SLAC-PUB-12607
June 2007

DATABASE APPLICATIONS TO INTEGRATE BEAM LINE OPTICS CHANGES WITH ENGINEERING DATABASES*

A.Chan[#], P.Bellomo, G.R. Crane, P. Emma, E. Grunhaus, K. Luchini, I.A. MacGregor, D.S. Marsh, R. Pope, P. Prickett, C. Rago, K.Ratcliffe, T. Shab, SLAC, Menlo Park, CA 94025 U.S.A

Abstract

The LCLS project databases provide key nomenclature information while integrating many engineering and physics processes in the building of an accelerator. Starting with the elements existing in the beam line optics files, the engineers add non-beam-line elements, and controls engineers assign 'Formal Device Names' to these elements. Inventory, power supplies, racks, crates and cable plants are databases that are being integrated into the project database. This approach replaces individual spreadsheets and/or integrates standalone existing institutional databases.

STREAMLINING DATA AND WORK PROCESSES

In the past, data relating to engineering and physics were kept in disparate spreadsheets and files. Although changes to beam line optics elements can affect many engineering processes downstream, such changes had to be incorporated manually into these various spreadsheets and files. Within LCLS, the goal of the project database is to integrate the engineering and physics systems, as well as integrate with legacy SLAC databases already in use. We focused on processes that are commonly needed during the construction as well as the operations phase.

Figure 1 is an overview of the different information systems needed by LCLS.

BEAM LINE OPTICS ELEMENTS

Since the conceptual accelerator model is based on the beam line optics output (MAD output), a database to store the different versions of the MAD deck was an important starting point for the project database.

Additional functions were built into the database to compare older and newer versions of the MAD deck for any of the four LCLS beam lines. Figure 2 shows a report from the database highlighting elements that were added, removed, or changed in the comparison of two versions of the MAD deck.

A DIRECTORY OF ALL DEVICES

The physicists, the engineers, and the controls group often refer to the same accelerator device (or element) by different aliases, different coordinate systems, and different units of measurement. Similar to how a phone directory for people streamlines organization work, we

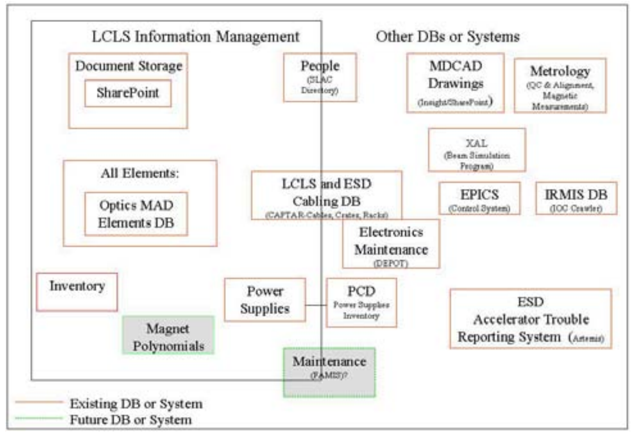


Figure 1: Project-specific and site-wide databases used by LCLS.

needed a directory of all devices where key attributes are stored and mapped to each other, in order for cross-functional groups to quickly translate aliases and coordinates. With the backbone of elements from the MAD deck in the database, we built an All Devices database where the mechanical and electrical engineers can add their devices and attributes, and where the controls group can add the 'Formal Device Names' (controls names). Database triggers orchestrate how the data are placed in the All Devices database so that the newest version from the MAD deck is used. We used Oracle's Application Express interface (APEX), which works together with the database triggers to provide a uniform view of MAD and non-MAD data in one editable report (see Fig. 3).

From the All Devices database, one is able to select information stored for that element in the cabling database, power supplies database, inventory database, etc.

DATABASE FOR CABLE PLANT, CRATES AND RACK PROFILES

The engineers and controls group invest a lot of time configuring racks, crates and cables. SLAC has a legacy cabling database (CAPTAR) that holds the official cable plant configuration (see Fig. 4 for a rack profile from the CAPTAR database).

However, this database lacks referential integrity support, which opens the door for inconsistent data within the database. Only one legacy Oracle Forms application is able to enforce the needed integrity between the 60+ tables. Therefore, in order to provide

* This work was performed in support of the LCLS project at SLAC and funded by Department of Energy contract DE-AC02-76SF00515.
[#]achan@slac.stanford.edu

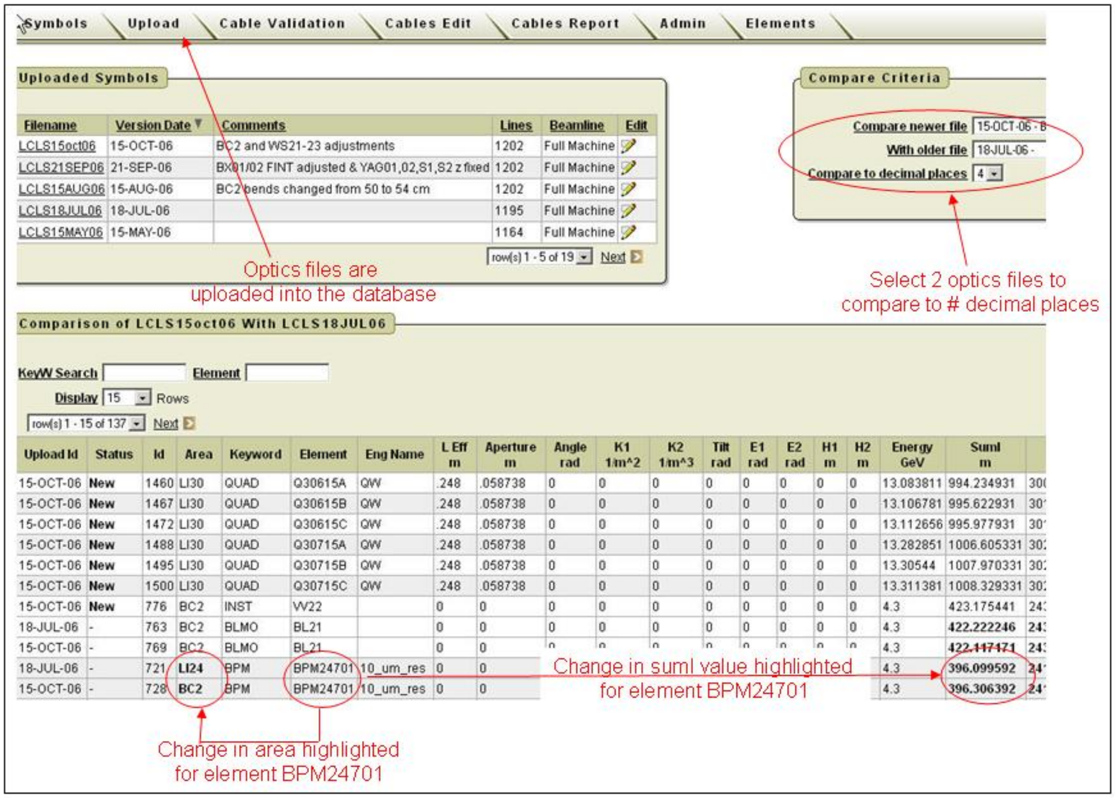


Figure 2: Comparison program for different MAD deck file versions.

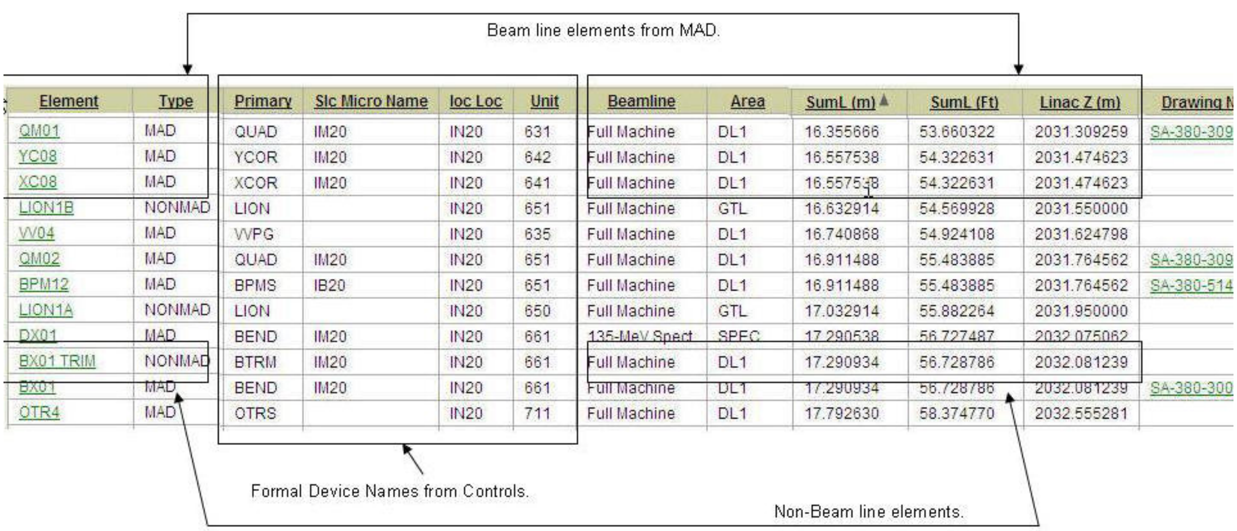
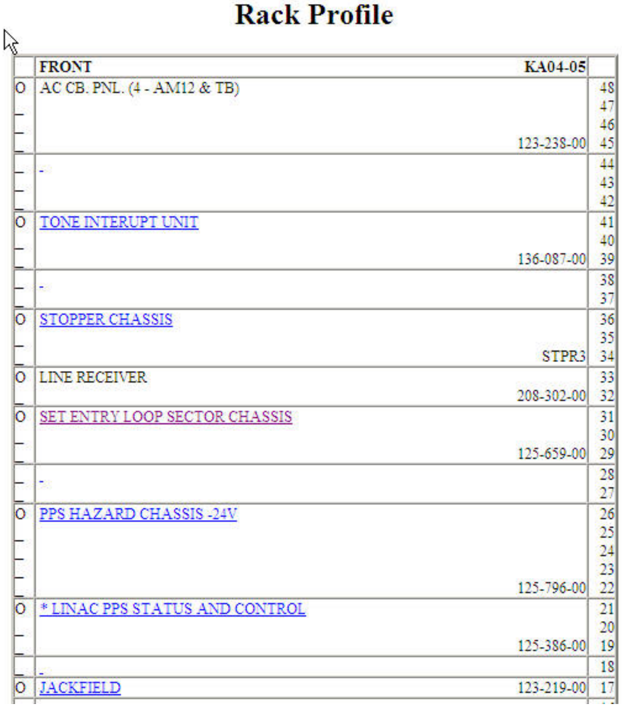


Figure 3: Sources of data for the All-Devices database.

- new functionality for LCLS
- integration of different types of data (LCLS and CAPTAR)
- change history
- workflow for review and approval of changes

we built a staging database to interface with CAPTAR, so that we can add these new functions without having to re-build CAPTAR. Users will be able to add or edit cable plant data and circulate a version for review before sending the configuration to the CAPTAR administrators for approval and loading into the CAPTAR database (see Fig. 5).



Rack Profile

	FRONT	KA04-05	
O	AC CB. PNL. (4 - AM12 & TB)	123-238-00	48 47 46 45
	-		44 43 42
O	TONE INTERUPT UNIT	136-087-00	41 40 39
	-		38 37
O	STOPPER CHASSIS	STPR3	36 35 34
O	LINE RECEIVER	208-302-00	33 32
O	SET ENTRY LOOP SECTOR CHASSIS	125-659-00	31 30 29
	-		28 27
O	PPS HAZARD CHASSIS -24V	125-796-00	26 25 24 23 22
O	* LINAC PPS STATUS AND CONTROL	125-386-00	21 20 19
	-		18
O	JACKFIELD	123-219-00	17

Figure 4: Rack profile from CAPTAR database.

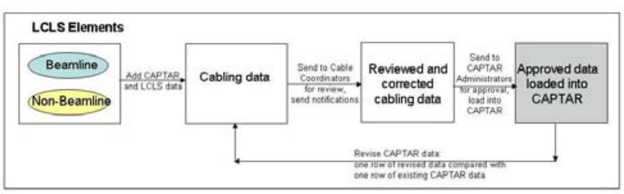


Figure 5: Workflow for editing, reviewing and approving cable plant data.

INTEGRATING SCHEMATICS AND DATABASE

It is useful for users to transparently move between a graphical view of the elements and the information stored in the database. A prototype program has been developed that scans the schematics for a match to the unique element names to concatenate a hyperlink to the element's database record (see Fig. 6).

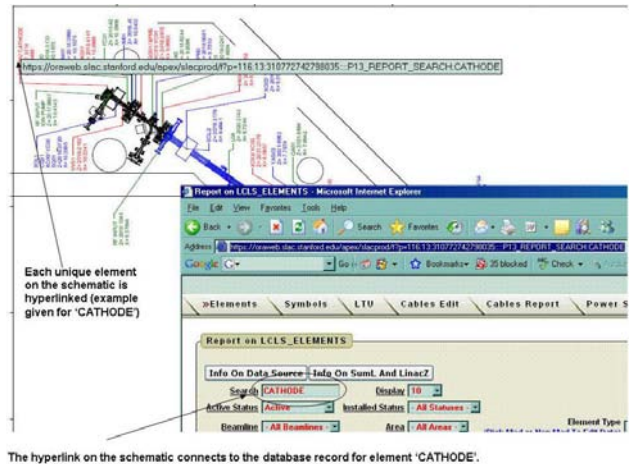


Figure 6: Schematic drawing in PDF format, with links to the elements' records in the database.

DEVELOPMENT TOOLS

APEX is a tool that allows those familiar with SQL and Oracle's PL/SQL language to build, test, and deploy database-centric web applications in significantly less time than with other web-based Integration Development Environments.

This tool provides an intuitive interface in which developers "declare" more than "code" an application. Such declarative-type programming makes for quick turn-around times between capturing/documenting user requirements and presenting an application usable with any industry-standard web browser.

APEX automatically takes advantage of Oracle's built-in set of tools and functions such as Single Sign On (SSO), table referential integrity, optimistic locking, and trigger/sequence integration within web screens so as to allow developers to concentrate on the presentation of data, rather than on how that data should be saved or modified within the underlying Oracle tables.

ACKNOWLEDGMENT

We thank the LCLS management and staff, and SLAC's SCCS staff. Without their vision and their many contributions, this work would not have been possible.