Table of Contents

Minutes of the Data Quality Meeting of 30 June 2008

Problems database

We need to create a problems database to log all problems encountered while running. Typically shifters would add a new problems and experts would debug and close the problem.

The requirements are

- It must be accessible from the pit and the outside world.
- It must be easy to browse, search.
- One must easily be able to get the list of active problems, and the list of problems affecting a given run.
- It must be easy to add a new problem.
- It would be nice to be able to link a problem to a particular histogram and vice-versa.

Many of these requirements match what Savannah and Launchpad do. The former seems more appropriate and is supported by CERN. But they both focus on software developments. We would be abusing the system, using the run number as a sort of version. There was some concern about the user-friendliness of Savannah, but the alternative would be to write an interface that would also have to be learned (especially for Savannah users). **Action : Patrick to investigate how easy one can configure it**.

It was also not clear if one would locate it at IT or at the pit. It depends on the firewall at the pit. **Action : Patrick to see with Niko what is more appropriate**.

Bookkeeping database

The bookkeeping database would contain a pointer ot a data quality table for every file. Data quality experts would then update this table according to what is found. The file is the right granularity for the bookkeeping. Data quality experts would probably more deal with runs or fills, and then this would propagate to the files.

There was some discussion about the work flow of the DQ experts. We decided that in 2008 one would operate in **red light** mode, meaning that only data flagged as good by the experts would be processed. We have several days worth of disk space on the Tier1 sites.

From then on the only reason for a DST to become bad is that the whole production is bad. One exception is the case when some raw data is identified as bad after the processing occurred. The basic assumption is that this is rare and that one would then mark the whole sample as bad. Trying to recover good events in a bad file is impractical. LHCb is a precision experiment, we cannot deal with non-perfect data. In such a case the DQ flags would be propagated to the decendents of the file.

As for the table itself, we could not identify good reasons for having a fine grain structure. **One global flag is good enough**, and we'll revise this later. The values this flag can take are `GOOD`, `MAYBE`, `BAD` and `UNCHECKED`. The last would be default. **Action : Bookkeeping people to provide a DQ table in the bookkeeping database**.

The single flag implies there's a single decision made for every file (practically, for every run). The subdetector and working group experts will have to meet regularly (initially daily on week days) and decide for each run. An expert will then flag the runs in the bookkeeping, which will trigger the reconstruction. Ideally there should be a very limited number of people allow to do this. Typically the DQ responsible and a deputy.

There was a question about how we would freeze data samples for conferences. This would be done by running a query at a given moment in time and storing the result. The bookkeeping database would not have to know about this. If something needs to be changed lter, one would have to reproduce this data list and advertise it.

Trigger bits

There are 96 bits written by the HLT into the data which will be used to determine which events go to which task in the monitoring farm (MF). The last 32 are for internal handling in the MF. Clients of these bits will have to provide

- The task to be run in the MF.
- The **code** to be run in the HLT taht sets this bit. The trigger will not do that for you.

Offline histogramming

Brunel will produce histograms. These can be produced online in the monitoring farm and offline in the reconstruction. The main difference is that in the latter the final calibration is available (if changed) and the whole sample is processed. The 50 Hz Brunel process run in the MF corresponds to 30000 events per 10 minutes cycle. A 30 second file will contain 60000 events, i.e. roughly the same number. Most monitoring and calibration that needs tracks will have to be done in the monitoring farm. The monitoring farm will process more events a day than we have ever used from MC.

Monitoring requiring more data would have to be done by users on the output of the stripping.

Yet one could imagine the need for looking at histograms for more than one file, i.e. merge the histograms coming out of Brunel offline. Some issues:

- A clear uses case has to be provided.
- Events are not sequential at the file level.
- The concept of Run number is lost during the stripping.

It was decided that this was the responsibility of the DQ team until a clear use-case was found. It wll be revised for 2009.

Actions:

- Production to provide a clear labelling of histogram files (containing run number)
- Patrick to write a script retrieving and merging files. It can be tested on MC production

Summary of action list:

- Patrick to investigate Savannah. Where would one run it?
- DQ table in the bookkeeping database to be provided by Computing group
- Production to provide a clear labelling of histogram files (containing run number)
- Patrick to write a script retrieving and merging files. It can be tested on MC production

-- PatrickKoppenburg - 30 Jun 2008

This topic: LHCb > DataQualityMinutes080630
Topic revision: r4 - 2018-09-23 - MarcoCattaneo