RELATIONAL DATABASES FOR SSC DESIGN AND CONTROL

E. Barr, S. Peggs, and C. Saltmarsh

SSC Central Design Group*
c/o Lawrence Berkeley Laboratory
One Cyclotron Road
Berkeley, California 94720

Introduction

Most people agree that a database is A Good Thing, but there is much confusion in the jargon used, and in what jobs a database management system and its peripheral software can and cannot do. During the life cycle of an enormous project like the SSC, from conceptual and theoretical design, through research and development, to construction, commissioning and operation, an enormous amount of data will be generated. Some of these data, originating in the early parts of the project, will be needed during commissioning or operation, many years in the future. Two of these pressing data management needs—from the magnet research and industrialization programs and the lattice design—have prompted work on understanding and adapting commercial database practices for scientific projects.

Modern relational database management systems (rDBMS's) cope naturally with a large proportion of the requirements of data structures, like the SSC database structure built for the superconducting cable supplies, uses, and properties. This application is similar to the commercial applications for which these database systems were developed. The SSC application has further requirements not immediately satisfied by the commercial systems. These derive from the diversity of the data structures to be managed, the changing emphases and uses during the project lifetime, and the large amount of scientific data processing to be expected.

To be able to respond to the growth and change of data management, it seems impossible to construct an overall data design at this stage of the project. Rather, a "hub and spoke" arrangement is envisioned. A hub database will manage the interaction between several spoke databases, each of which can service the specific needs of its primary users while maintaining well defined public interfaces for the wider audience. In this way it is hoped that data from the lattice designers and from magnet testing and mapping groups, being generated now, will be accessible to the commissioning and operation teams when it is needed, and into the 21st century.

To respond to data processing needs, a number of techniques, utilities and libraries are being developed. They will allow efficient and well-managed transfer of data between the database and scientific data collection, display, and analysis programs. They use a commercial rDBMS, Sybase, running on a Sun 2-280 dataserver, and a data discipline, SDS, (for Self Describing Standard [1,2]) which attaches format and context information to datasets wherever they are.

The two applications considered in this paper are:

1. Management of and access to raw and analyzed test data from the magnet R&D laboratories, and,
2. Management and integrity control of the lattice design databases.

Magnet Test Databases

Work on management of the data from magnet quench runs has concentrated so far on the development of tools and techniques for a clean interface, between the data acquisition and analysis programs and the database itself. It is proposed that the SDS data discipline be used as early as possible in the life cycle of data generated from the SSC Magnet Test Laboratory to be built in Texas. Current data from the Fermilab Magnet Test Facility are translated from their somewhat specialized binary format to the more general SDS on arrival at the LBL Vax cluster. Once that is done, process to process data communication, including moving data into and out of the database and between processors, can be accomplished by general purpose tools and interface libraries.

The Sybase database management system runs on the SSC-CDG magnet division SUN 4-280 server. This product uses a client-server model to allow distributed databases and distributed access. The SQL servers accept the industry standard Standard Query Language and provide some crucial data integrity control, including the ability to trigger external processes on the reception or modification of data. Separated servers (at present servers are being brought on line at FNAL and at the Brookhaven Magnet Test Facility) are able to cooperate to provide data management on distributed databases. Client software on the SUN workstations and on Vaxes can be accessed through VT100-like terminal access. The delivered C language interface has been layered.

Figure 1 is a block diagram showing the process connections followed by data from current FNAL magnet quench runs. Data put into the SDS discipline from the data acquisition system are semi-treated: a listing of the data structure is shown to the left of Figure 2. These data contain run and instrumentation context information (description strings, device parameters and units and so on) as well as the data contained in 87 1024-element integer arrays (Log_I2.data) and 23 1024-element floating point arrays (Log_R4.data). This complete dataset is given to the database for storage as a large binary object; as such, few of the powerful query capabilities of SQL are available. Even so, the database can use the standard tagging information of SDS—timestamps and data object naming—to provide the first level of data management. SDS also provides the graphic and ASCII dump utilities shown to the right of Figure 2.

First pass analysis of the raw data generates more complete tables of summary information for data control. The program interface on the top right of Figure 3 is a simple browsing tool built using the "forms" library of Sybase. Here the quench summary table is displayed and an SQL query generated. The results of the query are used to generate another SDS, listed on top left, and, if required, C structure and Fortran common block declarations for use by specific application programs. In Figure 3 the resulting SDS has also been sent to the standard interactive graphics tool of SDS. At bottom right is the screen of the general purpose database access program delivered with Sybase, the "data workbench."

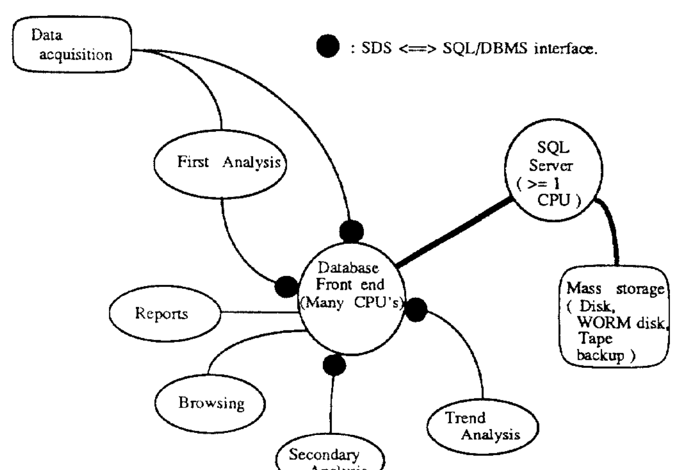

Figure 1. Block diagram showing the process connections followed by data from current FNAL magnet quench runs.

*Operated by the Universities Research Association, Inc. for the U.S. Department of Energy.

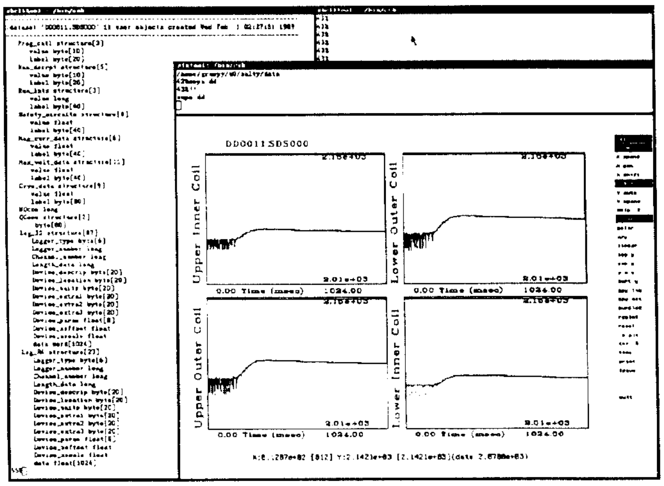

Figure 2. Raw data structuring.

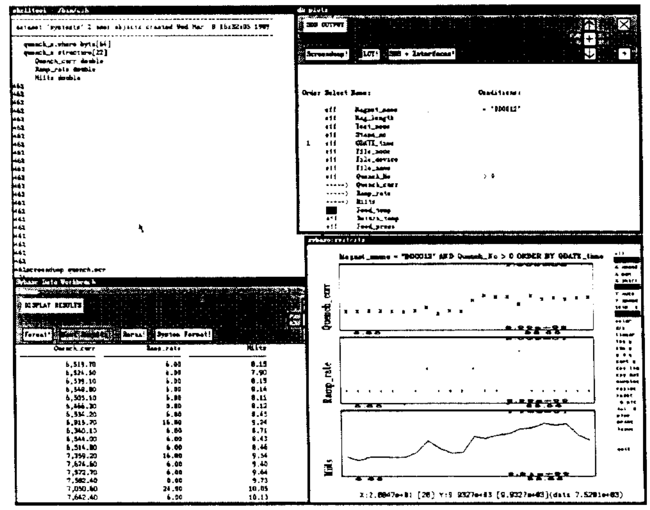

Figure 3. Database access.

Work is continuing to put cryogenic logging information and field mapping data under the control of the database. By starting with firm and early control of the data, reusable utilities can be created (none of the programs described above are specific to magnet data) and relieve the programmer of as much mundane bookkeeping work as possible.

Lattice Databases—Optics and Footprint

Several database structures have already been constructed, or are nearing completion, under the general topic of the SSC lattice. Strictly speaking, the lattice is the sequence of magnets in the SSC ring (more exactly, in the two SSC rings), although the context is very different for different people. One accelerator physicist, for example, may be interested in optics design, that is, in optimizing the linear optical behavior of the ring, subject to the constraints of realistic (model) magnets. Another may be interested in tracking particles around the ring, and observing the effects of nonlinearities, both intentional and accidental. On the other hand, a surveyor does not care what a quadrupole magnet is, but needs to know exactly where the footprint of the ring lies in the Texas countryside, so that the correct parcels of land can be bought.

Each of these three rDBMS users needs databases which contain different information organized in different ways. A first round of design has been made to satisfy a broad spectrum of users like this, breaking down lattice information into three sets of tables, or abstract database structures. Each of these structures may be present in several concrete Sybase databases. For example, the performance of lattice variants, or the merits of different footprints, might need to be compared. The three structures discussed in detail below are called *hierarchical optics*, *flat optics*, and *footprint survey*. Their relationship to each other is sketched in Figure 4, including typical programs which transform the data from one structure into another, and make the data available in the application computing environment. An *optical properties* structure, not yet constructed, is shown with dashed lines, to emphasize the point that new demands can be met in a flexible, modular, manner.

The optics designer naturally has a hierarchical view of the lattice, since she deals with idealized objects at both high and low levels which may be repeated very many times. For example, if the current in an arc cell quadrupole is adjusted to give a certain phase advance, all arc quadrupoles should acquire that current. The *hierarchical optics* structure reflects this view in six tables. At the highest level, the *beam line* table describes large structures such as clusters, arcs, sectors, and cells, with the beam line name as primary key. A beam line is constructed of other beam lines, and ideal magnets. The *ideal magnet* table describes the set of complete physical magnet models, with field free ends, et cetera, exactly as they would be installed in the SSC.

An ideal magnet is constructed from elements found in the *magnet piece* table. Thus, an arc quadrupole includes field free ends, a marker at its center, two halves with fields, and a beam position monitor. Objects in the *magnet piece* table have lengths and strengths which are specified, perhaps recursively, in the *geometry* and *strength* tables. These tables distinguish parameters which are varied by changing the current in a bus from parameters which are set when the magnets are installed. The sixth table, *name location*, listing the name of the table in which an object is found, is built from the other five tables by a simple SQL command.

Connection to a hierarchical database is made via the C program DBSF, which provides the user with a variety of options. Many lattice design and tracking programs in the regular computing environment, such as MAD, DIMAD, TEAPOT and TRANSPORT, accept compatible ASCII files written in the "standard lattice format" [3]. Output from DBSF may be requested in this format, and may be terse or verbose. Flat format output may also be requested, in either ASCII or SDS forms, ready for uploading into a database constructed with the *flat optics* structure. This consists of five tables, as illustrated in Figure 5.

It is a computer science cliche that data structure design is a major part of application software development. The *flat optics* structure is described in some detail in the figure and below, at the risk of boring the reader, to illustrate that this cliche contains a significant amount of truth. Both the *flat optics* structure and the standard lattice format lead to compatible input into many programs. However, the former is defined at the low level of database tables that are very close to FORTRAN common blocks or C structures, while the latter is defined at the level of an ASCII text file. Definition at the lower level leads to the possibility of greater modularity, and a larger number of smaller application programs.

The accelerator physicist interested in tracking does not want to know the hierarchical structure of a lattice, but considers it as a sequential flat list of objects traversed one by one. In the *flat optics* structure these objects are ordered by their azimuthal location, *sindex*, in the *sequence* table. The magnet at a given location is identified by a magnet model pointer, *pmagnet*, and a magnet instance pointer, *pinstance*. The *instance* table contains the multipole errors attached to a particular magnet, in the commonly used (b_n, a_n) notation. A particular magnet has a generic model type, which is described in the *magnet* table in terms of attributes like length, strength and tilt, and by pointers into *parameter* and *multipole* tables. The *multipole* table contains statistical information about the multipole field errors averaged over a particular set of magnets.

It is not possible to fill the *multipole* and *instance* tables directly from a hierarchical lattice database, because the information is not available there. When magnets are constructed and installed in the SSC tunnel, information from their magnetic measurements will be recorded in these

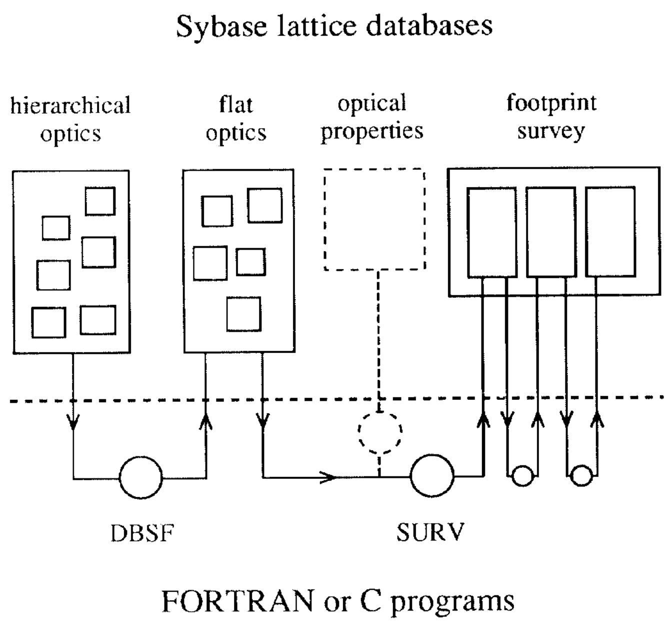

Figure 4. The present set of SSC lattice database structures, above the dashed line, and typical data transformation programs in the regular computing environment, below the line.

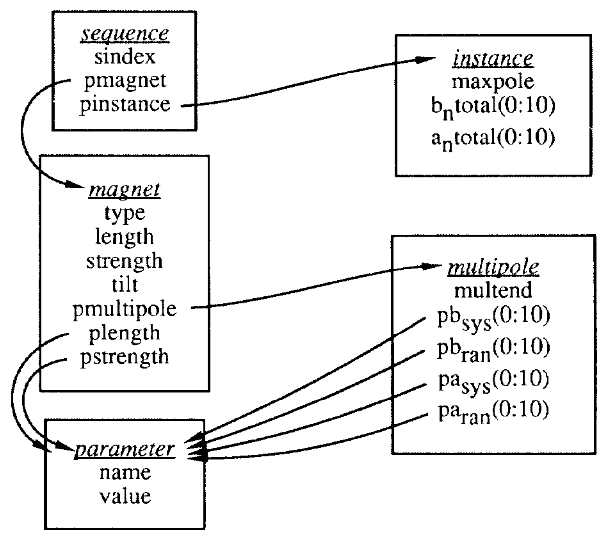

Figure 5. The five tables in the *flat optics* database structure, describing the sequence of magnets in a lattice.

tables, but until then it is only possible to enter statistical values into *multipole* "by hand", and then to fill *instance* with the aid of a random number generator. ASCII or SDS representations of flat databases are read by programs like SURV, which generates lattice coordinates ready for database uploading, and XMAP, a differential algebra based polynomial map maker [4].

The location of a lattice object needs to be known in (at least) three coordinate systems, *lattice* coordinates, *state plane* coordinates, and *site* coordinates, which correspond to three tables in the *footprint survey* structure. The *lattice* table records where magnets are relative to each other in three dimensions, and records the direction of a tangent to the ring at that point. Its primary key is *sindex*, the same as in the *flat optics* structure. The *state plane* table fixes the center and orientation of the SSC at an absolute location in Texas, and records the location of the ring in the same coordinates that are used to define the location of parcels of land. At some point in the future construction on the SSC site will be begun, using an accurate mesh of survey monuments which will define site coordinates. Only then can the *site* table be filled with information.

Acknowledgements

We gratefully acknowledge the contribution of Vern Paxson in developing the DBSF program.

References

1. Saltmarsh, SDS usage documentation, SSC report in preparation, Berkeley.
2. Paxson et al, these proceedings.
3. Carey and Iselin, Proc. of the SSC Summer Study, p. 389, Snowmass, 1984
4. Irwin and Peggs, these proceedings.