

THE SAS® SYSTEM AS AN INFORMATION DATABASE

Randy Betancourt
SAS Institute Inc. Cary, N.C.

ABSTRACT:

In implementing a successful data access strategy, it is important to recognize there are appropriate and inappropriate ways to access data depending on the nature and distribution of that data and the types of applications requiring access to the data. In some cases it may be appropriate to give users access to the data through views. But, if the views are to a production or transaction-oriented database, the prospect of having 300 users making ill-timed and ill-framed queries can quickly lose its appeal as the database performance grinds to a slow crawl. In such a case, giving users access to separate extract files organized in an information database might be more appropriate.

This paper will examine the role of the information database in enterprise computing, and database features of the SAS System that allow it to be a cost-effective alternative to a commercial DBMS as a source for data required by ad-hoc query and reporting, and decision support applications. In addition, the paper will demonstrate how popular SAS routines can be easily applied to views of operational data in order to "roll up" or summarize the transaction-level data, apply user-friendly formats, perform filtering and merging tasks, and otherwise enhance an organization's raw data assets in preparation for turning that data into meaningful information. The final section of the paper will be devoted to sharing SAS Institute's development direction for SAS information database technology.

HISTORY:

For the purposes of this paper, it is useful to characterize applications into two broad categories. These distinctions are based on the primary use and audience addressed by the application. The first of these is operational applications. Operational applications are on-line, transaction-based applications generally, centered around direct customer order/fulfillment, financial management/control, inventory management/control and the like. Many of these applications are written using COBOL in a CICS (Customer Information and Control System) environment, and update data stores such as IBM's hierarchical database, IMS-DL/I, or record oriented stores

such as VSAM files. These applications are considered mission-critical and are designed primarily for use by the clerical community.

A characteristic of these operational applications includes the need for high-availability by having significant priority over other applications. In addition, the I/O requirement for a single transaction is relatively low, requiring access to a small number of records with any given transaction. While each transaction may involve a small number of records, there may be at any time, a large number of transactions being processed simultaneously. And finally, the transaction may require read, write or update to the data elements in the database.

Over time, organizations have developed a number of these operational applications. Each of these applications was designed and deployed independent of other operational applications. Another common characteristic of operation applications is the lack of consideration for analysis and reporting applications needing to attach to this data. This is not an application design flaw as much as a reflection of the way organizations first began computerization of business functions.

The second application category is decision support (DSS) and executive information systems (EIS). As the name suggests, these applications are designed to augment the decision making process of management by making available detail-level data in summary form. The data needed for decision making needs to come from a variety of operational applications throughout the enterprise.

Business analysts and decision makers began to see how more could be done with data beyond just servicing high-volume transaction processing. Previously, it was the Information Technology (IT) group, with their intimate familiarity with the operational environment, that was used to drive management decisions. This model, which persists today, involves the business analysts needing information to pose a programming request to the IT staff to produce the desired report. In turn, the IT staff who understood the database organization and access methods produced reports using tools like COBOL, Mark IV, RPG, or other third generation reporting tools.

The characteristics of decision support applications involve access to large numbers of records in single or multiple passes of the operational data. Application logic is generated that applies routines reflective of business needs to the detail data to provide additional meaning. From the standpoint of decision support applications, that means taking detail-level data from the operational environment and 'rolling it up' or summarizing it to higher levels of aggregation. These summaries might include adding totals for geographic areas or time periods (e.g., totals for regions or months). This task would also include the application of well-known statistical routines to data to uncover relationships or exceptions.

ANALYSIS OF PROBLEM

While the preceding describes both the operational and decision support model for many organizations, three major problems can be identified with this model. They are:

- ♦ The notion that a single database can serve both the operational high-performance transaction processing and decision support, analytic processing at the same time.
- ♦ The deployment of decision support applications which must contain logic specific to the data access methods required by the operational data.
- ♦ The lack of timely access to operational data for up-to-the-minute decision making needs.

A number of different solutions were attempted to solve these problems. The first efforts were mainly attempts by the IT professionals to better understand the needs of the business, and produce custom reports as demanded by the decision maker and business analysts.

These reports remained difficult to produce because the programs used to produce them had to contain logic that understood how to access the data, as well as logic to produce the desired report. Oftentimes, it was the writing of the program logic to access the data that became the most time consuming aspect of report generation. This was mainly due to the fact that data elements stored in IMS-DLI and VSAM were good for accepting transaction processing elements, but very poor at allowing retrieval of data elements for analysis and decision support applications.

The difficulty in programming these requests, along with the ever-increasing demands for new information, led to new conclusions about aligning information processing technology with the business goals of the organization. Information delivery became the new strategy for IT professionals to better serve the organization's decision making process.

This new strategy means the removal of IT professionals from creating custom reports and applications. Instead, the role of IT is to surface operational data elements into an environment dedicated to exclusive use by business analysts and decision makers. The decision makers then have at their disposal the necessary tools that attach to this new data, providing a wealth of methods for data analysis. It is the extent to which organizations are willing to empower end-users that may well determine overall competitiveness in their particular business.

BUILDING AN INFORMATION DATABASE

The strategies for building and designing an information database should consider:

- ♦ Coordinated access to the various operational data stores along with the appropriate data access tools.
- ♦ A robust and integrated transformation engine for applying some logic to the data from various operational environments before delivery to the decision support environment.
- ♦ The location and architecture of the decision support data repository.
- ♦ The end-user tool set to be used for desktop deployment.

The rest of this paper will be dedicated to describing the feature set of the SAS System in addressing each of these challenges.

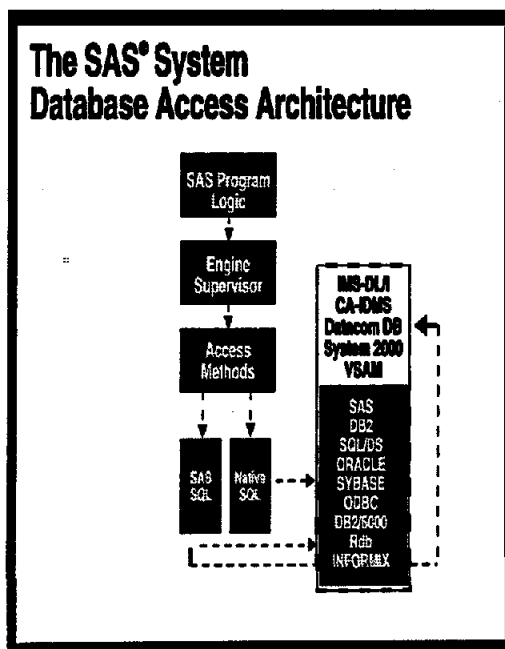
ACCESS TO OPERATIONAL DATA

A strategy in providing access to operational data is the use of a single tool that can attach to a wide variety of operational data stores. The single tool approach obviates the need to master a variety of data access languages. The tool set for the SAS System's data access strategy is Multiple Engine Architecture (MEA). In Version 6 of the SAS System, all data, regardless of its type or form, are

accessed through a set of engines or access methods. These engines provide the framework for translating SAS syntax for read, write and update services into the appropriate database management system or file structure calls. Presently, the SAS System provides more than 50 different access methods for a variety of file types found in different hardware environments. These access methods are a part of the SAS/ACCESS family of software and include access to:

- ◆ relational database management systems
- ◆ hierarchical database management system
- ◆ network database management systems
- ◆ data gateways and standard APTs such as ODBC
- ◆ external file formats such as VSAM
- ◆ SAS Data Sets

With the Multiple Engine Architecture for Version 6 of the SAS System, a single access environment is provided. Furthermore, the SAS System has support for Structured Query Language (SQL). With SAS SQL support and the support for a variety of access methods, SQL in the SAS environment can be used as the data access language for relational as well as non-relational file structures. A pictorial representation of this model is presented below.



In addition to translating SAS data management syntax to the data access language for the target data store, the SAS System provides a method for passing SQL statements native to the target RDBMS. This is particularly useful in those instances where the SAS internal SQL processor cannot optimize queries for the target RDBMS or one wishes to support SQL extensions provided by the RDBMS. Through MEA, users of the SAS System have a single and consistent view of enterprise data, regardless of its access method or location. These access methods can surface operational data in two forms: as views to data or as extracts from their native form into SAS organized data.

SAS/Access views are similar to the traditional RDBMS views in that they do not contain physical data. View descriptors, as they are called in the SAS environment provide three basic functions to accessing operational data:

- ◆ provide the path and instructions for SAS to access the target data source and may include data management specific logic
- ◆ provide name mappings from target resource names into names conforming to SAS conventions.
- ◆ Provides data type mappings from target resource into data types supported by the SAS System.

Advantages in using of SAS/Access views to surface data are:

- ◆ reduce data redundancy
- ◆ provides access to current data
- ◆ requires little storage
- ◆ allows the combining of dissimilar data sources, between and among different hardware environments
- ◆ can be defined as subsets of the original data
- ◆ can be defined as supersets of the original data

As part of the strategy for accessing operational data, many organizations have experimented with providing SAS/Access views to their end-user community with varying degrees of success. A more practical model may be to allow the IT group to build and access view descriptors as a means for surfacing relevant data into an environment different from the operational environment and one designed exclusively for decision support processing.

The following scenario illustrates an approach for using the SAS System to attach to and migrate operational data into a decision support environment. To begin with, the one-time effort of building the SAS/Access view descriptors is

required. SAS/Access descriptors can be built either interactively or in batch mode. Once built, SAS/Access descriptors need no additional maintenance, unless the form of the target data source is altered. Next, a batch job is scheduled to initiate a SAS job step that uses the view descriptors to attach to the operational data. This is also where we have an opportunity to enhance data by combining it with other data, and perform additional data management logic. The result of this step is to produce one or a number of temporary SAS data files. The next job step then executes the syntax used by SAS/Connect software to instantiate a SAS session in a remote environment. Once the two SAS sessions are connected, then a download of the data can be formed. The final form of this data in the decision support environment can be either be SAS data set form or data managed by a RDBMS. See the section below on *Data Repository Architecture*.

DATA TRANSFORMATION ENGINE

In addition to being able to access operational data, it is probably the case that some pre-processing of the data is in order. After all, reporting and analysis activities are designed to provide a broad view of what the data represents. It is seldom the case that a report will be composed of displaying all the detail level items. Similarly, moving all of the detail level data from the operational environment into the decision support environment rarely, if ever, makes sense.

From a policy viewpoint, it may be difficult to convince management and business analysts such a strategy makes sense. The common refrain heard is "... but I want access to *ALL* the data." This is where it makes sense for those responsible for data migration strategies to examine closely what end-users are doing with the data they use today. In nearly every case, their programs will contain data summarization and reduction tasks. To the extent these data reduction tasks can be identified, provide clues to what transformations are appropriate as data is surfaced to the decision support environment. In 80% of the cases, end-users' requests can be satisfied with a static view to data already summarized, and 20% of the time, some new view of the data may need to be formed.

The strategy is to provide access to operational data, with some data management logic already applied. In an ideal situation, the end-user tool sets that access data in the decision support environment would never need to form

any data management logic. Instead, all data management logic will have either been formed ahead of time, or will be stored as part of the decision support data repository.

The SAS System provides a large number of tools for data transformation. They include:

- ability to open multiple input files simultaneously
- ability to open multiple output file simultaneously
- perform look-ahead reads
- perform table look-up logic
- sorts that can use a variety of character sets and collating sequences
- SQL for Groupby, Orderby, and summary functions
- data step programming with arithmetic, trig, random number, probability, and string manipulation
- functions
- PROC SUMMARY for grouping by classification values
- PROC MEANS for collapsing numeric data using a number of different univariate statistical methods
- PROC FREQ for one-way, two-way, and n-way classifications
- multivariate statistical methods for numeric analysis

DATA REPOSITORY ARCHITECTURE

The model used by most organizations for providing enterprise data access has been the attachment of selected Window's tools directly to the operational data stores. With desktop users allowed to formulate SQL queries through point-and-click menus, the likelihood of creating an ill-framed query is inversely proportional to the skill level of the end-user. That is, the more unfamiliar one is with SQL, the greater the likelihood of producing non-sensible, run-away queries. If these non-sensible requests are allowed to attempt retrieval from production OLTP data in the operational environment, then OLTP service objectives can begin to degrade, not to mention network overload. By maintaining the desktop perspective for end-users, organizations are looking at not only segregating operational and decision support data, but also segregating the hardware environments where the different data stores are located. Rather than allowing the desktop tool set to generate queries which run directly against the operational data, these queries are executed against the data repositories which often reside outside the hardware environments containing the operational data. Many organizations are moving to a three-tiered approach. Tier

one is the host environment where existing high volume transaction applications continue to execute. This is also the source for most of the operational data. Using tools for data access and transformation described above, many organizations are electing to build their data repository for decision support in decentralized environments such as UNIX or with high-end Intel processors running network operating systems such as Novell or Banyan.

In agreeing to make operational data elements meaningful for data analysis outside the operational environment, an issue to be addressed is what form should the repository take. Before attempting to answer this question, it is useful to review the requirements for a data repository. The fundamental purpose of any RDBMS is to provide a repository for data. The RDBMS is responsible for storing data elements and restoring them upon demand. Users are shielded from the details of storage and retrieval, thus allowing the end-user to concentrate on the analysis and presentation components of his or her application.

Using a model presented by Billy Clifford, SAS Institute Database development staff, the column on the left describes the feature set found in the traditional RDBMS environments, while the column on the right describes the SAS component for providing the particular service.

Service	SAS Feature
File Management for create, populate, delete & backup databases	Data Step, SQL, CPORT, UPLOAD, DOWNLOAD, Procedures
Data Inventory services for information about databases	DATASETS and CONTENTS procedures
Query Processing to retrieve, filter, organize, present and display data	Data Step, SCL, PRINT, FSEDIT, FSVIEW, SQL, FSBROWSE, & REPORT Procedures
Update Processing to change existing data or add new data	Data Step, SCL, SQL, APPEND & FSEDIT Procedures
Relational Data Model to provide abstracting of data elements independent of application logic	SAS Data sets are rows columns subject to standard SQL manipulation

With these services viewed collectively, and the need for the abstraction of application logic from data access and

management processing, the SAS System is clearly in the same class as the commercially available relational database management systems with respect to these services.

Many of the commercial RDBMS offer advanced services such as referential integrity constraints, audit trails, roll forward, two-phase commits, transactions with rollback, and high volume transaction processing. These advanced features are essential requirements for data repositories in an operational environment. However, for a data repository in a decision support environment, such advanced features are not necessary, and their presence may even be a source of unnecessary overhead, not to mention costs.

DESKTOP TOOLSET

The final component of an integrated information delivery scheme is the selection of the desktop tools. Over the past decade, organizations have either by design or through a laissez-faire approach acquired large numbers of desktop workstations. Historically, these workstations have been used to address office-automation tasks using personal productivity tools such as word processors for document management, spreadsheets for simple economic modeling, and electronic mail for the dissemination of information. As these systems have matured with advances in microprocessor performance and better human interface systems, organizations see an opportunity to provide a larger percentage of its professional workforce access to enterprise data and thus allowing the widening of the decision making process.

Many organizations have developed internal standards for the selection and deployment of desktop tools. The following is a partial list of the criteria commonly encountered.

- ◆ Microsoft Windows compatibility
- ◆ applications enabled through Window's GUI
- ◆ compatibility with corporate network standard
- ◆ compatibility with corporate middleware standard
- ◆ attachment to various RDBMS sources
- ◆ generation of SQL for data requests
- ◆ applications development front-end tools
- ◆ object-oriented attributes
- ◆ data sharing between applications

Over the past several years, a major strategy pursued by SAS Institute is the development and support of the SAS System for desktop environments, notable, the Microsoft Windows environment. Each of the aforementioned criteria is attributes of the SAS System. Some of these criteria, such as SQL support are a portable feature of the SAS System, having been supported since the introduction of Version 6 software in 1989. Others, such as support for OLE and DDE are host specific extensions that are standards for the Windows environment. It is beyond the scope of this paper to describe these features in detail, except to point out that from a point of view of organizations seeking standards for desktop software, the SAS System feature set has been designed to meet these needs. Many new features and enhancements to the existing feature set are the goals for Release 6.10 of the SAS System. This release is targeted exclusively for the Windows environment and is scheduled for general availability in mid-1994.

FUTURE DIRECTIONS

A major step toward expanding the use of the SAS System as the decision support repository is the opening of data managed by the SAS System to other applications. With the SAS System has always been to the ability to surface SAS data elements for use by other applications. However, for the SAS System to surface this data, involved the direct execution of SAS along with instructions on how to form the data. SAS software has always been able to form the data in any shape or format needed by the requesting application. Up until now, the model for sharing SAS data has not been direct and transparent.

Using the Microsoft's ODBC specification, it will be possible for non-SAS applications in the Window's environment to request direct access to SAS managed data as well as data from other sources accessible by the SAS System. The Windows client application can access either SAS data in the local environment or SAS data in some remote environment. For local access, a new SAS ODBC driver will be packaged with Base SAS Software, Release 6.10 under Windows. The ODBC driver will allow local ODBC-compliant applications direct and transparent access to SAS managed data.

For remote access to SAS managed data sources, extensions to SAS/SHARE software will be made in all supported environments to receive requests from other non-SAS applications using ODBC-compliant SQL. This extension, known as SAS/Share*Net will reside in the

remote environment, and act as the listener piece for incoming ODBC-compliant requests. Once the request is received, it is then forwarded to the SAS/Share server for generation of the appropriate results set. This means that not only are data objects managed by SAS software accessible, but any other data sources to which SAS software has an access method to.

An ODBC driver from SAS Institute will be needed in the Windows environment. This driver will contain the necessary connectivity to support network access, such as TCP/IP to communicate with SAS/SHARE software executing in remote environments, along with the requisite routines to convert ODBC-complaint SQL into SQL syntax understood by SAS's own SQL processor. In addition, server side support for an ODBC access method is planned for the next release of the SAS System under Windows NT scheduled for delivery at the end of 1994.

Another area of continued development effort is in the area of SAS/ACCESS Software. Some of the development priorities include:

- ◆ client-side support for SQL Server for Windows NT
- ◆ enhancements for PC File formats to include.WK1 and .WK3 support for Win32, Windows NT
- ◆ and OS/2
- ◆ client-side support for ODBC for Window's NT
- ◆ server-side support for ODBC for Windows NT
- ◆ client-side support for Oracle under OS/2
- ◆ client-side support for Oracle under Win32 and Windows NT
- ◆ investigate IBM's DB2/2 client application enabler support
- ◆ client-side support for ODBC in the Apple Macintosh environment
- ◆ support DATA step interface to IMS/DL-I under MVS
- ◆ support Informix for Solaris, HP, and AIX environments
- ◆ begin development for DB2/6000 in the AIX environment

CONCLUSIONS

As organizations begin to re-architect their decision support environment, careful attention should be paid to the service set offered by the SAS System. This paper is an attempt to make end-users and decision makers aware of the adaptability for decision support and applications development in a wide range of hardware environments.

The traditional strengths of the SAS System have been to provide strong data management tools of its own, as well as the ability to access a wide range of data managed by other software. By supporting industry standards such as SQL, as well as emerging standards such as ODBC, the SAS System is well positioned to continue its leadership role as a viable solution as an information database to support end-user and management decision making.

ABOUT THE AUTHOR

Randy Betancourt is a Program Manager for Enterprise Computing at SAS Institute Inc. He can be reached electronically at sastrb@unx.sas.com.