OVERVIEW
OF
THE NEW
MSC/NASTRAN EXECUTIVE SYSTEM

INTRODUCTION

The rapid increase in the capabilities of computers has enabled the engineering community to solve larger and more complex problems. The rapid increase in computer types and environments has enabled the engineering community to best suit their tasking requirements for both price and performance. These advantages have come, however, not without a price. The penalty involved is the increase in data management requirements and the handling of data distribution.

Also, advances in engineering technology has expanded the horizons of what is feasible for inclusion into MSC/NASTRAN. Thus MSC must now concern itself with a many facetted integration of engineering disciplines such as design optimization.

The new executive system of MSC/NASTRAN addresses the problems of data management and distribution and the integration of engineering disciplines.

MAIN FEATURES

The main features of the new executive system are:

1) A MSC/NASTRAN Definition Language (NDDL). This language is the key feature in the description of the data base and is of prime importance in handling the distribution of data between machine types by providing the schema necessary for representing the data block structure. The NDDL allows for the description of a hierarchical data structure (which is the prevalent structure in engineering analysis) but also provides the necessary mechanisms to form, for data blocks selected from the hierarchy, the flat table description required for relational type data base systems.

2) An automatic restart capability based on the dependencies of data blocks, data items within data blocks, and parameters as they relate to changes in Bulk data. The dependencies are described in the NDDL. At execution time the MSC/NASTRAN Executive System (NES) locates the appropriate restart version and using the dependency information flags any modified data blocks or parameters. When the NES invokes a module it checks the output of the module for flagged data blocks or parameters. If any have been flagged, the NES causes the module to execute. If none have been flagged, the module is not executed.

3) A Data Base Management System (DBMS) which allows for easy and flexible control of secondary space (disk) or real memory (to simulate disk within the random access memory of the system). In addition, the DBMS provides the correlation between existing data blocks and parameters saved on the data base and data blocks and parameters modified because of restart. Thus, DBMS provides the mechanism to associate various data blocks or parameters with the appropriate restart version. This allows for more efficient use of the data base storage area and for efficient collection of the required information for a particular restart.

4) An enhanced Direct Matrix Abstraction Program (DMAP) language which allows for structuring of solution sequences into task oriented areas of engineering analysis. This is accomplished by providing modern block structured techniques with SUBDMAP calls. The enhanced DMAP also provides, through the NES, for efficient and direct communication with the database and allows

for the definition of relational views of the data structure. The structuring of the solution sequences provide, for the DMAP writer, more maintainable and understandable solution sequences. The SUBDMAP concept in conjunction with the restart concept will provide for the analyst, more efficiency in the execution of the solution sequences.

HIGHLIGHTS OF THE NDDL

The purpose of the NDDL is:

1. The NDDL describes the hierarchical data structure of the MSC/NASTRAN Data Base.

2. The NDDL provides the mechanism for determining which MSC/NASTRAN generated Data Blocks or Parameters or both will be stored on the Data Base.

3. The NDDL provides the schema necessary for representing the Data Block data structure.

4. The NDDL and its associated query language provides the means for selecting from the hierarchical Data Base structure specific Data Blocks in the form of flat tables thus admitting them to relational Data Base manipulation.

5. The NDDL provides the necessary data dependencies for automatic modified restarts.

Thus the NDDL may be considered a language for completely describing the structure of the engineering data.

The NDDL then provides:

** A unique identification of each item of data.

** A description of the functional dependence of a data block or parameter on another unit of data.

** A mechanism for portability of data between computers.

** A means of indicating to the NES the storage location of a data block or parameter.

** A means of identifying to the NES the structure of a data block or data type of a parameter and its default value.

** A data management technique consistent with modern data management concepts.

The command structure of the NDDL is:

** DATABLK used to describe the structure of data blocks and the location of data blocks to the DBMS and NES.

** PARAM used to describe parameters and their type and location to the DBMS and NES.

** DEPEN used to express the functional dependence of a data block or parameter on another unit of data.

** PATH used to describe the hierarchical data structure to the DBMS and NES.

** QUAL used to describe external qualifiers which are used to locate data blocks and parameters in the hierarchical data structure.

HIGHLIGHTS OF THE AUTOMATIC RESTART

There are two basic restart concepts in the new MSC/NASTRAN executive system:

1. Unmodified restarts which are normally used when it is indicated that for some reason the MSC/NASTRAN job did not have a normal completion. In this case 'snap shots' at predefined locations within a solution sequence are taken (with only the most current retained) and on resubmittal the job will commence execution at the 'snap shot' location.

2. Modified restarts which are normally used to continue the solution process from one type of solution sequence to another or to execute the same solution sequence with modified input. In this case, the restart always starts at the top of a solution sequence and determines from data base information and data dependencies whether a module or SUBDMAP need be executed.

Automatic restart then provides:

** Unanticipated job ending recovery.

** Efficient and flexible restart of solution sequences based on problem requirements determined by the analyst.

** Restart from various versions of multiple generations of a specific project.

** Overall restart strategy still valid after analyst modifies a solution sequence with her or his own DMAP variations.

The command structure of the restart is:

** RESTART used to indicate to the NES that either a UNMODIFIED or MODIFIED restart is to occur.

** VERSION used to select from the data base the explicit previous set of data blocks and parameters to be used in the modified restart.

** STOREAS used to delete from the data base, data blocks and parameters from the restart version and replace them with the data blocks and parameters from the modified restart.

** LOCATE used to override the VERSION selected set of data blocks or parameters with selected data blocks or parameters from versions other than the restart version.

** CHKPNT used to indicate to the NES that any DMAP CHKPNT, CHKEXIT, or REPEAT statement is to take a current 'snap shot' of the solution for unmodified restarts.

HIGHLIGHTS OF THE DBMS

The purpose of the DBMS is:

1. Externally it allows the analyst or data base manager to specify physical locations and the amount of space for data base usage. In essence it correlates the logical data base structure (the location attributes of data blocks and parameters as described in the NDDL) with the physical location of the data. The user must, however, still supply according to his or her installation requirements the necessary machine dependent control statements tying the logical device to the physical devices.

2. Internally it maintains the pointers to existing data blocks and parameters and between restart versions. It is the mechanism through which the NES accesses the data base.

The DBMS then provides:

** The vehicle through which the NES accesses the data base.

** Job version identification as multiple jobs are generated.

** The facility for the analyst to accesses MSC/NASTRAN source DMAP libraries from which he or she can build their own special DMAPs.

** The facility to accesses and execute MSC/NASTRAN precompiled solution sequences.

** The facility by which the analyst can store, and execute his or her own precompiled solution sequences.

** The facility to initialize disk or real memory (RAM) for data base space allocation.

** The facility to expand the number of physical devices within an existing data base.

** The ability to DUMP and RESTORE the data base.

** The ability to DELETE portions of the data base.

** The ability to ATTACH portions of other data bases.

** The ability to RECOVER from system aborts.

The command structure of the of the DBMS is:

** INIT used to initialize disk space or RAM for data base management usage. It assigns the logical name to the data base set and defines the number of words in a GINO block and specifies the cluster size of a unit of GINO space allocation.

** EXP used to expand the number of physical devices associated with an existing data base set.

** SECURITY used to place security on data blocks an parameters stored on the data base.

** PROJ used at INIT time to allow the data base manager to assign additional PATH qualifiers to the data base hierarchical structure to meet particular installation requirements.

** COMPILE used to specify which NDDL or SUBDMAP to compile and where to place resulting object deck.

** SOL used to specify the DMAP main program which will be linked to form the executable Operational Sequence Control ARray (OSCAR). Allows for the saving of the OSCAR for direct execution in subsequent runs.

HIGHLIGHTS OF THE ENHANCED DMAP

The purpose of the enhanced DMAP language is:

1. To provide the structuring of the DMAP. DMAP instructions consist of three distinct types: Executive operations, Functional modules, and Output processor modules. It is the Executive operations that provide the new structuring features. DMAP control statements now include SUBDMAP capability to break solution sequences into specific engineering tasks. Within SUBDMAPs there is now block structuring capability such as DO WHILE and IF THEN. The structuring makes solution sequences more reliable and much more understandable.

2. To provide direct communications between the solution sequences and data base via the NES. This is the heart of the automatic restart which when combined with the SUBDMAP concept results in highly efficient run times since conceivably, depending on modified input, large sections of the solution sequence will not be executed.

The enhanced DMAP then provides:

** For NDDL defined data blocks and parameters, and in conjunction with the NES, automatic retrieval from the data base at module invocation and automatic writing to the data base at module completion.

** More understandable solution sequences via the structured approach.

** Efficient restart in conjunction with the NES and SUBDMAP.

** Expressions and operator capability for parameter manipulation.

** Assignment statements for setting parameter values.

** Better error handling and information dissemination through inline user definable message writer.

The most pertinent new and modified DMAP command structure is:

** The Assignment statement (computational) evaluates an expression and assigns the resulting value to the variable parameter specified.

** The Expressions may be simply a string or numeric constant, a variable, or may combine constants and variables with operators to produce a single parameter value.

** The Operators are divided into the following categories:
Arithmetic
Character
Relational
Logical

** The CALL statement calls and executes a SUBDMAP from another DMAP unit.

** The SUBDMAP statement identifies a DMAP as a SUBDMAP, gives it a name, and identifies the dummy arguments of that SUBDMAP.

** The DO WHILE statement repeatedly executes the statements contained in the current DO-loop as long as the logical expression contained in the statement is TRUE.

** The ENDDO statement terminates the range of a DO WHILE.

** The IF statement (logical IF) evaluates a logical expression and, if the value of the expression is TRUE, executes the statement given.

** The IF THEN (block IF) evaluates a logical expression and, if the value of the expression is TRUE, begins executing statements in the IF block. If the expression evaluates to FALSE, control transfers to the next ELSE IF, ELSE, or ENDIF statement at the same IF-level.

** The ELSE statement marks the beginning of an ELSE block.

** The ELSE IF causes the evaluation of an expression of the form: ELSE IF (expression) THEN.

** The ENDIF statement terminates a block IF.

** The RETURN statement returns control to the calling SUBDMAP.

** The TYPE statement specifies a parameter type and whether the parameter is a NDDL parameter. Also the TYPE statement is used to identify NDDL data blocks.

** The View statement creates a virtual data block from data blocks defined in the NDDL. It forms a data block that does not exist as a separate physical entity but only as a logical entity.

** The DELETE module deletes data blocks and marks their names as not generated.

** The DIRPRT module reads directory print requests and prints data base directory information.

** The MESSAGE module allows the DMAP to write user defined messages.

Some other modules that are either new or have had modification to make them work more efficiently are:

DMAP MODULES

CHKEXIT
CHKPNT
DMIIN
DTIIN
EQUIV
IFP
IFP1
PRTPARM
PURGE
PVT
XSORT

FUNCTIONAL MODULES

SEP2DRV
SEP2
SEP2CNT

SELA

SEDRDRV
SEDR

SEMA

SEP4

PLTVECDR
PLTVEC

CONCLUSION

The above description represents a brief summary of the current development of the MSC/NASTRAN new executive system now under testing at MSC. The aim of the new executive system is to provide the analyst with efficient handling of his data management and data distribution needs and to also provide more efficient and understandable solution sequences. In order to achieve these goals, the new executive system has many new features. For the most part, however, these new features are internally defaulted. Thus, only if special data handling is required or special solution sequence modification is needed does the analyst need to concern himself or herself with the particulars of the new executive and then only for the specific need at hand.