The LHC Super-Table Project

Elliott McCrory, LAFS

17 June 2010

It is important to benefit from the experience at Fermilab’s Tevatron in developing a Super-Table for the LHC. We outline the goals and the basic requirements for this project, and then break the project into sub-tasks.

Please refer to the document “Collider Super-Table for a Modern Hadron Collider”, by Dave McGinnis; September 5, 2008

# Goals

We believe that the basic goals of an LHC Super-Table are

1. Provide a summary of the key parameters that detail the performance of the LHC.
2. Provide a concrete mechanism for determining these key parameters on which everyone can agree (or, at least, through which the parties that disagree can have a basis for their argument).
3. The Super-Table is available in forms that everyone understands: Web pages and spreadsheets.
4. The Super-Table should be small.

### Comments on these goals

We suggest that the important “Key Parameters” for the LHC be only those that directly impact the delivered luminosity or that delineate a description of the fill. The former is the intensities and the emittances; the latter is things like the time stamps of the fill and the fill pattern. Goal no 4 should be kept in mind when contemplating the addition of new parameters.

The existence and acceptance of a *unique* Super-Table satisfies goal no 2.

No 3: At Fermilab, we also provide:

* An AIDA interface to the Super-Table, and
* A GUI that accesses the Super-Table database to make scatter plots from these data.

## Secondary Goals

Secondary goals, which have proven to be very useful, are,

1. Each Super-Table cell is calculated from data in existing databases. In other words, the Super-Table does not collect live data.
2. The algorithm for calculating a cell changes from time to time, so the implementation of these algorithms must be flexible.
3. When an algorithm changes, it is necessary to recalculate this cell for old fills.
4. Columns will be added to and removed from the Super-Table

Footnote to no 2: By “algorithm,” we mean the general way to calculate the content of a cell. This can be as simple as the fill number or a parameter’s exact value from the logging database, and as complicated as the calculation of the luminosity lifetime, based on the raw luminosity data (i.e., performing a decay fit).

We will comment further on these secondary goals in the remainder of this memo.

# The View from Fermilab

We hope to be able to convey our experience to LHC Operations so that the LHC Super-Table is the best it can be.

## Some Fermilab Implementation Concepts

The Tevatron Super-Table project has developed into these separate and distinct projects:

* **Sequenced Data Acquisition (SDA).** This is a data acquisition (DAQ) system that is driven by transitions in the Beam Mode (we call them “Cases” and “Sets”) in the Tevatron Collider Complex. The instructions for this DAQ and for the collected data are stored in the SDA database. This is a stand-alone DB that is separate from our Logging databases. We have several GUI tools for dealing with this system, including:
	+ Managing the DAQ instructions
	+ Examining the messages from SDA, especially the errors
	+ Examining the data

Web site: <http://www-bd.fnal.gov/sda/>

* **Offline SDA Analysis API (OSDA).** This is a Java API that is the basis for the creation of the Super-Table. It includes an API into the SDA database, and also a façade for access to the Logging databases. If a scientists needs to analyze SDA data in new ways, she would write a program using this API.

API Web site:<http://www-bd.fnal.gov/javadoc/build/api/gov/fnal/controls/applications/osda/package-summary.html>

* **Super-Table**. There is a Java application that is run at several points in the Collider cycle to populate the Super-Table. Additionally, the Super-Table can be recalculated at any time (by an authorized individual).

Web sites: <http://www-bd.fnal.gov/sda/supertable/index_IV.html> and <http://www-bd.fnal.gov/SDAMisc/AJAX/STIV.html>

* **Mini-Tables**. There exist several other Excel-ready tables that we call the Mini-Tables. This name is a bit misleading as these tables usually have information that is more detailed than the Super-Table. These tables, also derived through the OSDA API, are specified and used by machine experts.

Web site (incomplete): <http://www-bd.fnal.gov/SDA_Viewer/more_tables.html>

## Some Lessons Learned

We learned rather quickly that the contents of the Super-Table, especially in the beginning, changes frequently. We quickly settled on a Super-Table database schema that allows the columns to change easily. We recommend this schema for the LHC Super-Table.

The Tevatron Super-Table is no longer small. We recognize this and propose that the LHC Super-Table be hierarchical: Level 1, Level 2, etc. We encourage you to keep the LHC Super-Table small and focused.

The Tevatron Super-Table schema has no provision for storing the algorithm, units or error messages. We propose that these items be included in the schema.

## Implementation Suggestions

We suggest the following implementation decisions:

1. Data collection for the Super-Table is separate from the creation of the Super-Table.
2. The Super-Table creation process writes to a stand-alone Super-Table database
3. The schema for the Super-Table database contains several tables, but the table containing the actual data in the Super-Table cells is independent of the column definitions. We propose a simple schema, described in “Data Storage Schema for the LHC Super-Table. “
4. The mechanics of creating the Super-Table is a Model-View-Controller (MVC) pattern:
	1. **Model**: The Super-Table database and I/O with that database
	2. **View**: The creation of the web pages and spreadsheet files
	3. **Controller**: Calculating the (new) contents of the cells for a fill or a series of fills.

 We propose a specific architecture for the Super-Table MVC in “A Proposal for a Framework for Building the LHC Super-Table.”

# LHC Super-Table Projects

In our opinion, the Super-Table project can be divided into separate sub-projects, most of which are underway.

1. Define what data should be in the Super-Table. Define the calculations to transform data from the raw Logging (or Measurement?) DB and Mario’s DB (below) into Super-Table data – Mike, Verena and Mario
2. Beam Mode-triggered DAQ, with the data stored in the Logging (or Measurement?) DB – Mario.  He has made a fantastic start on this already!
3. Retrieving the appropriate data from the available LHC databases for consumption by the Super-Table process – Mario and Elliott [MVC: Model]
4. Transforming raw data from the LHC databases into the Super-Table, and storing these data appropriately – Elliott [MVC: Controller]
5. Transforming the Super-Table stored data into web pages and Excel-ready files. – Elliott. [MVC: View] (There can be other views constructed into the Super-Table data.)

## Sub-Project Contracts

Here is a summary of the input and output for each of these sub-projects

|  |  |  |
| --- | --- | --- |
| **Sub-Project** | **Inputs** | **Outputs** |
| 1. Defining Columns
 | Knowledge of the LHC Collider program | A spreadsheet (or similar document) that contains column names and details on the data sources and the algorithms for each column. |
| 1. Event-Triggered DAQ
 | Knowledge of the data in the LHC | Event-driven DAQ database table(s). (Logging DB?) |
| 1. Retrieving Data
 | Sub-Projects 1 and 2 | A Java API for accessing these data |
| 1. Cell Calculations
 | Sub-Project 3 | The Super-Table database |
| 1. Views
 | Sub-Project 4 | HTML and tab-delimited data files |