AD-A049 262 NAVAL INTELLIGENCE PROCESSING SYSTEM SUPPORT ACTIVITY--ETC F/G 9/2
DATA BASE MANAGEMENT SYSTEM (DBMS) SELECTION CRITERIA.(U)
FEB 75

UNCLASSIFIED NL

| OF |
AD
A049262

END
DATE
FILMED
3-78
DDC

(12) 21p.

(11) 13 February 1975

Naval Intelligence Processing Sys Support Activity

(1)

(6) DATA BASE MANAGEMENT SYSTEM (DBMS) SELECTION CRITERIA

1. The selection of an effective DBMS for use on the IBM 360 to support the Naval Intelligence Command On-Line System (NICOLS) is based on a set of ten evaluation criteria.

a. Networked Data Structure - A networked data structure permits unlimited association of inter-related data records, providing the intelligence analyst with broad data-association flexibility.

b. Data Record Placement Control -

(1) The DBMS must permit physical placement of data in specific location areas of the data base without application programmer interaction. This feature improves the processing efficiency without increasing the complexity and development effort of application programs.

(2) Data records within associated groups (sets) must have the ability to be stored in multiple physical data sets without programmer knowledge or interaction. Separation of certain types (or classifications) of data must not increase the complexity of program development.

(3) The DBMS must permit operation with only those portions of the data base required to be on-line at one time, but allow dynamic loading (mounting) of additional physical data sets upon request without programmer knowledge or interaction. Hardware constraints must not affect the logic or complexity of program development. Physical removal of sensitive data must be possible when access to that data is not required. These physical functions must be easily performed without degradation of the system performance.

(4) The Data Base Administrator (DBA) must be able to 'TUNE' the DBMS by designating certain records to be stored randomly and others to be stored in close physical proximity to their 'OWNERS'.

c. COBOL Interface Capability -

(1) The DBMS must provide a preprocessor which processes ANS COBOL source programs with COBOL-like Data Manipulation Language (DML) verbs, translating the DML verbs to 'CALL' statements without programmer action. Such a preprocessor reduces the complexity of program coding and the

H10 508

programmer experience level required to support user requirements.

(2) The preprocessor must load into the data division of the COBOL program all control areas required by the DBMS. This feature eliminates extra programmer coding effort and possibilities of error during coding and/or punching of statements.

(3) The preprocessor should load into the data devision all working-storage entries (COBOL) required to utilize data base records. Standardization of data names, reduction of programmer effort, and elimination of coding and/or punching errors are achieved by this capability.

(4) The preprocessor must exercise syntatical and logical evaluation of the programmer's use of DML verbs to insure that invalid or restricted verbs are not employed. The additional error checking performed by the preprocessor speeds up debugging and reduces the time and costs necessary to implement user's requests.

d. Logical Independence of all Data from Programs -

(1) The application programs must utilize only those logical records, or portions thereof, required for the function being satisfied. An integrated data base contains a broad scope of data subjects, not all of which are pertinent to an individual application.

(2) The application program must not be able to affect the structure of the data base. An application program views only a portion of the data base. Changes to the data base structure would have consequences beyond the scope of the modifying program.

e. Data Privacy -

(1) Application programs must be restricted to only those areas of the data base authorized for their use. The NICOLS system will contain data for many users of varying sensitivities.

(2) An effective means of restricting the types of access (FIND, OBTAIN, STORE, MODIFY, etc.) to areas, sets and records must be included. Programmers must be limited to those functions necessary to perform the task requested without endangering other areas of the data base.

(3) Physical separation of data onto specific direct-access devices must be logically transparent to

programmers. The physical arrangement of data within the data base is of no concern to the programmer.

(4) Physical removal of a portion of the data base must not impact or prevent use of the remainder of the data base. Sensitive portions of the data base may be physically mounted only during times of actual use.

(5) The software controlling the functions associated with the DBA (Data Base Structure Design, Privacy, User Authorizations, etc.) must be separate from those of the applications programmer using the data base. The integrity and security of the total data base is dependent on the separation of definition and manipulation functions.

(6) Pages (blocks of records) extracted from the data base must be physically isolated from the application program so that unauthorized access to nearby records can be prevented. Physical separation is interpreted to mean a separate Operating System (OS) partition or region (Logical block of core storage).

f. Concurrent update and retrieval capability. The DBMS must be capable of concurrently updating the data base while permitting retrieval of data. Activity must be restricted only to the occurrence of a record type which is being updated. The NICOLS system will support multiple users operating remote terminals. The response to user's inquiries to the system must be rapid and not dependent upon the activities of other users.

g. Automatic restart and recover capability -

(1) The DBMS must be capable of automatic recovery, without operator intervention, from abnormal termination of individual run-units (executing programs). The on-line use of the system is dependent upon the ability of the system to quickly recover from problems in application software. Abnormal termination of one user's program must not adversely affect service to other users.

(2) The DBMS must be able to restart following a software or hardware failure during update without restoring a previous copy of the data base. Restoring a previous copy of the data base is a time-consuming task. During such a restore operation on-line users of the system would be unable to utilize the system's facilities.

h. Hardware Constraints. The DBMS must be able to operate effectively and without restriction within the following hardware constraints:

(1) Central Processing Unit (CPU) - An IBM 360 with 350,000 bytes of core storage available for DBMS, teleprocessing (TP) module, and application programs. A minimum of 150,000 bytes must be available for application programs.

(2) A maximum of six drives (Spindles) of 2314-Type disk storage.

i. Teleprocessing (TP) Interface. The DBMS must be readily attached to a TP monitor program. The NIPS/FFS TP monitor has been selected for use during initial implementation of on-line processing. The DBMS must satisfactorily interface with the TP monitor and, including the 50,000 bytes of core storage required for the TP package, reside in 200,000 bytes of core or less.

j. Training -

(1) The vendor must supply, as part of the initial contract costs (travel excluded) adequate training to familiarize ten operations, six systems software, and twenty-five applications software personnel with the system.

(2) Additional follow-on training must be available.

(3) Training courses for applications personnel on video tape is highly desirable.

2. The DBMS packages reviewed during evaluation of the NICOLS requirements were:

a. ADABAS (Software AG, Inc.)
b. IDMS (Cullinane Corporation)
c. IMS (IBM Corporation)
d. MODS (IBM/FSD - IPAC, A Government-owned package)
e. NIPS/FFS (IBM/FSD - DCA, A Government-owned package)
f. SYSTEM 2000 (MRI, Inc.)
g. TOTAL (CINCOM Systems, Inc.)

3. Each of the systems have been evaluated against the detail criteria to identify their ability to support the NICOLS system.

DBMS EVALUATION RESULTS TABLE

CRITERIA	ADABAS	IDMS	IMS/360	MODS	NIPS/FFS	S2000	TOTAL
A	Yes	YES	NO	NO	NO	YES	NO
B(1)*	YES	YES	PARTIAL	NO	NO	NO	NO
B(2)*	NO	YES	YES	NO	NO	NO	NO
B(3)*	NO	YES	NO	NO	NO	NO	PARTIAL
B(4)*	NO	YES	NO	NO	NO	NO	NO
C(1)*	NO	YES	YES	NO	NO	YES	NO
C(2)*	NO	YES	NO	NO	NO	NO	NO
C(3)*	NO	YES	NO	NO	NO	NO	NO
C(4)*	NO	YES	YES	NO	NO	YES	NO
D(1)*	NO	YES	YES	NO	NO	NO	NO
D(2)*	YES	YES	YES	NO	NO	PARTIAL	YES
E(1)*	YES	YES	NO	NO	NO	NO	NO
E(2)*	PARTIAL	YES	NO	NO	NO	NO	PARTIAL
E(3)*	NO	YES	NO	NO	NO	NO	NO
E(4)*	NO	YES	NO	NO	NO	NO	PARTIAL
E(5)*	YES	YES	YES	NO	NO	PARTIAL	NO
E(6)*	YES	YES	YES	NO	NO	YES	NO
F*	YES	YES	PARTIAL	NO	NO	NO	NO
G(1)*	NO	YES	NO	NO	NO	NO	NO
G(2)*	PARTIAL	YES	NO	NO	NO	NO	PARTIAL
H(1)*	YES	YES	NO	YES	YES	NO	YES
H(2)*	YES	YES	NO	YES	YES	NO	DEGRADED
I*	NO	YES	NO	YES	YES	NO	UNDET
J(1)*	YES	YES	NO	YES	YES	NO	YES
J(2)*	YES	YES	YES	YES	YES	YES	YES
J(3)*	NO	YES	YES	NO	NO	NO	YES

4. Results:

SYSTEM	MANDATORY CRITERIA (17) MET	FAILED	DESIRABLE CRITERIA (9) MET	FAILED
ADABAS	9	8	4	5
IDMS	17	0	9	0
IMS/360	8	9	3	6
MODS	3	14	2	7
NIPS/FFS	3	14	2	7
S2000	4	13	3	6
TOTAL	6	11	5	4

5. IDMS was able to satisfy all mandatory requirements while all other systems failed at least eight mandatory requirements.

6. Procurement of the Cullinane Corporation's IDMS data base management system is strongly recommended based on the evaluation performed.

13 February 1975

SPECIFIC EVALUATION POINTS -

1. Data base development -

a. Amount of effort required to establish the schema for the Resources Accounting function.

b. Enhancements to the data base structure which were achieved under IDMS but not possible/practical under MODS.

c. Amount of effort required to add an additional function capability to the developed schema:

(1) before system generation
(2) after system generation
(3) after data base loading (Resources Accounting data is resident in the data base)

2. Data Base effectiveness-

a. Number of bytes of data in MODS Resources Accounting data base vice number of bytes of data in IDMS version of data base, broken into:

(1) bytes of raw usable data
(2) bytes of overhead data (pointers, count bytes, sort keys, etc.)

3. Data base updating

a. Record update volume per minute using MODS and IDMS.

b. Man hours required to develop an input processing module using the boilerplate approach. An average of several modules plus high and low man-hour range will be used.

c. Improvements in updating capability, if any, afforded by IDMS over MODS.

4. Data base retrieval -

a. Man hours required to develop output reports for IDMS using COBOL Report Writer. CULPRIT is not part of the evaluation but would be used in batch retrievals under a complete IDMS configuration.

b. Machine time required to produce specific reports of the Resources Accounting System under MODS and IDMS.

5. On-line utilization. Testing and evaluation of on-line utilization is dependent upon development of control programs necessary to permit interactive access to IDMS via CRT terminals or card reader/printer. If development of these programs can be completed during the test-period the following evaluations will be made.:

a. Test the ability of the system to accept randomly ordered input data (data for multiple record formats). Throughput in records per minute and CPU seconds per record will be recorded.

b. Retrieve selected data from the data base utilizing a formatted query and prepare the output in a selected report format previously defined. Record response time to query and ability of the system to service multiple queries concurrently.

c. Test the ability of the system to concurrently update the data base at the same time a retrieval(s) is in progress.

6. Data base recovery -

a. Test the ability of the system to recover automatically from the abnormal termination of a run-unit (an individual program accessing the data base) in update mode. This will be done by utilizing a program which deliberately aborts with a processing error.

b. Test the ability of the system to recover automatically from a hardware failure during data base update. This will be done by deliberately causing a hardware malfunction.

c. Test the ability, and ease of, data base save and restore functions.

7. Data base security -

a. Attempt to access data in areas outside those authorized to the using program.

b. Attempt to update data in areas outside those authorized to the using program.

c. Attempt to access data elements in records which are not authorized to the using program.

d. Attempt to process in update and retrieval modes prohibited by the subschema.

e. Attempt to violate the data dictionary with a normal run-unit.

f. Attempt to retrieve data being used by a concurrently operating run unit.

g. Attempt to lockout the system by concurrent access of the same area, record type, and record occurrence by two run-units.

h. Define the relative security stability of MODS vice IDMS.

i. Is it actually practical to restrict the application programmer to only those data records and/or data elements required to produce his assigned task? Can the programmer effectively produce his assigned task with this limited knowledge?

8. Ease of Operation -

a. What is the reaction of operations personnel to the manner in which IDMS operates? What is its relative ease of use in comparison to MODS?

b. How much operator interaction is required for IDMS vice MODS?

C. What operator actions are necessary to load and unload IDMS?

9. Core Storage Utilization -

a. What are the core requirements for IDMS (central operating environment) for update and report output functions. Include the application programs required to update and produce reports. How does this compare with OS MODS?

b. What are the core requirements for an on-line version of IDMS (central environment)? This core requirement is dependent upon the manner in which in-house prepared programs are written and linked together to form the on-line structure.

c. Can IDMS function effectively as a batch processor in available core storage? What constraints must be placed on IDMS capabilities to achieve use in available core storage?

d. Will IDMS function in association with the NIPS/FFS teleprocessing module in available core storage? What constraints must be placed on IDMS capabilities to achieve use with the TP module in available core storage?

10. Programmer use -

a. What features found in MODS, to assist programmers in program development, are not found in IDMS?

b. What features found in IDMS, to assist programmers in program development, are not found in MODS?

c. What is the level of programming knowledge necessary for programmers to effectively utilize IDMS?

d. What is principal advantage(s), to application programmers, of IDMS over MODS?

e. What is principal advantage(s), to application programmers, of MODS over IDMS?

f. Is the IDMS DML manual adequate for programmers to effectively utilize the system? What deficiencies were noted?

g. How does the IDMS DML manual compare to MODS in ease of use?

h. What features of IDMS are most effective in achieving working programs in a minimum amount of time?

i. Are messages produced by the system and by the DML processor adequate to locate apparent logic errors in programs?

11. Vendor support -

a. What is the response time by Cullinane Corporation, to questions, on use of IDMS? Are these responses adequate?

b. What is the response time, by Cullinane Corporation, to apparent system malfunctions?

13 February 1975

IDMS EVALUATION RESULTS

1. Data base development -

a. Criteria. Determine the amount of effort required to establish the schema (logical data base structure) for the Resources Accounting function.

Results - Thirty-four hours were required to develop the basic Resources Accounting schema. Twenty-eight additional hours were required to add related logical enhancements to the schema.

b. Criteria. Describe those enchancements to the data base structure which were achieved under IDMS, but not possible/practical under MODS.

Results -

(1) A major data category for individual staff members was added providing personnel data capabilities both related to, and independent of, the original resources accounting function.

(2) Multiple organizational units can be processed as readily as one. Outputs may be organized by unit.

(3) A separate training file (currently dormat) was included to enhance the information available on personnel.

(4) Security clearance data, foreign language skills, and occupational skill data were added to enhance personnel data.

(5) Set structures were established which permit elimination of sorts for virtually all output products.

(6) Personnel billet information was added to improve personnel information.

c. Criteria. Determine the amount of effort required to add an additional function capability to the developed schema:

Before System Generation
After System Generation
After Data Base Loading

Results -

(1) Approximately two hours per record format is necessary to determine the set structures required.

(2) The same time as in (1) above plus a schema compile run requiring one hour of 360 machine time.

(3) For records unrelated to the existing data base, for records related to the existing data base, but which may be added to an existing set as members, for existing records which require additional data elements and space is available in the record format, for existing records which require additional data elements and filler space is not available a member record is defined containing the additional elements and attached to an existing set, the same time as in (2) above.

(3a) For records related to the existing data base and which require new member pointers in an existing record- (3A1) if allowance has been made for pointer expansion in the record format, the time described in (2) above. (3A2) If allowance has not been made for additional pointers the affected records must be off-loaded and reloaded after schema redefinition. Fifty to one hundred man-hours may be required depending on record/set complexity, (Cullinane is developing a utility program to simplify this task).

2. Data Base Effectiveness -

a. Criteria. Determine the number of bytes of data storage required to retain data in IDMS, MODS, and COBOL program files.

Results -

(1) The resources accounting data base supported by MODS requires 1,125,908 bytes (915,524 Data, 210,384 Overhead). The same file, supported by IDMS requires 989,721 Bytes (741,841 Data, 274,880 Overhead). The IDMS version requires 12.2 percent less data storage space than the MODS version.

(2) Another application operating under COBOL program control requires 12,160,879 bytes of disk storage to store the data base. Under IDMS the data volume would be reduced 87.99 percent to 1,460,762 bytes.

3. Data Base Updating -

a. Criteria. Define the record update volume per minute using MODS and IDMS.

Results -

(1) MODS. Update volume is dependent upon the number of records on tape and gradually increases as the

size of the data base grows regardless of the number of raw data items. All updates require a minimum run time to pass the magnetic tape file.

(2) IDMS. Update volume, using input processing subprograms, ranges from three to twenty records a second. The range is dependent on the mode of update for each record type. No minimum run-time is required due to the use of direct access storage.

b. Criteria. Define the man-hours required to develop an input processing module using the boilerplate approach. An average of several modules plus high and low man-hour range will be used.

Results -

(1) The number of modules developed using this approach were twenty-six. Man-hours required to prepare the module ranged from a minimum of one hour to a maximum of thirty-four hours. The average time for the twenty-six modules was twelve hours.

c. Criteria. Define the improvements in updating capability, if any, afforded by IDMS over MODS.

Results -

(1) IDMS allows use of data structures beyond the capabilities of MODS. IDMS permits broader use of the data base as a basis for validating raw input data.

4. Data Base Retrieval -

a. Criteria. Define the man-hours required to develop output reports for IDMS using the COBOL Report Writer.

Results.

(1) Depending on the complexity of the report, four to 48 hours were required to develop IDMS reports using the COBOL Report Writer feature. A majority of the reports produced were written in less than eight man-hours.

b. Criteria. Determine the machine time required to produce specific reports of the Resources Accounting system under MODS and IDMS.

Results -

(1) The project worksheet is produced in approximately seven minutes under MODS. An accurate parallel run

using IDMS/Report Writer was not possible as the IDMS data base does not contain the full volume of data resident in the MODS data base. Tests of a smaller amount of data indicate that the IDMS/Report Writer approach will execute at the same or faster speed.

5. On-Line Utilization -

Control programs necessary to perform on-line operation of IDMS could not be completed during the limited test period. However, the input processing programs developed were designed to operate in conjunction with either a batch or on-line controller.

6. Data Base Recovery -

a. Criteria. Test the ability of the system to recover automatically from the abnormal termination of a run-unit (an individual program accessing the data base) in update mode.

Results -

(1) Tests were run in both batch and central operating modes forcing abnormal termination of programs. In batch mode the operator is informed of the error and a journal roll-back is automatically executed removing the bad run-unit activity from the data base. Generally operating mode recovery was automatic and the operator was advised following recovery. Normal operation continued in both cases.

b. Criteria. Test the ability of the system to recover automatically from a hardware failure during data base update.

Results -

(1) Deliberate disabling of hardware devices, including the data base disk unit, failed to adversely affect the system. When the data base disk was disabled IDMS waited until it was again available and continued processing.

c. Criteria. Test the ability and ease of data base save and restore functions.

Results -

(1) IDMS provides utilities to save and restore the data base totally, by OS file, or by data base area. The utilities perform as presented. The save provides a statistical report defining the loading and distribution of

data by data base area, thus permitting continual monitoring of the storage capacity of the data base.

7. Data Base Security

a. Criteria. Attempt to access data in areas outside those authorized to the using program.

b. Criteria. Attempt to update data in areas outside those authorized to the using program.

c. Criteria. Attempt to access data elements in records which are not authorized to the using program.

d. Criteria. Attempt to process in update and retrieval modes prohibited by the subschema.

Results -

(1) Definition of record types outside the authorized scope of the users subschema causes the COBOL preprocessor to terminate without compiling the users program. If a program is compiled withoug using the preprocessor the program execution produces erroneous results when invalid commands are used.

e. Criteria. Attempt to violate the data dictionary with a normal run-unit.

Results -

(1) The data dictionary is a separate OS data set. Normal system execution excludes this data set from the JCL. No data patch is possible unless it is included in the JCL. With the data dictionary included in the JCL a special subschema and DMCL module, not available to the normal run-unit, are required to access the data dictionary.

f. Criteria. Attempt to retrieve data being used by a concurrently operating run-unit.

Results -

(1) Execution of concurrent usage programs which would perform this test was impractical during the test period.

g. Criteria. Attempt to lockout the system by concurrent access to the same area, record type, and record occurrence by two run-units.

Results -

(1) Execution of concurrent usage tests were limited to 'Protected' mode which limits access to the current record of a run-unit. No noticeable lockout occurred. The second run-unit accessing a record current of the first run-unit waited for release of the record.

h. Criteria. Define the relative security stability of MODS vice IDMS.

Results -

(1) MODS. All record formats are accessible to the programmer.

(2) The programmer may change record formats without detection.

(3) The entire file description (all record formats) is required to effectively utilize the file.

(4) No privacy locks are available.

(1) IDMS. Only record formats required for an application are available to the programmer.

(2) Programmers may not alter record formats.

(3) A limited subset (subschema) of the data base may be effectively utilized to accomplish most applications. The subschema is tailored to the users needs.

(4) Privacy locks are available which limit access of areas, sets, and records and type of access (store, obtain, modify, delete, etc).

i. Criteria. Determine if it is practical to restrict the application programmer to those data records and/or data elements required to produce his assigned task.

Results -

(1) This practice is common in tape-oriented batch processing. The extension to IDMS keeps related data available while isolating unneeded data from the programmer.

j. Criteria. Determine if the programmer can effectively produce his assigned task with this limited knowledge.

Results -

(1)The programmer is provided with as much of the

data base description as is necessary to produce his task.

8. Ease of Operation -

a. Criteria. Describe the impact upon operations personnel by IDMS operation.

Results -

(1) Normal operation of IDMS is transparent to the operator.

b. Criteria. Describe the amount of operator interaction required for IDMS vice MODS.

Results -

(1) MODS requires tape handling. Operators must mount tapes for each job and most large sorts.

(2) IDMS utilizes disk data base and operator action is required only to terminate central version execution when all work is complete.

c. Criteria. Describe the Operator Actions necessary to load and unload IDMS.

Results -

(1) Load: Load a previously prepared JCL deck to start the central version. No action is required on batch version. Unload: Operator types message on console to terminate central version. No operator action is required on batch version.

9. Core Storage Utilization.

a. Criteria. Define the core storage requirements for IDMS (central operating environment) for update and report output functions.

Results -

(1) The central environment requires 46,280 bytes of storage when using eight input/output buffers of 1692 each. The typical application program requires 33,376 bytes of core storage (20,000 bytes for the program itself, the remainder for subschema and IDMS interface routines). The total core storage requirements for this arrangement is 79,656. This is 40,152 less than the 119,808 bytes required for OS MODS.

b. Criteria. Define the core requirements of the on-line (central environment) version of IDMS.

Results -

(1) The programs for the central version of IDMS require 32,744 bytes of core storage. Buffer core storage requirements vary depending on the number of buffers and buffer size. A four-buffer arrangement requires 6,768 bytes (39,512 total core size) and the eight-buffer requirement is 13,536 (46,280 total core size).

c. Criteria. Determine if IDMS can function effectively as a batch processor in available core storage.

Results -

(1) An average application program (IDMS application programs are much smaller due to removal of files from the program) and associated IDMS modules will operate in less than 100,000 bytes of core storage. No constraints are placed on IDMS capabilities to achieve full use in available core storage.

d. Criteria. Determine if IDMS will function in association with the NIPS/FFS teleprocessing module in available core storage.

Results -

(1) IDMS will function with the NIPS/FFS teleprocessing module in approximately 98,280 bytes of core storage. IDMS requires 46,280 bytes and the NIPS/FFS module requires approximately 52,000 bytes. Interface programs to support teleprocessing functions may function in the remaining 50,000 bytes of storage available. No constraints are placed on IDMS functions when operating in this configuration

10. Programmer Use -

a. Criteria. Determine what features are found in MODS, to assist programmers in program development, that are not found in IDMS.

Results -

(1) System formatted output, variable text scan, partial notation, and geographical conversion routines are found in MODS, primarily as an output function. A majority of the MODS routines performing the above functions can be adapted for use with IDMS.

b. Criteria. Determine what features are found in IDMS, to assist programmers in program development, that are not found in MODS

Results -

(1) IDMS provides:
- DML preprocessor checking of validity of DML verbs to access the data base.
- The DML preprocessor loads all record formats to working-storage of each program.
- User is permitted multiple entry points into the data base.
- IDMS is a language extension of COBOL, rather than a full language.
- Unlimited numerical processing capability supports all forms of numerical data found in IBM 360/370.

c. Criteria. Define the level of programming knowledge necessary for programmers to effectively utilize IDMS.

Results -

(1) Junior level programmers trained in ANS COBOL and IDMS DML usage can effectively write programs using IDMS.

d. Criteria. Define the principal advantages, to application programmers, of IDMS over MODS.

Results -

(1) Flexibility of using COBOL with it's broader features.

(2) File definition requirements are eliminated.

(3) More effective control over data may be achieved with less logic coding.

(4) Logic is easier to develop when repetitive processing is required.

(5) One language to be familiar with.

e. Criteria. Define the principal advantages, to application programmers, of MODS over IDMS.

Results -

(1) Easier to produce data base queries.

(2) Simple reports are generated more quickly.

f. Criteria. Determine if the IDMS DML manual is adequate for programmers to effectively utilize the system. Note any deficiencies.

Results -

(1) Programs have been written, debugged, and executed using the manual as the only point of reference. Training emphasizes the manual rather than supplements it. Deficiencies were found in error status descriptions. A few questions are unanswered except by experience or call to the vendor.

g. Criteria. Describe the comparison between the IDMS DML manual and the MODS manuals as to ease of use.

Results -

(1) The IDMS DML manual is much easier to use. Effective programs can be written immediately, without external training, using only the manual.

h. Criteria. Define the features of IDMS which are most effective in achieving working programs in a minimum amount of time.

Results -

(1) The DML pre-processor loads all necessary control areas, working-storage record formats, and standard routines into COBOL programs automatically.

(2) File descriptions and file control logic are eliminated for data base access.

(3) DML pre-processor provides diagnostic messages pointing out logic or syntax errors in DML verb usage.

i. Criteria. Determine if the messages produced by the system and by the DML preprocessor are adequate to locate apparent logic errors in programs.

Results -

(1) Messages pinpoint statements (DML preprocessor) which are in error or potentially in error. Error status information (DB Manager) specifically indicates the type of error. Descriptions of all messages can be improved.

11. Vendor Support -

a. Criteria. Determine the response time, by Cullinane Corporation, to questions on the use of IDMS and the adequacy of the response.

Results -

(1) Cullinane has provided information immediately in all cases of requests for assistance in the use of the system. All questions have been answered adequately and accurately.

b. Criteria. Determine the response time, by Cullinane Corporation, to apparent system malfunctions.

Results -

(1) An error in the subschema processor utility was found and reported to Cullinane. An alternate approach to avoid the error was provided immediately. A new version of the system, with the error corrected, was hand-delivered and installed within seven working days