SAS Institute has set an ambitious course in rechristening SAS as the Strategic Applications System. But just the act of leasing of the SAS® System is not enough to provide a strategic pay-off. The term, strategic system, is still used by some in Information Systems management to refer to transaction systems critical to the success of the enterprise. Automation support for strategic planning on the other hand may be seen as nothing more than requiring a spreadsheet and presentation software to support a largely intuitive process. High level management of month-to-month progress of the enterprise is often no more than a custom application feeding data to a spreadsheet-based Executive Information System (EIS). The EIS might have a pretty face and flashy functionality, but underneath the information architecture to support a robust strategic planning and monitoring function does not exist. This paper discusses such an architecture and how the SAS System RDBMS and SAS System tools can play a vital role in it.

Strategic Applications Systems Further Defined

Strategic is one of those words that can fast lose its meaning when applied to business applications. It is synonymous with crucial, essential, and important. Any application that can't use at least one of these adjectives would not retain the support of management for long. With that caveat let us assume for the sake of this discussion that the terms strategic and tactical can be used to describe relatively equal levels of importance, but from different perspectives in time. In business as well as war, tactical matters are usually regarded as being short term performance issues with limited scale. Long term, enterprise survivability is a strategic concern. Comparisons of "how to do" versus "what to do"; "part" versus "sum of parts"; and "structured" versus "unstructured" tasks are useful in differentiating between tactical and strategic. But when profits fail, cash runs short, and projects without immediate payoff are canceled, tactical issues can take on a strategic nature. On the other hand, healthy enterprises inevitably have greater time horizons, place a greater value on capital, and show a clear delineation between strategic structuring and tactical performance.

If we were to evaluate most business applications today as to whether they were strategic or tactical in nature using the above criteria, most would be considered strategic only on an exception basis. In other words their failure might threaten the enterprise, but their success would not necessarily mean long term survival. So what would truly be a strategic application? We are just finding the answers to that question. The development of strategic data applications might well be the single largest growth area in enterprise computing in the 90's.

An application can be described as a program or collection of programs that are used to process data in order to support a business function. Information is broken down into electronically stored data that is gathered and maintained via the application in order to support specific business requirements that are usually operational in nature. These applications are especially useful in enforcing "how to do things". The scope of tactical applications is encapsulated within "how", and human intuition, not a piece of software, still determines "what to do". For some time our profession has been aware of the need to provide tools in support of intuition, but the data needed for this is often collected by rummaging through the garbage left over by tactical applications. Few, if any, applications can be found in most organizations that support strategic planning as their primary function. (Good Executive Information Systems help management to monitor critical factors of success but fail short in supporting "what if" analysis beyond a scope defined by the application data centered on those factors.) Even if many tools (how many languages does the SAS System encompass?) are gathered under one system — is this enough if we don't have the right data?

On the technical side of things we tend to regard systems as foundation software, as in operating systems or data base management systems, on which we build applications. However, the terms, system and application, are frequently used in place of one another in the business world. This makes it tricky for SAS Institute to market SAS as the Strategic Applications System. Certainly one thing that will make it easier for SAS Institute and SAS users will be to recognize certain characteristics of strategic applications and the difference between planning support and tactical management. One such difference is the way data is managed.

Thinking In Conceptual Data Terms

Much is made today about being able to get to data that is maintained by different applications in various formats. Tools, such as SAS/ACCESS® software, help us get to data when the data step is not enough. But after we get
there, more often than not, we find that the information we need is not being maintained in the way we need it. For instance, we wish to control, but we need n for an nem turnover frequency study. We could possibly get a snapshot of our application that is designed for real-time inventory management. However, if we needed n, we could not get what we wanted on an ad hoc basis. In another situation, can we tell how many dollars were transferred from active checking accounts versus how many new deposits were made when we offered a new type of account featuring free checking to seniors if periodic account history is not maintained? Making changes to these transaction systems to support such ad hoc data needs after the fact is like shutting the barn door after the horse has gotten out. Such issues would come under the realm of application design, not tool selection. If we think of data produced by tactical applications in conceptual data terms, we will quickly find that data often has value beyond the scope of the application, if it can be managed beyond the scope of the application. Cross organization and consolidated historical information have been two areas rarely addressed by Information Systems groups.

In the early '80s a major airline set a new goal structure into place for its reservations sales representatives. The main thrust of the system was to increase sales per 100 calls as opposed to an earlier objective that emphasized calls serviced per hour. Costs per call soared as reps took the time to book more seats, and reservations managers throughout the system complained about how the new system decreased productivity (as measured in earlier terms of calls per hours). However, the regions that performed well under the new goal system did show higher load factors with greater revenues which more than offset the increased costs in reservations. Although there was one other significant factor that year that certainly influenced revenues (the introduction of a frequent flyer program), the company did show a nice increase in profits that year that was associated at least in part with the new reservations goal system. Cross application information was maintained outside of the applications themselves over the course of that period. This data was kept in a SAS data base and at the end of the year PROC GLM and PROC PLOT were used to graphically show the relationships between sales, reservations costs, and load factors. Although line managers still worried about an apparent loss of productivity on the local level, the positive effect of the new goals was demonstrated at the strategic level.

Information Management as a Process

Over the course of the last decade a few businesses have begun to regard data as an asset. Information Data Bases (IDB's) such as the one developed by a major NW bank were organized along the lines of dual data bases. IDB master files were loaded on a daily, weekly, monthly, and yearly basis into files that were tuned for reporting. Annual files were kept up to seven years and monthly files were maintained for 18 months so that comparisons over time could be made. Many of the reporting requirements for each application were met by using IDB's. Perhaps the most important feature of this environment was the ability of systems programmers to manage transaction, production batch (of which the IDB load was a part), test, and ad hoc query environments as separate entities. Usage statistics were maintained on each file so that the value of a given IDB could be determined by those clients who used it the most. Security was tightly controlled and the integrity of the files was maintained through production batch updates plus read only access by clients. Over this period the organization offered one of the best returns on investment in its market area. However, the IDB's reflected a number of separate transaction systems and as such contained a great deal of redundant data. Demographics were repeated for each account that a given customer maintained. Cross application studies were supported only in a very rudimentary way. And the data was maintained primarily in proprietary DBMS's with limited tool access. The upgrade path for the IDB environment became pretty obvious not only to those responsible for IDB maintenance, but to others thinking about implementing similar dual data base environments.

Among those who have recognized the value of information data bases was IBM Inc. who advanced the idea of an Information Warehouse last year. The Information Warehouse includes many of the features of dual data base environment described above. An enterprise wide, historical perspective would be facilitated by an integrated design. IBM relational DBMS's would provide the storage and retrieval of information and an open SQL based applications programming interface (API) would encourage business partners to create interfaces so that clients would not be restricted to a limited set of data analysis and reporting tools. SAS Institute was one of the primary partners to support IBM in the Information Warehouse concept. SAS/ACCESS, when coupled with the other pieces of the SAS System and a well designed Information Warehouse, would put more information at the disposal of analysts and management than at any time in the past. The Information Warehouse approach could obviously be developed on in other operating systems such as UNIX or VMS with other DBMS's such as Oracle®, or RDB®, but the heart of such a framework
would still be data maintained within a Strategic Information Architecture.

Rescue from Data Redundancy and Other Myths

Concerns often expressed regarding the wisdom of implementing a Strategic Information Architecture include the risk of not using primary data (i.e. data redundancy), data is not current, resources used to access this data cannot be managed, and storage is costly.

The issues of primary data and currency are intertwined. Is the balance in account z at this moment the same information as the balance in account z at the end of last month? And can the same report that was run yesterday against account z be replicated? What is the value of the real time average of all accounts of the type z versus the comparison of an average of all accounts of the type z from period 1 to period 2? The answers to these questions points to the kind of fuzzy thinking about data that we all suffer from on occasion. The time value of data is often different to a business function than it is to the enterprise. Periodicity allows the user dealing in strategic issues to use time-series tools against an historical data base and to replicate his or her analysis, something that is usually impossible with a real-time application. Data stored within an Information Warehouse may originate with a tactical application but should be considered now part of a different information structure. Alternative schemes to the Information Warehouse approach that rely on such features a repository of information about transaction files and maps on how to get there should be by now, regarded as inadequate.

As the data in an Information Warehouse takes on some distance from its source application, there is a danger in equating tools and applications. Several applications lumped under a single acronym can obscure cost/benefit links and result in generating bogus performance issues. The tool should not be confused with the business function especially when some of those functions are accomplished with COBOL or C and are named for the business application they serve while others functions are supported with SAS and are named SAS this or SAS that. SAS is really a tool set for building applications and analyzing data. In this case, system is equivalent to tool set. Keeping this distinction in mind is important.

The issue of cost is usually not an issue within organizations that have a client oriented resource accounting system. In this case the client manager can make the call as to whether the cost to the manager's organization for this service is below, at, or above the benefit it brings. The Resource manager should be off the hook. A charge back system can provide support for the system by showing that non-MIS managers are willing to pay for it. However, not all enterprises have an effective charge back system. The value for an information architecture can also be enhanced by using it for applications reporting where possible. Successful, high impact ad hoc uses need to be made visible, and SMF Type 14/15 records defined with Merrill's MXG® software can be used to track database opens and closes (in MVS) by user if additional data is needed to gain organizational support. The key to all of this is allowing the client manager to tie the use of distinct pieces of information to a business activity. The processing costs should be part of that use.

Physical Design for Query Optimization

Information Warehouses by their very nature require large amounts of machine resources. Therefore it is important to plan well and choose the most cost effective platform possible. Information Warehouses are not good candidates for a decentralized data base structure. This does not mean that a sound client/server environment would not be cost effective, as long as the server is centrally managed. SAS V6®, DB2®, Oracle®, or Teradata® make good candidates for a relational data base server. In a large enterprise with a heterogeneous tool set, an open DBMS such as DB2 makes sense. This will maximize the value of the data to all clients. SAS users have SAS/ACCESS and other software vendors are likely to have interfaces to DB2. For large volumes, a data base machine such as Teradata can be used to supplement or replace DB2 (SAS/ACCESS for DB2 works with the Teradata machine for query operations). Using SAS for the RDBMS makes sense in smaller shops with a more homogeneous tool set. This approach obviously will minimize support costs and take full advantages of all of the features of the SAS RDBMS such as compression, support for ANSI level 2 SQL, and a host of data management tools.

Data modeling tools have been a great boon to the implementation of transaction applications on relational DBMS's. Since an Information Warehouse relies on other applications for data, it would seem that there would be little need to do further modeling work. However, it is the business functions that are addressed by the application that are embedded with the data structure. The Information Warehouse is based on a different business function. The physical design of the Warehouse will most likely end up to be quite different than that of the application. Doing a logical model gives us a check that the new structure makes sense.

One issue that modeling should address is the fact that there is no front end application that guides a user of an Information Warehouse to just the right place for just the
right data. Information about applications data needs to be included in the design. This data about data or meta data should at least include locations, aliases, sources, structure, usage, and cross references. Tables should also be included that assist in the administration of the Warehouse such as security and access tables.

When it comes time to do the physical design for the data that comes from transaction applications, it is often tempting to just copy in the transaction design. Resist the temptation. Account query flexibility in an on-line application usually means that small tables will be used to maximize the number of rows per page in order to maximize join performance. Multiple on-line updating means that indexing will be kept at a minimum -- often limited to primary and foreign keys -- and that the structure will be normalized at least to 3rd normal form. Decision support and other ad hoc query activity is not so predictable. Since there is no on-line update capability within the Warehouse, data can be denormalized into fewer if larger tables and indexes can be used generously in order to improve access to subsets of the data. Often, in many RDBMS's, frequency queries can be satisfied by just using the index. In other cases, join activity has much greater performance implications when analysis requires multiple table scans. Therefore large tables should be designed to meet as many foreseeable query needs as possible without requiring joins. Where possible, table size should be reduced where it makes sense from a decision support perspective. Summary tables which provide views of a given transaction from several different perspectives save machine time and make information more readily available. A good example of this sort of organization can be found in Legent's MICS® data base design. A job in MVS can be tracked from user, resource, time of execution, and/or program point of view. Of course summary tables and extra indexes take disk space. But in the scheme of things that is the cheapest alternative to increased processing.

Update strategies include using the DB2 bulk load facility with the replace option to processing application transaction files during off hours if complete loads are impractical because of size. This might be called the mini batch approach. It requires the maintenance of a master file with inserts originating from it within the Warehouse.

Enhancing Ad Hoc Capability Through Planning!

The SAS System has a number of uses outside of a Strategic Information Architecture, but it is this area that is getting the attention. Before a favorite product becomes attached to an effort that is highly visible and has the potential for a costly failure, it would be wise to spend some time reading and planning for it. Building a Strategic Information Architecture is non trivial.

Further Reading


The following article is a good description of the Data Warehouse concept:

Inmon, William H., "Data Warehouse", (DATA BASE MANAGEMENT, February, 1992)


Inmon, William H., "Examining the issues of the Dual Data Base", (Date Resource Management, Winter, 1990)

This article puts forth IBM's strategy for the Information Warehouse Concept:

Orr, Ken, "IBM HAIL FOR '90s: "FREE JAILED DATA", (SOFTWARE MAGAZINE, July, 1991)


Von Halle talks about integrating Data Base Servers with 4th GL Tools:


John Zachman got the whole architecture thing going. This article is worth looking up:


SAS and SAS/ACCESS software are registered trademarks of SAS Institute Inc, USA and other countries DB2 software is a registered trademark of International Business Machines Inc, USA and other countries RDB software is a registered trademark of Digital Equipment Corp. Inc, USA and other countries