ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S16MS01

SSC Lattice Database and Graphical Interface

C. G. Trahern and J. Zhou
Superconducting Super Collider Laboratory
2550 Beckleymeade Ave., Dallas, Texas 75237*

Abstract

The SSC lattice database and the graphical tools used to access it are discussed.

I. INTRODUCTION

When completed the Superconducting Super Collider will be the world's largest accelerator complex. In order to build this system on schedule, the use of database technologies will be essential. In this paper we discuss one of the database efforts underway at the SSC, the lattice database. The original work on this database system began at the SSC Central Design Group and is described in reference [1].

The SSC lattice database provides a centralized source for the design of each major component of the accelerator complex. This includes the two collider rings (top and bottom), the High Energy Booster (HEB), Medium Energy Booster (MEB), Low Energy Booster (LEB) and the LINAC as well as transfer and test beam lines.

These designs have been created using a menagerie of programs such as SYNCH, DIMAD, MAD, TRANSPORT, MAGIC, TRACE3D and TEAPOT. However, once a design has been completed, it is entered into a uniform database schema in the database system.

In section II we further discuss the reasons for creating the lattice database and its implementation via the commercial database system SYBASE[2].

Each lattice in the lattice database is composed of a set of tables whose data structure can describe any of the SSC accelerator lattices. This data structure will be discussed in section III.

In order to allow the user community access to the databases, a programmatic interface known as dbsf (for database to several formats) has been written. This interface is the subject of section IV. dbsf creates ascii input files appropriate to the above mentioned accelerator design programs. In addition it has a binary dataset output using the SDS (Self Describing Standard) data discipline provided with the ISTK (Integrated Scientific Tool Kit)[3] software tools.

In section V we discuss the graphical interfaces to the lattice database. The primary interface, known as OZ, is a simulation environment as well as a database browser.

OZ has been created using techniques of object oriented modelling and coded in C++ using the ISTK software

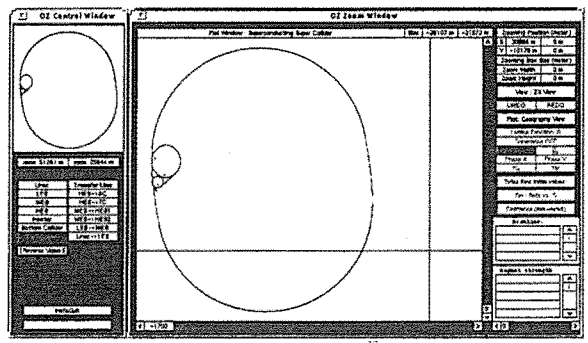

Figure 1: Control and Display Windows of OZ

tools. OZ is SSC specific in that it presents a geometrical view of the collider complex from which one may select the lattice of interest. This geometrical data is also the source of the data used to site the complex geographically. OZ is an interactive simulation environment in which the user can change various parameters such as a steering magnet's field strength and then see whether a beam of given emittance will survive its propagation within a fixed spatial aperture.

In addition to the geometrical view of the complex, one also needs a more abstract view of a lattice structure, and this is provided by a program known as LATVIEW. Because the beam line hierarchy implicit in an accelerator design can be highly nonintuitive, LATVIEW gives an interactive graphical view of this hierarchy.

II. PHILOSOPHY

We have implemented the lattice database using a relational database management system (RDBMS). The particular software system currently used is SYBASE operating within a UNIX workstation computing environment. By putting the lattice information within a RDBMS tied to a network, essentially universal access to the data can be supported. In addition by maintaining a uniform description of the lattice information, various groups such as mechanical and civil engineering, survey and alignment as well as diagnostics and simulation can be coupled to the same data in an efficient manner thus reducing the probability that different groups will use incompatible design information.

In addition, because accelerator design is usually done with a variety of design codes such as the ones mentioned earlier, a lattice database of some kind is the only way to

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S16MS01

effectively manage the design process. The fact that our choice has been to use an RDBMS reflects the desire to learn new software technologies as well as to avoid the paper trail of past accelerator construction experience. The lattice database effort should be regarded as one of the first experiments at the SSC in using new technical organizational schemes.

SYBASE is one of the major SQL (structured query language) RDBM systems available. It runs on a variety of hardware platforms, and has an excellent network interface. Lattice database applications have been written using C and the Open Client/C software library supplied by SYBASE. The performance of these applications has been exceptional. In the future the applications should be written with embedded SQL so that other users at different sites should be able to use the lattice database system with another RDBMS.

III. LATTICE DATABASE

Lattice Structure as Database Tables

In creating the lattice data structure, we have adopted the MAD 8.1 element definitions[4] since MAD provides a *de facto* standard for accelerator design descriptions. Each lattice is assigned its own database in the RDBMS with the same set of generic tables in each database.

A generic lattice database is made of five primary tables: beam_line, slot, magnet_piece, geometry, strength, and 12 secondary tables: quadrupole, sextupole, octupole, multipole, rfcavity, bend, drift, collimator, closed_orbit_corrector, monitor, elseparator and solenoid. The primary tables define a basic lattice hierarchy of beam lines, components and parameters and the secondary tables provide additional information which completes the MAD 8.1 element definitions.

The beam_line and slot tables define the beam line hierarchy. Drifts, magnets or other elements are entered into 'slots' as sequences of character strings by their names and slots are strung together in other character strings to make beam lines. Beam lines can also be concatenated into larger beam line structures and so on.

The magnet_piece table contains columns which describe the basic physical properties of each element such as its type, length, strength, etc. The geometry and strength tables store physical parameter names and values such as lengths and magnetic properties such as field values or field gradients. Two tables were created for parameters instead of one because there will certainly be different versions of the strength table associated with different operating conditions of an accelerator such as injection, collision and points in between. The application program can be directed to choose the strength table of choice.

In our experience to date the MAD element definitions have been appropriate for all accelerator lattices excepting the linac. In order to provide a faithful database representation of the linac design, we have had to implement an additional set of tables which define new lattice elements needed by TRACE3D, a linac design program in general use at the SSC.

In principle it is now straightforward to adapt to new demands on the data structure by creating new tables when necessary. For example, in order to describe the layout of the assembly of magnets to determine the existence of physical interferences, information concerning the outside dimensions of magnet elements is needed in the database. This information is maintained in a table known as magnet_size which contains columns detailing the shape and external dimensions of a given element. The primary key of this table (and most others) is the element's name. Breaking this information off into another table rather than adding a new column to the magnet_piece table for example, helps to separate applications which are independent of each other. For example a beam line survey program which models physical interferences may require information about the outside dimensions of beam line components, but does not need the magnet strength values appropriate for linear optics calculations.

SQL example

An example of SQL code which creates the magnet_piece table:

```
create table magnet_piece
(name              char(20),
type               varchar(20),
tilt               varchar(10)     null,
length_defn        varchar(60)     null,
strength_defn      varchar(60)     null,
engineering_type   varchar(20)     null,
comment            varchar(130)    null)
```

The first column above specifies the table's column names. They are name, type, tilt, length_defn, strength_defn, engineering_type and comment. Name is the ascii name of the beam line component. Type refers to one of the MAD element types such as drift, quadrupole, sextupole, etc. Tilt specifies the orientation of the magnetic element around the beam line. The meaning of strength_defn can depend on the type. For example, the quadrupole strength is proportional to the gradient of the magnetic field, and the sextupole strength is proportional to the second partial derivative of the magnetic field strength. Symbolic algebraic expressions for these quantities can be entered in this column as long as the parameters used are defined elsewhere in the geometry or strength tables. Engineering_type is an alternative name which the user may introduce to flag components for special purposes in the application codes.

The second column above specifies the data types. For this table these data types are 'char' and 'varchar'. These character field types differ from each other in that char uses all of its allocated space and varchar only stores that portion of the allotted space of the field which is actually used. Finally, the null qualifier on some columns means that no data is required in that column; both name and type being

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S16MS01

non-null are required to be entered by the database user. The other database tables have similar structure, and all tables have a unique index defined on the primary key, usually the first column. These indices help maintain the integrity of a table by prohibiting multiple entries with the same primary key.

IV. USER INTERFACE to DATABASE

dbsf (database to several formats) is the primary interface to the lattice database. It is written in C and uses the Open Client/C libraries provided by SYBASE. To run dbsf one supplies any lattice keyword such as a beam line, component or parameter name. dbsf first performs a query to determine the location by table of this keyword, and then searches that table for information about the key. If other keys are associated with the primary key, these are also made the subject of similar database queries until no further unknown keys are encountered. At this point an extended symbol tree has been constructed within dbsf and depending on one of many options selected at the UNIX command level, dbsf will format an input file appropriate to that option. The first options available were for MAD. Since that time, all the accelerator codes listed in the introduction are supported.

Recently, an option to create a binary dataset using the SDS data discipline has been installed. SDS is provided as part of the ISTK tool set. The SDS dataset binds a description of the data structure used to create the data within the dataset itself as a header object. Consequently, within the context of database applications, SDS data can maintain the integrity of the database structure outside of the RDBMS. Application programs that read generic SDS data can then be used with this data. In particular, SDS data can be passed from one RDBMS to another through ISTK supplied interfaces. Such interfaces can provide a simultaneous solution to the problems of making heterogeneous database system communicate effectively and also allow large data to flow quickly from one point on a network to another. Several programs which are specific to lattice issues will be discussed below.

V. GRAPHICAL INTERFACES

OZ

Early in the development of the lattice database, the design of a graphical interactive view of the complex of accelerators was begun. The name of this program is OZ. OZ was required to display a geometrical view of the entire accelerator complex and to have the ability to use a 'mouse' to select a part of the accelerator system and see that area visually expanded in greater detail while displaying the relevant properties of the selected beam line components. In addition the interface was required to provide a basic simulation environment for each of the accelerator lattices. Consequently, it should not only display the design values of linear optical properties (betatron functions, dispersion) correlated with selected regions of a lattice, but also allow the user to modify the magnetic strengths and recompute the optics.

Finally, the interface should include a particle tracking module. The tracking module would use a fixed initial emittance profile to define positions and momenta of a small number of particles, and then propagate these particles and display the resulting emittance profile. This is done after one turn for circular lattices and at the end for transfer lines. The stated goal of the particle tracking module was to provide the user with a convenient way to modify steering elements strengths and see the result graphically. In this way one could study the effects of beam apertures quickly. The tracking module was not intended to model the various errors in magnetic field strength or alignment. It provides an ideal view of the particles' motion.

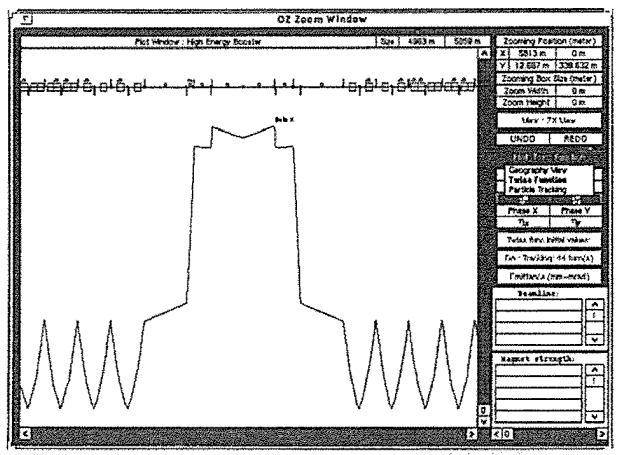

Figure 2: Betatron Function Window in OZ

As the general computing environment of the SSC Accelerator Division is based around UNIX workstations running the X11 windowing system, the graphical tools used to build OZ were chosen to be compatible with this environment. The original development was begun using the InterViews 2.6 toolkit[5], but was later changed to the graphics libraries provided with ISTK. These graphics tools, known as glistks, are based on a subset of InterViews' Interactor class and are tailored to build scientific graphical interfaces.

The interactive capability of OZ is manifold. After selecting one of the lattices from the command window, OZ displays the geometrical view of that lattice in the display window. One can select a portion of that lattice for expansion by grabbing the region with the mouse. One menu then allows you to view that lattice from different two dimensional perspectives, or to query the database to find a particular beam line name. Another menu allows the user to view the linear optical properties. A one dimensional version of the lattice is displayed at the top of this view so that the user can tie the optics to element locations.

Finally, one can select the tracking menu in order to define initial parameters for particle tracking such as

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S16MS01

coordinates and momenta or change the desired emittance profile. OZ will then track the particles for a few turns and display the final profile in a series of windows showing horizontal and vertical phase space as well as the purely spatial transverse profile of the beam.

LATVIEW

Another graphical interface to the lattice database is called LATVIEW. This program was designed to allow visualization of the lattice hierarchy. A lattice such as the main collider ring of the SSC is composed of more than 20,000 elements including drift spaces, magnets, beam position monitors and position markers. The collection of beam lines which combine these elements into a lattice design is sufficiently complicated that a graphical interface to its structure is absolutely necessary for most users.

LATVIEW uses the SDS output from dbsf as its source of information for display. This SDS dataset includes as one of its elements the enumeration of the 'level' at which a beam line exists in the hierarchy. The notion of level is defined so that level 0 corresponds to a completely flat lattice, i.e., with all trace of the beam line hierarchy eliminated. Levels 1 and 2 show the relationship of the basic elements to the slots in which they belong, and levels 3 and so on define the tree of nested beam lines whose highest level corresponds to the complete lattice's name, the original keyword given to dbsf needed to generate the dataset.

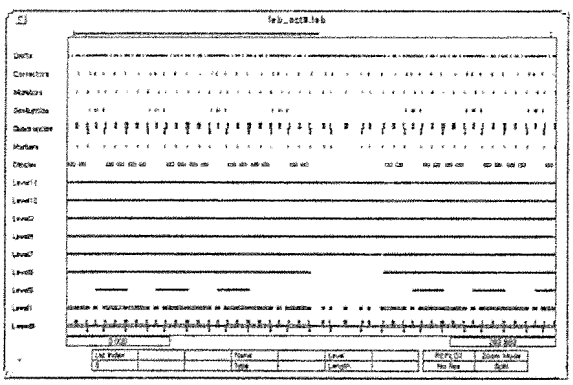

Figure 3: Lattice Hierarchy of LEB from LATVIEW

LATVIEW displays information on each element (beam line or individual magnet) by placing the mouse on the element of interest. This action provides sufficient information to point back within the SDS dataset and pick out the relevant information pertaining to that element. The information is displayed at the bottom of the LATVIEW window. One can also select a part of the lattice with the mouse, and LATVIEW will show the expanded view. In addition to its display of the beam line hierarchy, LATVIEW will also display groups of magnets by type. So, for example, if one is interested in locating all quadrupole locations along the beam line, LATVIEW provides a convenient way to do this visually without losing the relationships of these magnets to their parent beam lines.

VI. CONCLUSIONS

The SSC lattice database has been in existence for two years. It has been an essential part of ongoing operational simulation investigations as well as being the basis for engineering drawings of each accelerator system and the geographical footprint of the entire complex of accelerators. As the SSC moves into the construction phase, additional database structures will be needed to describe the hierarchy of cryogenic, electrical and control systems that overlay the basic lattice designs. In addition the specification of an automated storage/retrieval system for the names and locations of the multifarious pieces of equipment that are attached or associated in some way with the lattice complex is essential. Some of these problems are under investigation, and present efforts are directed at the determination of requirements for these global database systems.

References

[1] E. Barr, S. Peggs, and C. Saltmarsh, SSC-N-606, March 1989.

[2] SYBASE, Inc., 6475 Christie Ave., Emeryville, California, 94608.

[3] C. Saltmarsh, M. Allen and S. Acharya, ISTK, unpublished internal documentation at SSCL, July 1991.

[4] D.C. Carey and F. Christopher Iselin, Proc. of the SSC Summer Study, p. 389, Snowmass, 1984 and The MAD Program, F. Christopher Iselin, CERN/SL/90-13 (AP).

[5] Mark A. Linton, InterViews Reference Manual, v. 2.6, February 1990, Stanford University, Stanford, California.