

Responding to Change: Providing "Data Warehouse" Services as the Sources of the Data Evolve

Kathe Richards

ITS, Technical Environment Support Analyst

Texas Comptroller's Office

Abstract

Change is the rule in these days of evolving architectures, strategies, and tools. Consider what would happen if a significant portion of the application that supports the business processes that supply your data changes from traditional mainframe based to some flavor of client/server. Or what would happen if the equipment on the desk of your primary data users changes from a dumb terminal to a box that has significant processing power of its own.

These are common scenarios these days. Continuing to provide the service levels expected by users of data warehouse services can be a challenge. What follows is a discussion of some potential pitfalls in this situation and "real life" stories about dealing with them.

Who We Are and Where We Came From

The State Comptroller's office has three constitutionally mandated roles: keeping the books, collecting the taxes, and telling the Legislature how much money they have to spend. Because Texas is a pretty big state, this involves a lot of data and a lot of processing, to use the old fashioned terms. What we term our "production systems" run day and night.

A good bit of energy is invested on an ongoing basis to make these systems run "better, faster, and cheaper."

We are now in the implementation stages of a major upgrade to our production tax systems. We have interviewed, analyzed, correlated, documented, developed business rules, structured, and restructured. We have prototyped, demonstrated, coded, tested, and solicited end user interest. We have engineered and re-engineered. We are now converting, installing, acceptance testing, training.

This upgrade involves more than a few major changes. For one thing there is a major change in where the data lives. We are converting massive databases from VSAM and System 2000 to DB2. We are going from essentially flat and hierarchical structures to relational tables. In many processes, we are going from batch update to on-line update.

It also involves some major changes in technical architecture. Some of the processes are being developed in a client/server environment. This, of course, means new kinds of equipment both on the desktop and throughout the network. There are new gateways and networks, operating systems and development tools, development strategies, standards, procedures, life

cycles, and environments all learning to coexist with our old “legacy” systems and architecture. The complexity of the situation has increased. The learning curve is formidable. And the budget is skinny. And the time frame is tight.

Meanwhile - -

There are a lot of systems that support the Comptroller’s mission that are not formally considered to be “production” though they are certainly a vital part of our operations. Many of these systems are the creations and responsibilities of workers in the operational divisions of the agency. They often depend upon production data or abstracts and summaries of production data.

Among the many users of the tax data in particular, which is the data that is affected by our massive development project, are:

- The Research and Revenue Estimating divisions who use aggregated gross sales information and other tax data to contribute to the analysis of economic trends in Texas and produce the revenue estimate for the Legislature;
- The Open Records and Disclosure section, who provides information to the Legislature and the public in response to open records requests;
- The Audit division analysts who use tax data analysis as part of the information they take into account in deciding how to schedule and deploy their staff;

- The Enforcement division analysts and management, who use this data as input in judging the effectiveness of collection activities and spot trends in compliance;

- The Tax Administration division staff, who use it to selectively distribute information to taxpayers through mailouts and other contacts and estimate work load factors;

- The Local Government and GIS staff, who distribute information to local taxing authorities such as counties and cities;

- The Revenue Administration divisions who monitor and manage exception trends in tax data.

These tax data users are all users of the SAS System in our shop. And this partial list doesn’t include “utility” users like the systems people who use it for performance monitoring and analysis of SMF data, the security administration people who use it for reporting, and system development people who use it for just about anything, including in the case of our project, part of the data conversion effort.

It is clearly important for us to be sure that all these people continue to be able to find and use the information that we were in the process of transforming so radically.

In the early days of SAS at our agency, like at most places, the kind of “management reporting” that was done was mostly a “do it yourself”, often ad hoc, and sometimes underground effort. It wasn’t always appreciated by the professional information services staff.

As the industry became more aware of this kind of activity and as it became more apparent that it wasn't going to go away, the "Information Center" trend became popular as an effort to support it. The idea was to provide staff and support to people doing this kind of work.

In our agency, policy evolved to the point that outside users, meaning mostly SAS programmers, had read access to any production data that was not specifically confidential (like certain types of payroll and personnel data.)

Over time, the Information Center idea has progressed to the point that people now talk about "Data Warehousing" and similar concepts. The idea here is that it would be helpful to provide data access in a form that is suited to the needs of the community that uses summary level data and does selective ad hoc analysis.

This appears to be driven to some extent by some developments in architecture. The arrival of relational database technology allowed us to support business functions with elaborate data inter-relationships that were previously restricted by the flat file architectures. This makes the data enormously more complex, however, and the rules by which it is inter-related much more difficult to find out. Documentation became essential. Database design also began to be based upon fitting the business processes that generate the data and use the data rather than batch processing considerations. This made it more and more difficult for people who use the data after the process is completed, such as our many analytical

users, to find what they need and put it together.

The more recent developments in client/server technology is having the effect of further deconstructing information sources. We now have data anywhere (as well as processing anywhere), or perhaps more accurately, data everywhere.

"Data Warehousing" and all the related concepts seem to be a response to the increasing complexity of the environment.

The New "Challenges"

What We Knew - -

So here we were in the midst of a gigantic project that affects most of the data in one of the major agency production systems. We knew that we couldn't put anybody out of business. We needed, of course, to support all the agency business processes related to tax. We also needed to be able to take advantage of advances in technology and architecture. So in addition to bringing up the system, we needed to put in place a structure the agency analytical information users could work with.

We didn't have much in the way of technical resources to apply to this effort. By the time that the design of the project had jelled to the point that coding could begin, every technical warm body on the project had plenty to do. We knew we would have to start as early as possible, but didn't want to start until the data itself was pretty firm. The plan that

evolved included the following strategies.

- Build a Query Database

The database administration and system resources guys were concerned about the performance impact of summary level queries and analysis on the interactive processes. The DB2 databases were designed to support business processes and not the table scans necessary to do typical analytical tasks. We were unable to define specific summary information at this point since our users are so diverse and no one, of course, had any experience with the new system yet. The database administrators proposed a Query database that would begin as a duplicate of the production database in structure, with suitable indexes added to support our processing, that would be refreshed regularly (daily), and could be modified by adding summary tables as the requirements became more apparent. This was proposed the user community and they found it to be acceptable.

- Introduce the changes to the technical user community as early as possible.

We did this with a series of informal contacts and presentations that described the changes, the proposed schedule, and some strategies for conversion.

- Provide classes and technical assistance.

This included classes in SAS, PROC SQL, and QMF, as well as sessions in which analysts discussed the differences between the way the processes and information were and would be. We were able to call upon the services of

some of the user community analysts to help with this.

- Keep lines of communication open.

We have a group of technical staff members who can cover most concerns available for assistance, coordinated by a single point of contact. We have weekly meetings and e-mail lists for disseminating information and gathering feedback.

- Provide easy access to information.

This included information on the structure of the databases, the changes to the processes, and the cross reference between where data was and where data was going. This is more complicated than it sounds, because of course, the database was changing (usually subtly, at this point) as development proceeded. The conversion team built the cross reference and made it available on the network server.

Some Things We Found Out - -

- No matter how early you start, it isn't going to be early enough.

We started with the planning and information sessions the moment that the situation was stable enough to predict what was going to happen and when. We worked diligently to meet training schedules. Everyone worked diligently to meet development schedules. Things were going along pretty good, but little things began to slip. Pretty soon there was more and more to do in the last few weeks. And the analytical users had their own obligations. So with the

cutover date looming, there were still quite a few very big things outstanding. Then some unexpected system problems fell into the mix and we ended up scrambling to get production in place.

- The "data warehouse" users are at the end of the line.

Production comes first. If production doesn't run, there isn't any data to analyze. So as the inevitable program problems and conversion data problems were uncovered, the promises we made to user analysts about when their data would be ready began to broken. We had fall-back plans, but even these didn't cover everything. Unfortunately, there's not a lot that can be done about that except beg for patience, prioritize the problems, and take care of what you can.

- The processing overhead is a lot bigger than we thought. In fact, everything is a lot bigger than we thought.

We did pretty extensive research on what kind of resources this effort would take. We knew we'd need a lot of processing power and a lot of data storage. We knew that DB2 was going to process a lot slower than the access methods we've used in the past. We even had quite a bit of experience with using SAS and DB2. When we began to see the amounts of data involved and the amount of processing involved, even after indexes were applied and database tuning techniques exercised, it became clear that alternative strategies would be needed. These varied according to the specific circumstances and took a lot of time to analyze and develop.

- The users themselves were surprisingly well prepared.

They know what their code does and they didn't have any problems with letting us know what they needed. They were cooperative and patient for the most part. It's scary, though, to not know if your bread and butter jobs are going to run next time. It's even scarier, on test runs, to see jobs that used to run in a few minutes start taking a few hours, or even worse, be terminated by the DB2 governor.

- The interface problems are the hardest.

DB2 doesn't speak SAS. And SAS doesn't speak DB2. If you need to do something that requires several tables, about the only choice you have is to use SQL pass through, especially with the table sizes we are dealing with. This means that you have to learn to use both SAS and DB2's brand of SQL syntax and know where to use them. Using SAS access descriptors works for simple one-table queries and small stuff, but a lot of what we are now seeing is very complex and large.

There is some help available here through SAS ASSIST. In order for it to be helpful, however, we needed to have some administrative assistance from the DBAs to generate some catalog information that SAS uses to make the interface work. The DBAs were deeply involved with conversion and solving system problems, so, while they were willing enough, this task was put in the prioritization queue with everything else.

Some things we haven't found out yet

- How to use SAS well in this environment.

We're still working on it. We've got some really good technical people and some very good user analysts. And when we get some time to look at the situation after the dust has settled, we'll find out a whole lot more about what we had no way of knowing before we started.

- How to use the other new tools, architectures, and opportunities in this environment.

Client/server offers a lot of opportunities, as well as challenges. In general, we have learned a lot, will be learning a lot more, and most of it seems to have to do with the *challenges* so far. I'm sure the *opportunities* are out there somewhere.

What I would have done if I knew then what I know now - -

- Take early retirement? No, probably not. The experience has been invaluable. The days have gone fast. And ultimately, we will have systems and information that are much more closely aligned to our business processes. I'm sure they will be far from perfect, but we will have made a big step in a positive direction.

I am reminded of a report from a Gartner Group conference some months ago on a survey they did of users and technical people who had successfully completed a client/server development project of a reasonable size. They asked the users if they were satisfied with the results, if it met their business needs, and if they could validate savings or revenue flow attributable to the development. A heavy majority of these replies were positive. They asked the technical people if it was difficult. A heavy majority of these replies were to the effect that it was unbelievably difficult. Everybody was asked if they thought, in retrospect, it was worth it and would they do another one. A large majority of both (even those who had rated the project as a failure) said yes.

It takes patience and persistence, though. And a sense of humor doesn't hurt, either.

SAS, System 2000, and SAS/ASSIST are registered trademarks of the SAS Institute Inc., Cary, NC, USA.

Other brand and product names are trademarks of their respective companies.

TITLE: The M1A2 Tank Battle

AUTHOR: Judy Rosten
TEXCOM, Fort Hood, Texas

MEDIA: 2' by 3' chart

PURPOSE: Graphically analyze and display an instrumented force-on-force tank battle. The chart portrays each combatant, each engagement, over time and distance.

DATA: Data input to the analysis comes from Realtime Casualty Assessment (RTCA) instrumentation and realtime positioning instrumentation.

TOOLS:

1. The Statistical Analysis System (SAS) computer software is the analysis tool. This software includes data manipulation, data analysis, and graphics capabilities.
2. Computer hardware tools included a 486/66MHZ/16MB personnel computer and a drum plotter.

USE: The tank battle chart facilitates visualization and understanding of the actual field events. Instrumentation generates gigabytes of data. This data exists on magnetic tapes and on stacks and stacks of printouts. The data volume without graphic display made analysis very difficult.

SPECIAL USE: Assists the Army study causes of fratricide. These instances are immediately apparent on the chart.

