A VM/CMS - SAS (tm) Based Database Query Application Employed as a Function of a Contracted Business Data Management System

Michael L. Confer and David Septoff

ABSTRACT

In the course of developing a SAS (tm) based data management system (OMS), an end user requirement was uncovered which had not been identified during preliminary analysis or addressed by baseline system design. This requirement comprised the capability to "cut" a segmented database which documented business activity over a broad variety of contracted and in-house (eg: overhead) tasks in order to augment offline data analysis. A decision was made to execute this "cut" in the VM/CMS environment based on the availability of resources already dedicated to OMS development and comparative familiarity of end users with this operating environment. Output (report) requirements and key fields/characteristics upon which to base meaningful data extraction were identified through analysis and user interview and work commenced as described herein.

This paper describes the necessary interface required for successful interaction between SAS (tm) and VM/CMS in the maintenance and analysis of a segmented database. Problems inherent in the development of a successful methodology will be discussed with particular emphasis on the importance of establishment of a user friendly "buffer" between end user intent and processing in a novice user environment.

INTRODUCTION

During the spring and summer of 1985, the author participated in the design of a complex data management system (OMS) comprising an automated business planning system to be run utilizing SAS (notably FSCALC and FSEDIT) source code and datasets. These applications permit the collection and manipulation of data related to periodic organizational business planning cycles. These data are primarily based on the revenues and costs generated by contract and overhead tasks which are expressed in terms of manning, workload, contract type, fee, rates and other criteria common to the contract business environment. Generally, the planning process considers applicable business factors and applies these factors in an attempt to project near and long-term financial performance based on current data and alternate performance scenarios which are modeled to determine tradeoffs in management decision making. Performance of this contract was scheduled in such a manner that development of the SAS-based Data Management System (OMS) would parallel the development of concurrent organizational activities. This dictated an accelerated development schedule with several "hardwired" functions to be delivered quite early in the contract with iterative development of user requested OMS capabilities to be included in a tested 'baseline' version delivered with full documentation at the conclusion of the design and implementation effort.

The OMS, ultimately was designed to operate in the VM/CMS environment. Early applications were developed under OS/MVS via a remote site controller (RSCS). Ongoing communications problems forced an accelerated transfer of OMS to the local VM/CMS system. Design under VM/CMS must operate so as to:

- Integrate and store user input data from a variety of organizational elements;
- Analyze stored data to result in a cohesive database which can then be integrated into upper-level management applications;
- Automatically isolate key occurrences which delineate from statistical projections; and
- Permit the ready retrieval of data for offline analysis by authorized personnel.

It was decided, after consultation with several System users that it was necessary to support some form of online data query in order to satisfy the latter indicated requirements.

These criteria dictated that a high degree of user/system interaction would take place during the process of capabilities development and that system throughout, applications flexibility and accuracy would all be key components in system acceptance by the community. These factors were to no extent mitigated by the fact that, while several members of the user group were experienced data users, only a few were experienced users of a variety of computer systems; virtually no design or programming background was represented within the community. Since activity was in process, there would be extremely limited time for training. These factors led the designers to a number of conclusions:

- Experienced end user requirements had to be expedited through development whenever possible;
- The method of query must be user friendly, avoiding an extensive learning curve; a menu-driven query, if adequately flexible, would be the preferred method;
- Development testing must be adequate for delivery with fine tuning following actual implementation;
- The query must be able to be backfit and integrated with established OMS functions without modification of existing capability.

Once the basic design guidelines were agreed upon, analysis of user requirements commenced. Discussion of these requirements appears in the next section.

END-USER REQUIREMENTS

As discussed previously, many of the characteristics of the OMS were driven by the fact that end users of the system are by and large not "systems" people. As a group, they displayed a keen sense of the accuracy and turnaround of data supported by the system but did not have the technical knowledge of SAS required to directly query datasets via the interactive facility. Their requirements, however, far exceeded the capabilities of a pre-programmed "canned" query. They required a complex query to extract data to the unit-record level and flexibility in the selection of format, content, sort order and output method. Since no single method of extraction appeared to satisfy all user requirements, a four-option preliminary selec-
The selection of headcount as a key characteristic in three of the four available reports was based on two factors; first, that other facilities either existed or were planned within the OMS to support second and second that manpower is commonly the key factor in contract business planning over a long operational cycle.

Finally, requirements for output disposition and data integrity were fairly straightforward. Single point of write authority and unrestricted read access are built into the system under the constraints of the Control Program (CP). Use authorization follows the same constraint (application-specific least privilege) as does the reast of OMS. Output listings are produced by each body of SAS (tm) source code with a unique CMS filename assigned, is disposed of by print or browse facility and destroyed at the conclusion of disposal. The query dataset itself exists until end of session or restart and may have multiple reports run against it until EOS or restart is selected by the user.

THE SYSTEM

OMS is hosted on an IBM 4381/3083 supported through alternate diskup capabilities, running under VM/CMS with an OSVS operating system. OMS data is segmented by Organizational Program onto eight 3360 minidisks each consisting of ten cylinders of 1350 blocks, 4096 bytes in length. A central disk of thirty cylinders is employed to store executable code and common data. Each end user of the OMS must LOGON to VM/CMS utilizing a USERID which is validated by CMS for access to the OMS. A second level of least privilege protection is provided by a subsequent, applications specific password requirement. Successful access to the system results in CP linkage to a dedicated system minidisk for each program office. Individual user applications are protected through analysis of userid on an applications-specific basis. General process control and data management is implemented through the central Planning disk which is read-linked to each application disk. Classic database management functions controlled from the central disk include database backup and refresh, source code configuration control, applications disk restore and overall access control as well as specified applications functions which have been delegated from one or more organizational elements.

OMS as installed under VM/CMS is a multi-disk, multi-file network structure with access protection based on user profile and least privilege. Specific datasets are distributed to individual user disks for OMS use; certain CMS files may be read by any valid OMS user.

The OMS, from the standpoint of executive processing is divided into six major segments. These comprise Front end (Security) functions, Organizational applications, Database Mgr functions (including both system initiation and plan integration), financial applications, database query and executive summary reporting. From an applications interface standpoint, OMS is a series of functionally inter-related SAS (tm) applications conducted under both FSCALC and FSEDIT, involving multiple user input/output which is directed and scheduled through interaction between the VM/CMS EXEC/EXEC2 scheduler and SAS. The basic function
of SAS in this relationship is to execute processing steps based on end-use conditions; it is the primary function of VM/CMS to sense these conditions, interpret them, determine their legality within the rules established for system use and relay them to the SAS-executed process.

DMS front-end processing is required for any system application to be performed. Following VM/CMS logon, the user chooses to execute a program which resides on his/her primary (logical A) disk. Once this program is executed, the users disk is replaced with a OMS disk which is associated with his/her USERID only during DMS application processing. If processing is interrupted by hardware failure, power fluctuation, miskeying, etc., a re-IPL of CMS for that user is necessary which returns the user to pre-OMS status. From the OMS disk, read or write authority is granted to a database segment based on user profile. Currently, only read applications are supported for the DMS Interactive database query. Write access authority (Edit Queried Segment) is planned for October 1986.

Selection of the query function causes several CMS files to be copied to the users disk area. These files are automatically removed at EOS. These files comprise:
- ASKL SAS: which establishes the temporary "query" dataset based on user selection.
- ASKL SAS: which interprets upper case to lower case variables.
- QSET SAS: executes conversion of permanent datasets to temporary "query" datasets.
- LID SAS: report generator by OE.
- DEP SAS: report generator by OE.
- LOC SAS: report generator by OE.
- MAT SAS: matrix report generator.
- ASK EXEC: CMS EXEC program which modifies SAS source code, acts, in some cases as a code generator, presents user menus, sets order of execution and checks for user syntax and CMS entry error in the EXEC environment.

From this point, functional interaction between VM/CMS and SAS(tm) is governed via the ASK EXEC. The first selections presented within this environment provide general use information and present a "CONTINUE/EXIT" option to the user. Subsequent screens permit preliminary restriction of data to be accessed (individual OE permanent dataset or full database) and support the selection of query method as previously described. If the "Structured Entry of Criteria" option is chosen, four additional screen types are presented, in sequence, to the user. The first requires the selection of one of seven key variables which provide the basis for data analysis. These represent different characteristics of business activity represented within the database, such as:
- performance responsibility;
- customer and location;
- task/contract identification;
- resource type requirements;
- current performance status; and
- task/program relationship.

The user selects one of these criteria (through a "current performance" entry) as the basis for the query. The second screen displays the last selection made and requests the user to select an operand; EQ, LT, GT, LE, GE, or NE or, alternatively, to quit, back up to a previous screen or restart. The third screen requires the entry of associated value for the variable/operand combination.

Currently, only read applications are supported for the OMS Interactive database query. Write access authority (Edit Queried Segment) is planned for October 1986.

Selection of a statement terminator commences the next stage of query processing; the full expression, as entered, is presented to the user for verification. If an error is detected, a retry option is provided which preserves the preliminary selection of data segment. Once the statement is verified it is redisplayed in SAS (tm) format. If the expression is of adequate length (2 variables) and complexity (a mix of 'and' and/or 'or' connectors) the user may opt to select either an inclusive or exclusive or parenthetical structure and submits the query to run in the SAS(tm) interactive mode.

The actual "cut" of the database segment is accomplished using SAS code. The data, in the case of a full dataset option is brought together into one large file using the SET statement. The system, at that point, processes the query base on the content of the VM/CMS entered expressions or on the basis of statements entered in the interactive environment. Error checking and verification at this point, are the sole responsibility of the SAS system. The SET statements, contained in ASKL and QSET SAS generate a SASLOG which may be browsed or printed at EOS if errors are encountered.

At this point, ASKL and QSET SAS (via a ZINCLUDE statement) are executed and the structured query and an 'endsas;' statement are processed. This point, the dataset is based on query content. From either the interactive or SUBSTRING query options, ASK SAS is executed with a similarly resultant dataset being output. If the interactive mode has been selected, the user is prompted to provide one to six member title keys for insert to the PROC Print portion of the appropriate report generator.

Once the query has been "cut" in the manner the user is prompted to generate his/her finished reports. Four main options are presented, a minimum of two or a maximum of three may be selected in sequence (unless the option to quit or restart is exercised). These options comprise:
- T: Tailor report format/content/sort
- R: Run based on T (necessary only for Matrix reports);
- P: Print Output following browse; and
- N: No Print: Browse only.

Options P and N are mutually exclusive, and
cannot be executed during the same report generation sequence. Option T is the initial selection required for all reports. This option guides the selection and specific tailoring of the report types previously discussed. If the Matrix report is selected, subsequent screens permit sort selection and output field suppression. The OE report generators utilize PROC Print and are summarized using PROC Summary. The Matrix report provides the end user with all possible value combinations of seven key variables. The key variables are lined up in the CLASS statement in the SUMMARY procedure as follows:

```
PROC SUMMARY DATA=QUERY;
CLASS X1 X2 X3 X4 X5 X6 X7;
```

Each variable is assigned a numeric value corresponding to the base2 position which it occupies in the CLASS statement. For example:

```
BASE2 Position| 64 32 16 08 04 02 01
--------------|---------------------
CLASS Position| X1 X2 X3 X4 X5 X6 X7
```

The user is prompted to specify those variables which are to be excluded from his or her report listing. These are read, along with the associated numeric value and applied to a truth table which validates both their accumulated total value along with their individual positional notation. This resultant value is replaced in the subsetting IF state in the following data step for the TYPE variable:

```
DATA QUERY:
SET SUM QUERY:
IF TYPE = (total X1+X2+X3+X4+X5+X6+X7);
RUN;
```

The ASK EXEC takes the positional value of excluded values and assigns those specific variables a value of 0 for the accumulation. This application provides the users only the requested variable combinations. The key variables are subsequently inserted to the PROC Print section of the SAS routine and produced as CMS file output.

At the conclusion of report production, the user is presented with alternative (and mutually exclusive) options for output disposition which permit him/her to vary the type of output (terminal screen versus hardcopy). Additionally, in the case of hardcopy, number of copies and point of distribution are selectable under CP via the query menu. Once disposition of a specific report has been accomplished, additional reports may be run against the QUERY dataset which remains intact until EOS or restart.

Certain additions to this system are planned prior to completion of design and implementation/test. These are geared to improve the flexibility of user/CMS interaction. Currently planned are:

- Edit Queried Segment additions using FSEDIT and possibly FSCALC;
- Extension of current reports to include cost, finance and revenue applications and integration with SAS GRAPH applications;
- Inclusion of mixed inclusive/exclusive boolean statement validation and;
- Extended use of MACRO under Version 5.

These improvements should result in a highly useful tool supporting more sophisticated SAS(tm) - VM/CMS interactions. This may be of particular interest to those users who wish to apply SAS as a data management tool in the CMS operating environment.

CONCLUSIONS

The results of our design effort in this area have been a SASOCMS structure which is truly simple to use; flexible within expressed design boundaries; requires virtually no learning curve (providing that users are minimally familiar with the general operating environment) for effective use and is reasonably self-contained. To date, the designers of this function have managed to satisfy the applications/data requirements of a varied and user community utilizing VM/CMS and SAS(tm) even though CMS is not reputed to be an ideal SAS operating environment. Based on our experience, as evinced by our work to date, we recommend that current CMS facility managers seriously consider SAS(tm) as a data management tool.

Our observations indicate that the approach which we have taken in this development effort, has led to increased acceptance of SAS as a centralized data manager and report generator and has reduced user reliance on alternative, offline methods of analysis which have tended in the past to hamper data collection and integration.

ACKNOWLEDGEMENTS

The authors wish to thank Don Henderson, Roy Patterson, Vince Maleche, and Sharon Graybill of ORI, Inc. for their assistance with this paper.

SAS is a registered trademark of SAS Institute, Inc., Cary, NC, USA.

The authors can be contacted at:

ORI, Inc.
1100 NASA Road One
Suite 405
Houston, Texas 77058
(713) 333-2957