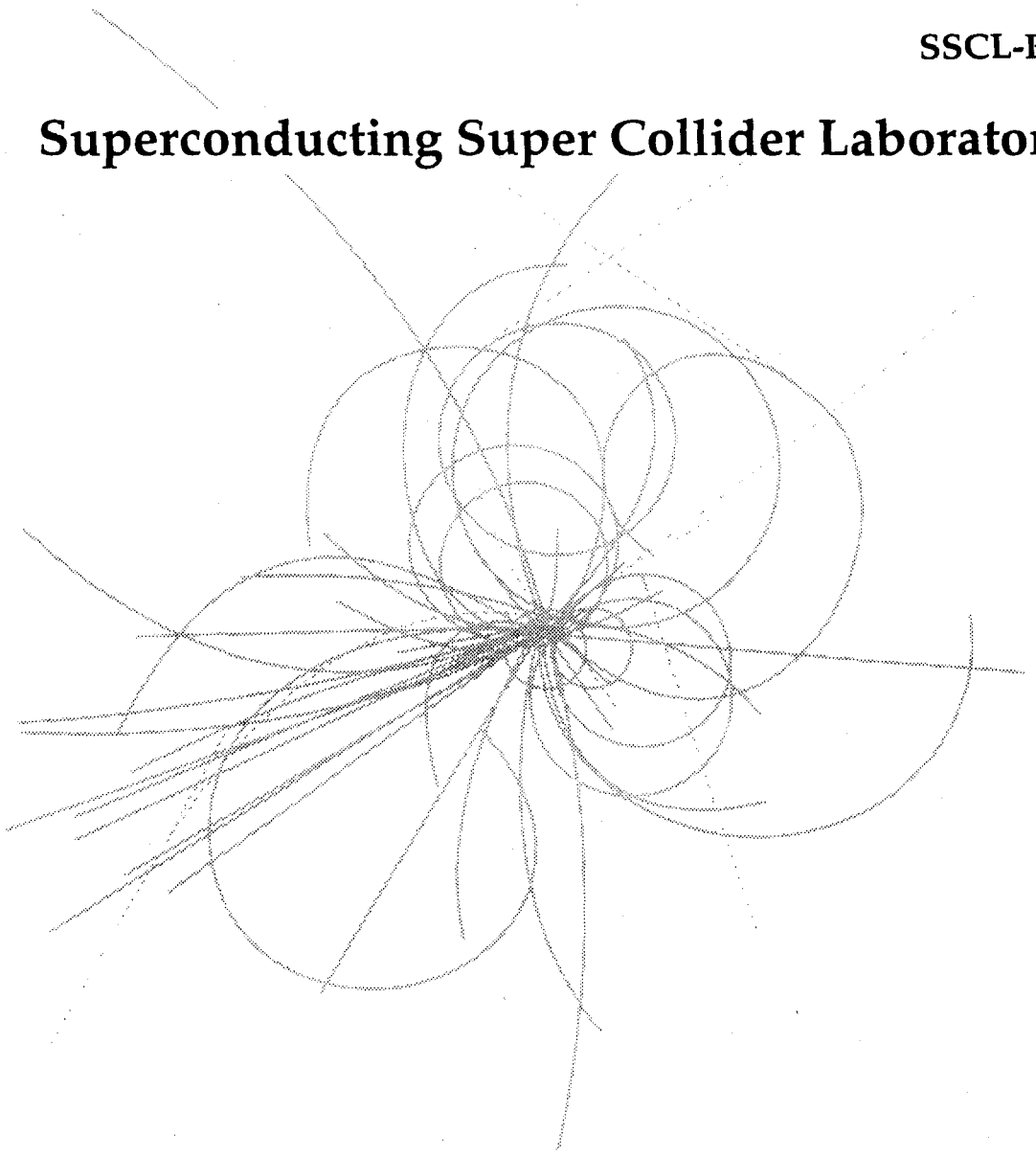
SSCL-Preprint-10

SSCL-Preprint-10

Superconducting Super Collider Laboratory



Object-Oriented Analysis and Database Design for the SSC Magnets

S. Robinson

June 1992

SSCL-Preprint-10

Object-Oriented Analysis and Database Design for the SSC Magnets*

S. Robinson

Superconducting Super Collider Laboratory†
2550 Beckleymeade Ave.
Dallas, TX 75237

June 1992

*To be presented at the Society of Women Engineers National Convention, Orlando, June 22–28, 1992.
†Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

OBJECT-ORIENTED ANALYSIS AND DATABASE DESIGN FOR THE SSC MAGNETS

SANDRA L. ROBINSON
Computer Engineer
Superconducting Super Collider Laboratory
2550 Beckleymeade Ave., MS-1002
Dallas, TX 75237
(214) 708-2939, FAX (214) 708-2863

ABSTRACT

The Superconducting Super Collider will be the world's largest particle accelerator, and will answer many fundamental particle physics questions. It will be 87 km (54 mi) in circumference running through a tunnel 15 to 75 meters underground. Two proton beams accelerated to 20 TeV will produce 100 million proton collisions per second. These protons will be guided by a series of superconducting magnets. We are developing a central magnet database accessible by scientists via network connections. This information will help scientists study magnet production and performance. We are using an object-based development approach. Through a combination of tools, we are transforming object diagrams into entity-relationship models and uploading SQL objects into a Sybase database on a Sun/Unix platform.

Background and Scope of Application

Some 10,000 superconducting magnets will guide two proton beams around the 87 km (54 mi) Superconducting Super Collider main ring to collide at a total energy of 40 Tev. The particles resulting from the proton-proton collisions will be detected and studied by scientists in order to gain new and fundamental knowledge governing the underlying laws of physics.

The magnets to be used in the Collider, which is scheduled for operation in 1999, will be built primarily by industrial vendors under contract to the Magnet Systems Division. The Data Management Group is developing a database to integrate all data relevant to analysis of the test results on the magnets. This includes configuration information as well as test results for critical components and materials. Data are collected at major production and testing steps. The total set of magnet data are projected to be approximately 100 Gbytes.

Why have a central database? One centralized database ensures that the identical data are available to all users. Magnet builders, accelerator designers, and operators will be able to access the same data from all magnets and vendors for the study of trends and performance among the various designs. We intend to develop a flexible, well-engineered interface for ease of access to magnet data.

Object-Oriented Development Approach

We are taking an object-oriented approach to modeling the data for the database. This means that we quantize the data into discrete entities called objects. Objects with the same data structure (attributes) and behavior are grouped into a class. Each object then becomes an instance of the class. Operators may use different methods for different classes (e.g., different methods of an operator "print" could be implemented for the different file classes ASCII, binary, and picture files). Attributes and operations may be shared among classes, or inherited. In contrast, conventional data modeling did not have the data structure incorporated with the data behavior; instead, these were loosely connected.

In object-oriented development, the application domain is identified and organized, rather than focusing too early on implementation issues. The result is a less fragile system, one

capable of change, and one that more closely defines the real world. The main benefits of this approach is not reduced development time--indeed, development time is no less than that for conventional approaches. However, revisions are easier and faster because they are more localized. Also, there are fewer revisions because more problems are uncovered during development.

There are various object-oriented diagramming notations available. We have selected Rumbaugh's notation (see references) for our analysis diagrams. Through many discussions with our database users, diagrams are drawn up and reviewed by informal walkthroughs. A typical set of classes is shown in Figure 1 with their associations. Figure 2 shows some of the Rumbaugh notation used. We are using a commercial product called Paradigm Plus from ProtoSoft, Inc., a Houston-based company. This product supports a variety of methodologies, including Rumbaugh, and a customizable version is also available. This flexibility is ideal for our research environment.

Database Tools

The database management system (DBMS) will provide the capability to store, access, and manipulate data. However, database definitions and queries are large and hard to maintain. We want to preserve the application-specific data structure and operations created by our object-oriented approach, and incorporate these definitions into the database implementation. We have thus adopted a set of database tools which were developed at Lawrence Berkeley Laboratory (LBL) for the Human Genome Project. These tools are aimed at increasing the productivity, efficiency, and accuracy of the DBMS, while maintaining the underlying application-specific data structures.

There are three primary tools in the LBL toolset. The first is a graphical schema specification tool called ERDRAW. It is based on an extended entity-relationship model. ERDRAW is an X11 based drawing program that can generate postscript files for documentation. In generating these files, we transform the object-oriented (OO) diagrams to entity-relationship diagrams. We hope to produce code to perform this function automatically in the future. Currently, the OO diagrams are consulted when the screen entry is done in ERDRAW. The view seen from the screen is shown in Figure 3. Each entity shown can be expanded by a click of the mouse to view attributes. Both pop-up and pull-down menus are used to define items and actions. ERDRAW also generates a file which is used as input to the second tool, a database schema design and translation tool (SDT).

SDT takes the extended entity-relationship schemas and generates relational DBMS schemas. The SQL code that is generated may then be loaded into the DBMS to create the database. Triggers and procedures are generated for verifying the consistency of the database regarding the referential integrity constraints associated with that database. The LBL tools support Sybase, Ingres, and Informix. We are using Sybase on a Sun/UNIX platform. Finally, SDT generates metadata and subject terms and maps them for use with the third tool in the toolset.

The third tool is a family of utilities for specifying and translating object queries interactively. This tool is designed for non-DBMS users. The user is guided through the process of specifying queries by selecting items (objects, attributes, operators, and values) on the screen. The architecture of the query tool is shown in Figure 4. The major components of the query tool include query specification and translation modules which produce a complete query for directly linking to the database. See the LBL references for a more complete description.

Data Input/Output

One of our most difficult problems in the early stages of our database development involved importing data from a wide variety of sources, platforms, formats, and media. The flow of data will become more standardized as the Laboratory grows, but we initially must go through the birthing pains of a new laboratory and new programs. This includes not only identifying the contents (objects) and organization of the database, but the sources of information and standard

identification schemes. We have used the importation/exportation utilities provided by existing DBMS's as well as customized software for importing data. This software is usually written in C or C++ or with Unix utilities.

Output of the data also comes from different paths. We have developed a menu system which can be used by local or remote sites for database access. A menu tree allows selection from standard reports. Custom code is also produced for some summary and detailed reports. In addition, we utilize the Sybase Report Writer utility. We are using commercial graphical packages such as PV-Wave to generate plots for our scientific users. We hope to have a direct interface between these tools to Sybase eventually, but we currently generate interfaces to provide needed functions. Finally, we plan to use the querying tool from the LBL toolset for some of our output needs.

Conclusions

We believe that it is important to establish a development methodology. This process organizes our development effort and provides the discipline needed for well-engineered products. We believe that the object-oriented approach will help us model the real world accurately, and will allow our system the flexibility and adaptability required for the inevitable changes that occur during the lifetime of any system. Our goal is to produce an easy to maintain and easy to extend system to grow with the needs of our scientific user community. The object model provides a natural unit of modularity to support this goal.

References:

Ball, M. et al, "The Magnet Database System," IISSC Proceedings, March 1992.

Markowitz, V., Shoshani, A., "Query Specification and Translation Tools," Lawrence Berkeley Laboratory Technical Report LBL-31155, October 1991.

Markowitz, V., Fang, W., "SDT, A Database Schema Design and Translation Tool," Technical Report LBL-27843, May 1991.

Robinson, S., "The Superconducting Super Collider Project," Society of Women Engineers National Convention and Student Conference, June 1991.

Rumbaugh, J. et al, Object-Oriented Modeling and Design, Prentice Hall, 1991.

Szeto, E., Markowitz, V., "ERDRAW, A Graphical Schema Specification Tool," Technical Report LBL-PUB-3084, Draft 2.2, May 1991.

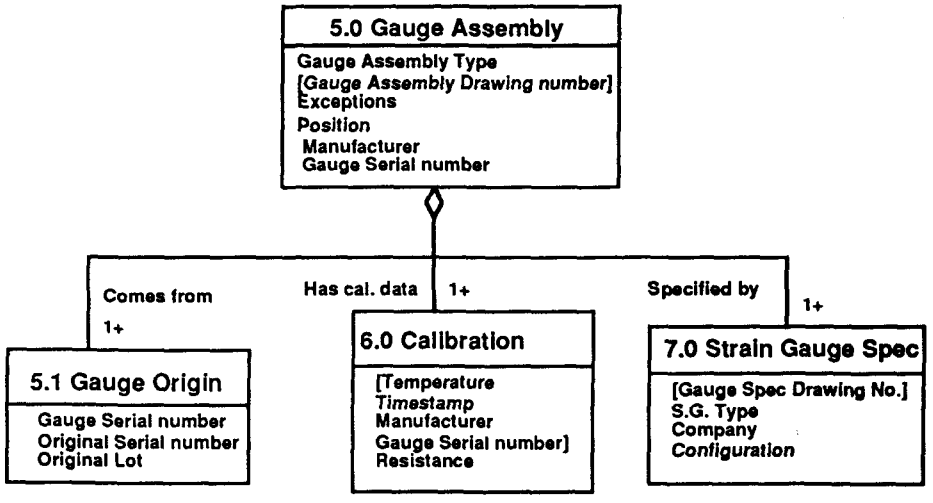


Fig. 1: *OOA Diagram for Strain Gauge Instrumentation*

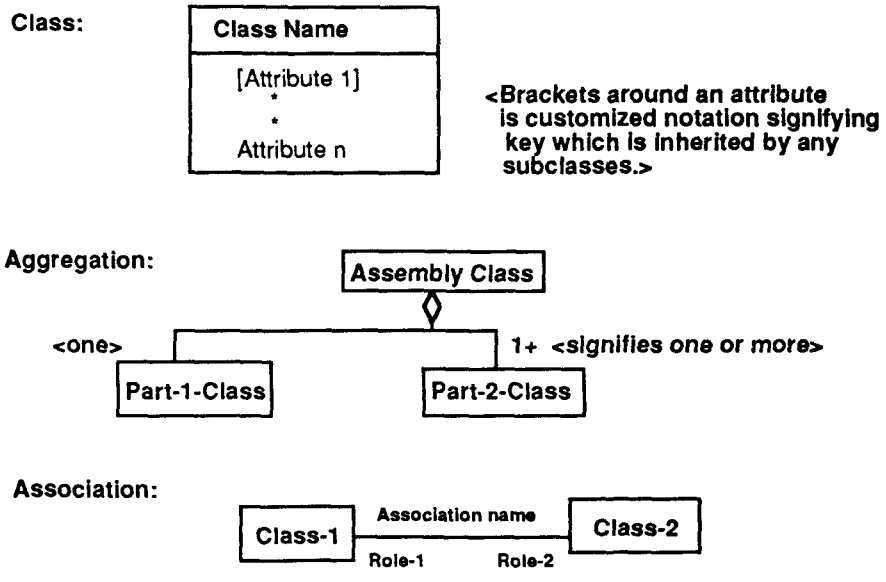


Fig. 2: Rumbaugh Object Model Notation Subset with <notes>

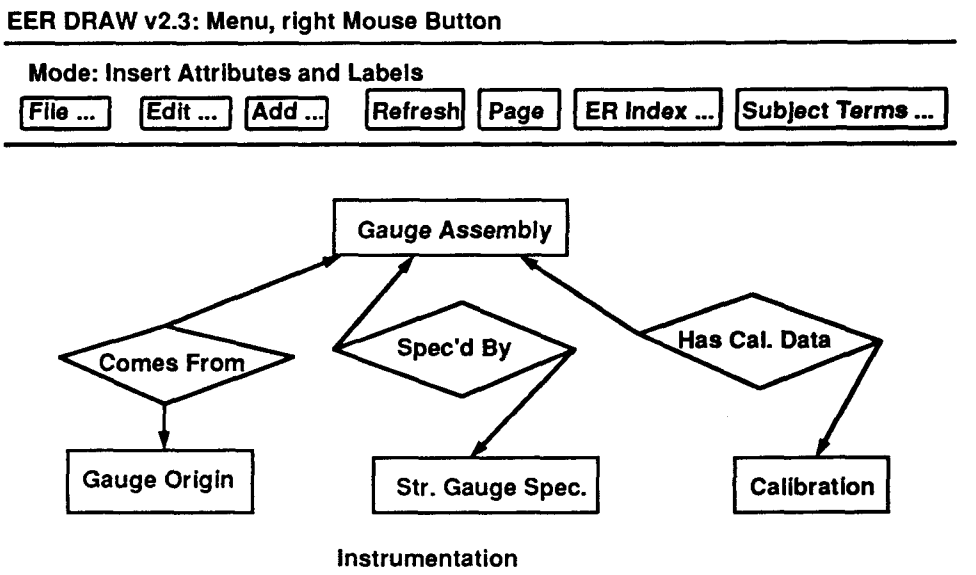


Fig. 3: ERDRAW screen view

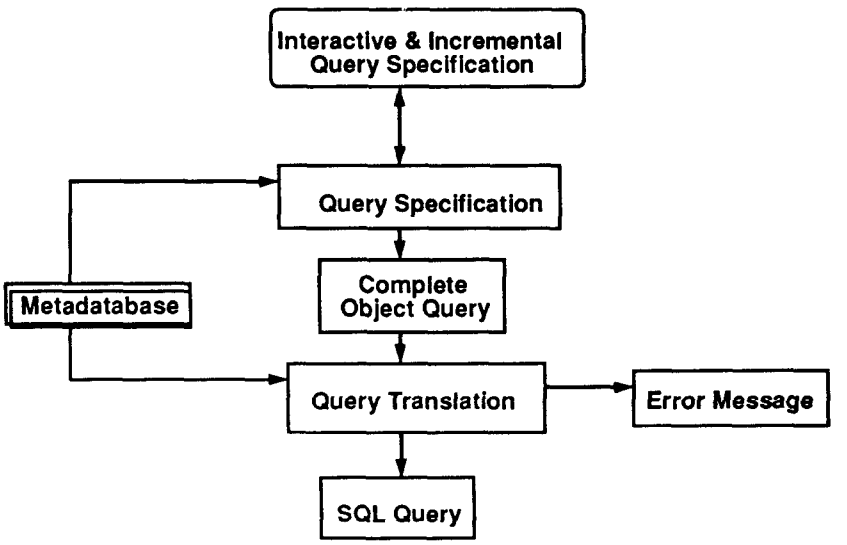


Fig. 4: The Architecture of the Query Tool