at random from within the level. The user inputs the value of Y which is used to initialize the random number generator. The initial Y value must be an odd integer between 1 and 67108863 inclusive.

The algorithm used for generation of pseudo-random numbers in the open interval 1 to 250 is a modified version of Algorithm 266 by Pike and Hill which is listed in Communications of the ACM, volume 8, number 10, October, 1965. This algorithm states that it will generate a pseudo-random number in the open interval (a,b) where a is less than b.

The value of Y is not changed between successive performances of the GENERATOR paragraph so that a correct sequence will be generated. The modified algorithm is as follows:

```

real procedure RANDOM (A,B,Y);
  real A,B;  integer Y;
begin
  Y:=3125*Y; Y:=Y-(Y/67108864)*67108864;
  RANDOM:=Y-((Y/251)*251);
end RANDOM;

```

After the questions are retrieved, they are unstrung into a multiple choice question format.

After questions from a level have been found and output, the user has the option of returning to the data base for questions from any of the levels or of exiting. If he chooses to continue, the initial value of Y for the random number generator will be the last value of Y that was determined. The procedure then repeats as before.

A slightly different procedure is followed to obtain questions from the fourth level. Currently, there are 30 problems stored there which ask the student to write a program or program segment. Due to this, a slightly different version of the above algorithm is used to obtain questions from this level. At this time, all these problems relate to the materials covered for level 3 questions. However, there is area available to store programming questions and assignments for the other 2 levels. Schemes for storing and retrieving these questions based on different levels within the one level can be developed with little difficulty.

The greatest drawback of using this test generating system is the fact that it requires a very large memory area to run in. This problem can be minimized if the user will choose a time to run when the core available is at a maximum.

References

1. Data Base Administrator's Procedures Manuel, Digital Equipment Corporation, Maynard, Massachusetts, 1973.
2. Programmer's Procedures Manuel, Digital Equipment Corporation, Maynard, Massachusetts, 1973.
3. Boyle, Thomas A. Combined text and test. Second International Symposium on Educational Testing. Montreux, Switzerland, July, 1975.
4. Boyle, Thomas A., Goodwin, William M., Hood, Gary A. Diagnostic test scoring and academic counseling by computer. Second Indiana Computer Network Conference on Computer Related Curriculum Materials. Kokomo, Indiana, March, 1975.