

# Fast Monitoring user interfaces \*

\* apart from JAS itself

Gregory Dubois-Felsmann

2 December 1999

**This presentation is mostly about the control of Fast Monitoring. Alex will talk about display issues later. (I'll allude to some, but if I don't mention your favorite one, please be patient and bring it up then.)**

---

# Present state

The user interface to Fast Monitoring is presently primitive:

- The Fast Monitoring data accumulation process and
- the JAS-based display,  
  
(largely independent of each other) are presently started by shell commands, typed by the DQM shift-taker.
- The data accumulation process sees only a tiny fraction of the full data stream, as it operates via trickle stream from only a single one of the 32 OEP nodes.
- Terminal windows showing output from the accumulation process, the JAS interface server which is paired with it, and JAS itself, appear on the display.
- DQM's are expected to browse through one JAS-HTML page of histograms per detector subsystem, evaluating the quality of the data based on written remarks in the subsystem HTML and non-overlaid reference histograms.
- The system isn't very reliable or transparent.

This is not the system originally designed, but rather a stop-gap assembled from off-the-shelf components when unplanned personnel shortfalls (several people were moved from OEP to other responsibilities) and missing external contributions prevented the completion of the designed system.

So, we claim: the original design remains viable, addresses most of the concerns expressed about the present scheme, and is in the process of implementation.

---

# Near-term plans (December-January)

There are two principal present projects:

We are about to begin the migration from trickle-stream, single-node data accumulation to multi-node operation.

From the users' point of view:

- **Much more data will be processed per unit time, rendering all histograms and displays more significant.**
- **Fast Monitoring will see the first events of a run.**
- **It will become practical to perform monitoring, as appropriate to each subsystem's needs, on selected data streams: Level 1 Accepts, Level 3 Accepts, Bhabhas, etc. (based on the Level 3 output lines), permitting certain tests to be made much more sensitive.**
- **Fast Monitoring data accumulation and DHP service will be started automatically by Run Control. (JAS - see below - will still be started on the command line, asynchronously to this.)**
- **Serious errors in the accumulation process will be logged to CMLOG and will be displayed in a dedicated Fast Monitoring error message browser, using the new CMLOG 2.0 tools. Lesser messages will go to log files, and we will lean on subsystems to thin these out as much as possible.**
- **Histograms will be stored in the conditions database on a run-by-run basis, entirely under the control of Run Control.**
- **The "advanced trickle stream" will allow remaining non-Fast-Monitoring trickle stream clients (event displays) to see all state transitions, including Configure and BeginRun.**

## **Near-term plans, cont.**

**Automated histogram monitoring is being commissioned now and should be ready for limited application on the same time scale.**

- **Automated monitoring, with an initially small set of comparisons contributed by the subsystems, will also be under Run Control control.**
  - **Messages will go to the dedicated Fast Monitoring CMLOG browser. (Airtouch pages can be generated for high-priority problems.)**
  - **Statistical comparisons to reference histograms and analytic spectra, and bin-by-bin occupancy comparisons, will be available at first.**
  - **Subsystems are just now starting to provide the first inputs to the system.**
-

# Near-term plans, cont.

There will be a transitional regime:

- The single-node trickle-stream monitoring may continue for a short while as a backup as the new framework is confronted with real data in January, and as subsystems learn how to exploit its new capabilities.
- Until people are fully comfortable with the database histogram retrieval tools, we will continue to write HBOOK files.

A Web page will be available for DQMs which, in addition to the presently available basic operating instructions, will list known problems with the operation of the Fast Monitoring system and with the contents of the subsystems' histograms and HTML pages. It will be based on Remedy. (This turned out to be highly non-trivial to accomplish, due to software licence restrictions (!), but with a lot of help from Charlotte Hee it is almost ready to release now.)

This is intended to be a readily accessible means of communication from one DQM to the next, and between DQMs, core OEP developers, and subsystem Fast Monitoring experts. We think that Remedy's problem reporting and tracking abilities will prove to be quite appropriate for this.

---

# Medium-term plans (January-April)

Some of these are simple things that we just won't get to before then.

- A control GUI for the JAS presenter, replacing the command line "LiveFastMonJas" and "LiveFastMonJasCleanup" tools. We *don't* want this to be under the central Run Control as long as JAS has not reached much higher levels of reliability, but it should take no more than two or three buttons on a small, permanently present GUI to allow DQMs to manage their JAS display. The same tool should allow remote users (e.g., in DataFlow) to start their own JAS to look at the live histograms. JAS error messages will go to a log file, but not to CMLOG (unfortunately).
- Automatic Run Control startup of an event display or two (the Level 3 one, presumably, with a couple of different selection criteria).
- A fairly primitive GUI for browsing through the stored histogram sets, selecting ones for extraction as HBOOK files or for live viewing. A prototype exists based on BrowserApp; some modest additional work will allow it to bring up a data server for JAS. NB: However, this tool cannot easily be integrated into the JAS display itself - see Alex' talk.

Some require subsystem development that can realistically only begin when the near-term work has been completed.

- Useful exploitation of event selection to optimize FM sensitivity.
- Extensive and well-tuned use of automated monitoring.

We'll have a existing Caltech graduate student moving to SLAC starting early in the year, who will be taking on the subsystem coordination role which has long been understaffed, and this should help with a lot of the above issues.

# Medium-term plans, cont.

And some will need additional, planned, core development to be completed.

- **Dynamic load balancing between subclasses of Fast Monitoring. This will become more important as the Level 3 load on the OEP nodes increases and FM time becomes more precious.**
- **User-configurable end-run transformations of the FM histograms, with storage to the database.**
- **Automated comparisons based on spectrum fits (Paul Bright-Thomas).**
- **Configuration database control of automated histogram comparisons, including references / expected values. (Allows convenient establishment of multiple sets of references.)**
- **Graphical editing of the automated histogram comparison control information.**

**(Once we complete the bulk of the remaining core development work, though, later in the year we believe that it will become appropriate to transfer this role to the operations team, and it would become a good candidate for a rotating collaboration service appointment.)**

---

## **Medium-term plans, cont.**

**There are also some good ideas outside the original design that have come up that we would like to provide. Here some help from outside the core group would probably speed things up quite a bit.**

- An Alarm Handler-type scheme for dealing with alerts from the automated histogram comparison system. The existing system supports CMLOG messages (which in CMLOG 2.0 can be decorated in color according to severity) and emails (and therefore pages), but it's clear that it would be good to augment this with an acknowledgement-oriented scheme like the Alarm Handler's.**

**We'll also clearly learn things during the commissioning of the upcoming round of new features - such as requirements for fine-tuning of the behavior and performance of the new system components - that will clarify what the medium-term priorities will actually be.**

---

# Longer-term options

There are still more desirable features that resources wouldn't permit us to deliver for some time, or even a long time, to come. Here, help from outside the core group could make a big difference.

- **Client-side (DQM-driven) clearing of histograms.** We long ago made this available in the JavaHist GUI prototype, but this would now require a significant amount of work in the JAS core (see Alex' talk), followed by a fair amount of BaBar-specific work. Nonetheless, it's a really important feature, especially during recommissioning and test runs, and this is the task we'd most like to be able to move up in the list.
  - **Run and connection status feedback in JAS.** (Remote users, especially, would like some confirmation in the presenter of what the status of data acquisition is and whether the presenter really has a functioning live connection to it.) We may be able to think of some interim kludges to help here, but this, too, basically requires core JAS development in order to do properly.
  - **Stitching together time histories (scalers) across multiple runs.** This is obviously desirable and conceptually simple, but fairly complicated in practice, and we lack a detailed technical design at this stage. Controlling this from JAS, as one would wish, is another "hard problem".
  - **Stitching together sets of histograms over multiple runs to make high-statistics plots.** This has been requested in the past, but we would want to wait and see how 32-node statistics actually look before embarking on this. It should also be noted that long-period monitoring may often be more appropriately performed in OPR.
-

## Longer-term options, cont.

- **Automated monitoring of trends in long-term time histories and in the history of run-by-run automated comparison results, with alarms.**
- **Interactive display of the actual results of the automated histogram comparisons, overlaid on the live JAS display. (E.g., display of fit results over the live histograms, with marks on the display showing the expected results for comparison.) Again this appears to require significant core JAS development.**

**(This should be distinguished from the overlay of live and reference histograms in JAS, which is quasi-independent of this and should be fairly straightforward once the JAS XML support - see Alex' talk - is commissioned.)**

---

# Summary

- **By January we will have a much more capable and somewhat more easily operable system.**
  - **Operability will improve in the next few months, along with the addition of selected additional capabilities.**
  - **Some valuable features will still not be there.**
    - **A lot depends on developments in JAS and the BaBar extensions to it.**
    - **Apart from this, some targeted contributions from outside the core group would help accelerate the delivery of the remaining core functionality.**
    - **In the long run, operational management of Fast Monitoring should transition from the OEP group to the operations group.**
-