Paper 055-2008

The Enterprise Melting Pot (Part II)
SAS BI – The Plan versus Reality
Or
Don't Try This at Home, Kids

Gail Davidson, Kforce Government Solutions, Fairfax, Virginia
Guinevere Mills, U.S. Census Bureau, Suitland, Maryland

ABSTRACT

The Decennial Census is one of the largest undertakings of the US government, trying to count everyone at least once. How is this undertaking planned and managed? Most people think the Census amalgamates huge amounts of demographic data; but in addition to the demographic data collected, there are huge amounts of management and operational data collected. This management data is used to prepare and plan for subsequent Censuses as well as day-to-day management of ongoing operations. Our challenge was to integrate and make this data available to the planning and management staffs for the 2010 Census. It was determined that SAS Business Intelligence provided the most robust solution for meeting our users needs. This paper discusses the planned implementation the SAS BI solution: the total solution from hardware and systems planning to training, production roll out and security. Then the paper discusses what actually happened when things didn't go as planned or in our naiveté were overlooked in the planning; the tricks and techniques used to overcome these bumps in the road, deliver the solution, and keep our sponsors and users happy

INTRODUCTION

> *In proving foresight may be vain:*
> *The best laid schemes of mice and men*
> *Go often askew,*
> *And leaves us nothing but grief and pain,*
> *For promised joy!*
> Robert Burns, To a Mouse, 1785

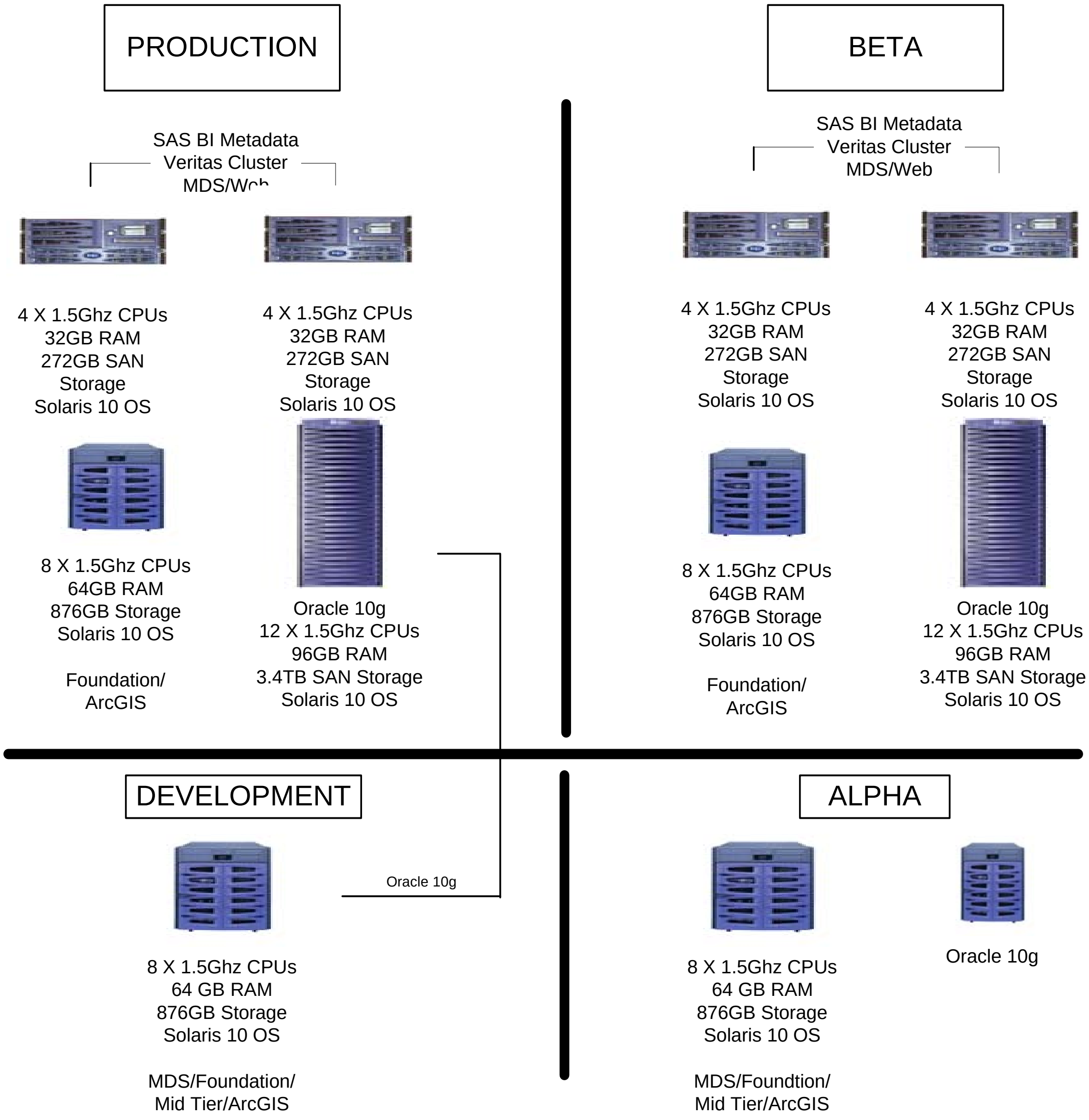
From time immemorial, reality has fallen short of the best-laid plans. In system and software development, how one triumphs (or detours) over these bumps in the road speaks to the adaptability of the chosen software and the creativity of the developers. Things don't always go as planned, but as with any new system being implemented by inexperienced staff the plans aren't always the best laid either. But you have to forge ahead, talk to people more experienced, devour documentation, and do the best you can. Hopefully, this paper will assist in your BI planning and help you avoid some of the situations that tripped us up.

The Decennial Census BI environment supports three separate and discrete applications, multiple data warehouses and data marts. One application is the historic operational and personnel management data from the 2000 Census. The next is a data mart containing demographic information that aids in recruitment and marketing planning for the 2010 Census operations. And the third is an application that captures summary cost and progress information on ongoing operations to allow for agile management and micro targeting of resources. Each application shares a common hardware plan and an overarching umbrella software plan.

CIRCUMSTANCES FELL SHORT

The BI environment supporting these applications initially was intended to be under full configuration management. Promoting development to alpha testing, then to beta test and production and enforcing very strict delineations between the separate environments. But for our configuration management plans to work, we needed the environments built. Building these environments is the first thing that went awry in our plans. The environments were not delivered in a timely manner and we had to improvise and manipulate the metadata to manage development, testing, and production in one environment.

Figure 0: Planned BI Environments



We planned for four environments, development, alpha testing, beta testing, and production (figure 1). Planning to selectively promote objects from one environment to the next. The development and alpha testing environments are our 'lite' environments, whereas the beta testing and production environments were configured for the full-anticipated load.

The 'lite' environments were configured for far fewer users of a completely different type than our production users, namely, the development staff. These users are categorized as high activity users, creating content for the general user community from scratch. Typical usage is generating ETL jobs, OLAP cubes, information maps, reports, and portal content. The production users typically are only consumers of predefined content, although there are some power users that will be creating content. To quantify, there may be 200 users on the system, but maybe only five to ten of those users will be doing ad hoc analysis and creating content. The hardware and environment plan is still the goal, but is not yet load tested.

At the Census Bureau, as in a lot of companies, the staff responsible for the applications is not the same as the staff responsible for building and maintaining the systems. One of the main things our plan depended on was the timely delivery of each environment. This did not happen and for the past year we have been running production processing on the development environment and fitting in testing where we could. This situation, to say the least, has played havoc with our configuration management.

We have been able to limp along by manipulating the metadata depending on what we needed to do when; changing physical data targets in the metadata and querying the metadata when necessary, programmatically, rather than manually changing code. The plan was always to have the same metadata names for target data sets in the different environments only pointing to the different databases associated with each environment. Luckily, switching database targets is very easy to do in the Management Console. We set up a server for each database target. Then within the Data Library Manager plug in we would change the database schema's Oracle server property and also the database server in the library's properties. Every night we would switch the target back to the production database for the overnight production processing, and every morning switch it to development or testing depending on the needs of the day.

Switching target databases between development, testing, and production posed some communication and coordination problems between the development and testing staffs. We are a relatively small team thankfully, but we sometimes had to compromise and prioritize development and testing needs and negotiate the times the schema and server would point to which database thus cutting into our overall productivity.

This switching back and forth to target databases was transparent to the users because we did not give them access to the raw data in the databases. During this time period they only accessed data in OLAP cubes through information maps. We made sure that the libraries were pointing towards the production database when the OLAP cubes were created and refreshed. Thankfully, for the most part, our users are not sophisticated data miners and didn't realize they didn't have access to the raw data. But they didn't feel deprived either with the limited access because they were able to do so much more with the reports and information maps than they were able to do in the past with static reports. We were also able to control their expectations in the user training. We only trained them on what they were allowed to do and didn't mention what they weren't allowed to do.

Another major problem caused by the late delivery of the BI environments, is that we had no place to test installation of hotfixes and upgrades forcing us to adopt a very conservative, rather than proactive, system maintenance program. Every so often we would hit a problem, call tech support and their answer would be - oh, that was fixed in hotfix so and so. Then to apply the hotfixes to our system would be a leap of faith. We had no alternate environments if the hotfix caused a conflict with some other software. In fact, the application of a hotfix did break our system once. This was not a SAS hotfix, but a Solaris patch. We installed the Solaris patch and all servers were up and operational and all appeared well. However, what it did break was the ability of our SAS servers to communicate with the ArcGIS server. None of our geographic maps worked. Our system's staff had to back out that patch.

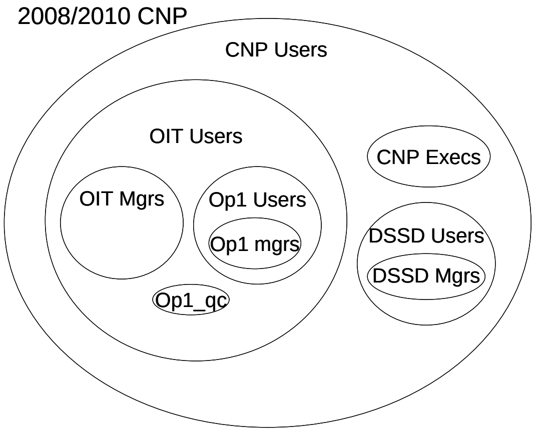
PLANS FELL SHORT

Sometimes reality didn't reach our plans and timetables. But most of our struggles were with the plans themselves. Some aspects of our system and applications were inadequately planned and some aspects were completely were overlooked in planning.

SECURITY ADMINISTRATION

One of the most glaring aspects elided over in our planning was security of the applications content. We assumed that the solution would be obvious from the requirements and easy to implement. In actuality, we struggled with defining logical sets of user groups and group nesting rules that would ease security administration in management console to the BI objects and still provide for the complete separation and security of content within and without the applications. These grouping structures and rules are now part of our requirements process and are addressed at the outset before any development is attempted.

Figure 2: Graphical representation of one operation's group structure in one application



The solution we settled on consisted of completely separate application groups; none of the applications had any groups in common and subgroups were created within the application groups with minimal nesting (and only hierarchical nesting) to protect and separate the application content. Although many of our users are authorized to access more than one application (and more than one subject matter area in an application), the groups are completely discrete and each user would have membership in many groups. Figure 2 illustrates the grouping structure for one application. It is a combination of subject matter groupings and organizational groupings. The OIT Users group, Operation Integration Team, in the diagram is an overarching subject matter grouping. Each OIT actually contains many Op groups (operational); however only one operation is depicted in figure 2. An operation is a further specialization in the subject matter of the OIT Users group. The CNP execs group and DSSD groups are organizational groups. Only the inner groups contain metadata identities as members. The outer groups, OIT Users and CNP Users, are only made of the nested groups and have no individual metadata identities as members.

Once we finally figured out how to organize and segregate our users and application content, we found we were still struggling with implementation. The root of this problem was the inexperience of the staff. Although BI security is well documented now, when we started building the system, security documentation was a little harder to find and pull together. We overcame this by dedicating a resource to scour the documentation, mostly various administration guides, to consolidate the information relative to our environments and develop a security plan/guide that is wholly relevant to administering our environments. The security plan is part plans and rules and part instruction manual tailored to our particular environments. It contains guidelines on group structures and memberships and what permissions are required for different object types. With the security plan/guide and more experience as time goes on, we are managing the security pretty well and having to call tech support a lot less (on security issues anyway). It did take a dedicated and concerted training effort to get a handle on it. Table 1 show an example of the security rules in the security plan that all the developers must to follow.

Table 1: Security Rules

1) Only metadata associated with data surfaced directly to the SAS EBI mid-tier clients will be registered in the Foundation repository.
2) The metadata identity name shall be the JBID.
    a) The metadata identity job title will consist of a string in which the first part specifies the application the user is accessing, follow by a shorthand of which main content resources they are authorized, followed by their organizational division.
    b) The metadata identity description consists of their last name, first name, mi.
3) The group name will consist of a string composed of the Application_GeneralContentDescription_Role, for example, cnp_ccmil_mgrs.
    a) Only three roles are generated by default for general content. These are users, mgrs, and qc.
4) Group membership should be nested wherever it makes sense to, but not to the point where artificial relationships are created.
5) No application groups will intermingle. Each application will remain completely discrete from the other applications.
6) Unless there is an overriding or regulatory reason for not sharing information, data, and content, it will remain open (RM) to the widest audience possible.
7) All content will be protected by group authorizations; no individual metadata data identity authorization will be set on BIP tree content.
8) Only the DBIA group will retain authorization to create libraries in the Foundation Repository.
9) The persons with responsibility for administering the content for an operation will be configured as its group content administrator prior to construction of its pages, portlets, and content.

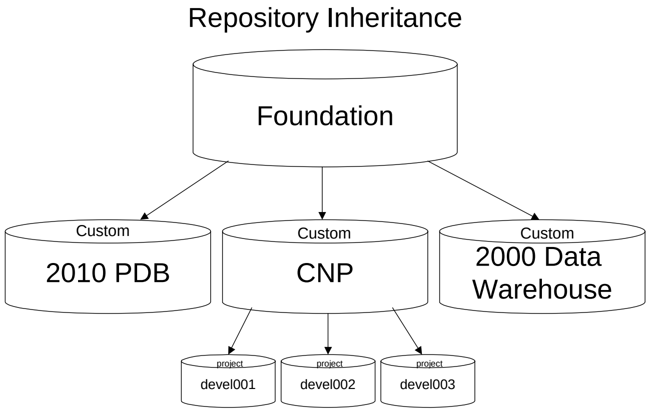
REPOSITORY AND APPLICATION SEPARATION

The DBIA System currently supports three separate applications, the Cost & Progress System (CNP), The 2000 Data Warehouse, and the 2010 Planning Database. Because of the way the SAS BI mid-tier applications (Portal, Web Report Studio) operate, some of the metadata of each of the applications must reside and share space in the Foundation repository. Any information that is surfaced to the SAS BI mid-tier web applications must be registered in the Foundation repository. However, separating as much metadata between the different applications as possible in custom repositories is the easiest way to segregate the application metadata. Default Access Control Templates are easily applied at the repository level.

An understanding of metadata repository structure and metadata inheritance is required at this point to understand where, why, and at what point the different DBIA applications metadata is stored in which metadata repository. Figure 3 illustrates the conceptual metadata repository scheme in the DBIA environments. At its base is the Foundation repository. Each application is configured with a custom repository, which inherits all metadata from the

Foundation. For example, the 2010 PDB metadata consists of the metadata in the Foundation repository and the 2010 PDB custom repository. However, it is completely blind to the metadata in the custom CNP and the custom 2000 Data Warehouse repositories.

Figure 3: DBIA Metadata Repository Structure



The impact of this feature of metadata separation and surfacing content to the SAS BI mid-tier client applications is that only information that is required to be accessed through the Portal or Web Report Studio need be registered in the Foundation repository. However, this would include the chain of information accessed: information map, cube, library, and data set. Any other metadata required by that application could be maintained in the custom repository.

NAMING STANDARDS

Another issue we struggled with early on was naming standards. This was part of the planning process but a late aspect of our planning. The problem became apparent during our prototyping phase when everyone devised there own naming standards, which was discernible to them but not to others. We called this our just in time naming standards. Our goal was to be able to identify each object in the metadata by its name outside of the folder structure. This was a big issue in the Foundation repository. It shares content with the three applications we support. All content surfaced to the mid-tier web applications and underlying data is stored in the Foundation repository. By various naming standards and metadata standards we can tell what each object is describing; whether it's a library, report, information, cube, metadata user identity, what application its for, and a general idea of what it contains.

All names are unique in our metadata and that sometimes leads to relatively long names. The problem with non-unique names came to light during our prototyping. The work we do is somewhat cyclic and in the past it was natural to call an apple and apple, and placing them in folders that identified what cycle they were from. Calling datasets the same name was our habit, only distinguishing them by their folder location. Some of the tools and wizards (most notably OLAP cube wizard), don't surface the folder structures, and having duplicate names was a real problem when telling the wizard which data to use because we'd have five or six datasets with the same name.

So far, long hierarchal metadata names are working for us. The only exception is that we discovered a small bug in the meta libname engine due to the long names for our Oracle libraries. There is a simple enough workaround for the bug, although rather inconvenient. The workaround consists of creating a new library with a short name, eight characters or less (libname rules apply) with exactly the same parameters as the long name; importing the datasets again; and then deleting the short named library. Something about the process of creating object IDs for the library and datasets again using the short library names overwrites the object IDs on the long name. I'm not exactly sure why that works, but it does.

Coming up with standards that everyone could live with required a high degree of collaboration between the staff and many meetings on the subject. In the end it was worth the effort. The staff now works from the same page, and it's easier to maintain each other's work.

APPLICATION DEVELOPMENT

The staff was divided into three groups: Requirements; Data Modelers which focused on getting the data into the enterprises data warehouse and developing the ETL jobs; and Front-End which focused on information delivery by developing the OLAP cubes, Web Report Studio reports, information maps and portal contents. The plan was for the three groups to execute their processes in a logical and linear fashion. The Requirements group would first collect the user requirements. Next the Data Modelers would develop the ETL jobs and load the data warehouse. And lastly, the Front-End developers would create the objects for the users. In reality, what is happening is the requirements are usually late causing the rest of the project development schedule to crash meaning the Front-End and Data Modeler development occurs at the same time. This requires a lot of collaboration between the two teams sometime resulting in miscommunication from what the data modeling team delivers and what the front-end development team is expecting resulting in one team or the other doing rework.

Additionally, a new global requirement presented itself from the data and how it was stored in the enterprise data warehouse. Because we were combining data from so many different sources, the date associated with each source no longer met our needs in defining the currentness of the data in the reports. For example, if we gathered progress data from two days ago and then gathered cost data from yesterday the dates can't be merged to form one picture of what the data is today. Today is the vintage date. Associated with the vintage date are the two-day-old progress data and the one-day-old cost data. This was very difficult to design and program. What we ended up doing was developing a macro program that generated custom Oracle SQL code and used the pass-through facility for Oracle to process it.

Among the many problems in our planning and development we've only come across a few things in the software itself that put a crimp in our plans. First, with SAS BI web application there is a limit in filtering on OLAP cubes. Only one variable at a time can be used in a filter. Because Census Bureau deals with huge amounts of demographic and operational data this has become a problem for our users with ad hoc reporting. SAS Institute is aware of this limitation and we've been advised this has been fixed in version 9.3. The next big issue is not being able to selectively promote portal pages and schedule flows. This hasn't been a huge problem yet because we've only been running in one environment, but as we mature and our other environments come on-line this problem will take on larger proportions. The last issue deals with our inexperience in working with the software. Our users have expressed a need to suppress statistics at certain levels in OLAP cubes. It's technically possible; however, no one on our staff knows how to program the MDX scripts yet.

TRAINING

The users are very pleased with the abilities they now to create their own reports, graphs, and maps. The SAS Add-In for Microsoft Office enables them to harness the power of SAS analytics and to access SAS data sources from within applications they use on a daily basis and are more familiar with. User training results exceeded our expectations. We developed a training manual based on SAS Institute training but tailored to Census DBIA applications. They have been very receptive to the ease of use of the SAS BI web applications. We've also been able to use the training to manage their expectations as well. We've wet the users' appetites with a taste of the new power at their fingertips and they want more. All the training courses have received good reviews with request for additional classes nationwide.

CONCLUSION

We gave our best shot with the planning but reality had its own ideas. We scrambled when we needed to. We dedicated resources to determining our naming standards and security plans. The application security is the marriage of naming standard enforcement and security implementation. We discovered creative workarounds when we needed to. We discovered a creative way to administer multiple environments when we only had one physical environment. We developed workarounds for the limitations with the SAS BI software. We incorporated our lessoned learned into the security plan rules. The rules listed in this paper, directly address problems in our original planning. And finally, the one aspect of our plan actually worked out as planned was the training. It was delivered on time and on budget and was particularly well received by our users. And the fact that the training was the final step in delivering the product to the users show that even though we had our problems with planning the users were happy with the end results.

ACKNOWLEDGMENTS

This project could not have undertaken without the tireless efforts of the project team: Loretta Williams, Bert Hotton, Bob Anderson, Annette Davis, Zbig Gorski, Hank Okoro, Jesus Lopez, Linda Flores-Baez, Mariamma Cherian, David Kiasi-Barnes, Van Nguyen. A special thanks to SAS Technical Support who have been more than patient with us over the course of this effort. Thank-you all for your support and assistance.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Gail Davidson
Kforce Government Solutions
301-763-3444
gdavidson@kforcegov.com
gail.s.davidson@census.gov

Guinevere Mills
U.S. Census Bureau
301-763-5421
Guinevere.z.mills@census.gov