.

Run 1 Online Postmortem:

DAQ View

Jan Guida
Alan Jonckheere
Philippe Laurens

July 25, 1996

Successes

- Relatively high efficiency.
  - With lots of smooth running periods, especially in 1B.

- Sort_Alarm was good:
  - need more (only did L2 and DL messages)
    Would be good to have a version for all DAQ messages.
      L2 and DL were summarized by run,
      some others (non-run dependant quantities) should be
      summarized by shift.
  - Then can use these (over an longer time period (stores, weeks, etc))
    to find other problems.

- Shift training:
  - training sessions where we "broke" the system and had them figure out
    what was wrong and fix it worked well, we should continue this.
  - for the 1C training we had some combined DAQ-detector sessions
    this leads to a better understanding of the system and hopefully
    some team work.
  - may want some type of "continuing" training for all shifters

- Monitoring was pretty good:
  - L2 Graphics
  - SYS display
  - TRGMON
  - ALARMS

Failures

- COPYCFG:
  - would sometimes delete files when they were still needed
  - info NEVER got into a database
  - required lots of disk space
  - required manpower to constantly backup files then restore when needed again (offline)
  - would sometimes be a CPU hog
  - lost heartbeat frequently
  - saved more info than was ever used/needed
    - what do we really need?

- Run Summary Database:
  - Nice "write-only" database
  - was never finished
    - not enough input from detector and physics groups on what should go into the database
    - instances where there was input, the manpower wasn't there to fill it properly
  - very few utilities exist to read this database

- Too compartmentalized on shift:
  DAQ, detector, captain, GM need to discuss their problems more and work as a team.

General comments

- Make the "TALK-type" programs (interfaces) more uniform.
  The different conventions causes confusion among inexperienced shifters.
  $\Rightarrow$ Should come up with a convention soon and advertize it.

- Need better inter-task communication.
  - needs to be faster (CALIB$\leftrightarrow$COOR$\leftrightarrow$COMM_TKR - is too slow)
  - messages go to the wrong place or get lost
    ex., COOR$\rightarrow$TAKER messages go to the alarm server, DL, etc.

- Make better use of the alarm system.
  - More processes should send alarms/info messages.
    - including all stages of the trigger, the examines,
      the HVMON program, etc.
  - Need to set priorities of alarms properly
    improve messages in HDB
      (many not filled properly or filled with a joke or not in English).

- Error messages are sometimes overwhelming.
  - Making it hard to find the important messages in the crowd.
    - Need to select out high priority messages.
    - Maybe separate windows for different priorities would help?
      (with only 4-5 levels of priorities)

- Shifters need to be better able to identify the different hardware (trigger) systems and who talks to them while defining triggers, or initializing, or etc.

- **All** programs that use "DAQ" devices should get
  ownership through COOR.
  ex., electronics examines, samus_calib, vax_local, scint_rate
  ⇒ Do not allow any new programs that do not get ownership.
  (I suspect that there will be many more of these in run 2 than in run 1.)

- Incorporate as much of the downloading (of DAQ devices)
  into COMM_TKR as possible,
  eliminating stand-a-lone programs. (ex., clock servers)

- Problems with special runs:
  run summaries, sort alarms, etc. were not generated properly.

⇒ special runs need to be designed into the system

- DAQ's L2/DL alarm logger was good.
  - it allowed us to see these alarm/info messages easier.
    info messages disappear quickly from the alarm display.
  - at times we had too many messages,
    so it was hard to pick out the important ones.

- Alarm system floods too easily and
  is too slow when there are many alarms.
  - Must wait for alarm display to catch up before you can look at anything.
  - Need a way to send RMI resets to all RMI's (after power outages).

- If a program detects an error communicating to another program or
  hardware device, it should not just blindly continue in all cases.

- We have order-dependent downloads.
  - COOR does not handle these properly (block_tkr command does not work for multiple triggers).
  - must assume there will continue to be some order dependence in the future.

- All log files should have time stamps.

- Improve store keeper so that it keeps better track of downtime.
  - Presently ignores downtime between end run and end store.
  - There exists downtimes that are not marked.
  - Other problems.

- Luminosity monitoring in level-1 was good, but needs improvements.
  - Need to verify that right scalars are in level-1.
  - Define interface and protocol to retrieve online info and compute luminosity-related quantities.

  $\Rightarrow$ Plan the database and servers that will accomplish this.

- Be able to modify the state of a system/program without deleting lines from the input files.

  ex., to remove a disk from data logger's list, you must remove lines from LOGGER$CONFIG_DEVICES.DAT file
  - would be better if you could just comment them out
  - when the disk is fixed and you want to restore the files, a different person is on shift.

    If they just have to uncomment out the lines, that decreases the probability of an error.

- Proliferation of CFG files:
  CFG and CMS areas filled up.
  $\Rightarrow$ slow access by COOR and also CFG_FILES, TRIG_PARSE, etc.

- GLBMON - global display was useful for DAQ
  - some modifications on what is plotted would be nice
    ex., data logger - don't just plot ALL stream,
      but plot rate to DL in mb/sec (not Hz)
  - If GLBMON is not updating, was sometimes difficult to figure out
    if it was a GLBMON problem or a problem with one of the
    programs it gets information from.
  - time scale (x-axis) was not easy to understand
  - many shifters (DAQ and GM) didn't know how to run it

- Luminosity may fall faster in run 2 than in run 1.
  Should we consider an automatic adjustment of prescales?

- Most monitoring and "talk" programs started out as
  expert-level programs $\rightarrow$ operations group.
    Do we want different programs for the two different groups?

- Do we want notebook computers for diagnostics (hardware)?

Ending Comments

Always need to test the modified system during shot setup, even if only *small* changes occurred ("that could not effect anything").

- new L2 exe
- hardware changes
- new trigger list
- etc.

Need a more automated way of looking for and reporting back some of the problems.

BUT DON'T AUTOMATE TOO MUCH!!!

- important for people to look at things carefully, which **doesn't happen** if problems are "automatically handled"

- if things are too transparent to shifters, they will never understand the system well enough to be able to fix problems.

Shift-type Stuff

Suggestions made for run 1C, but did not get implemented.

Control room:

		DET	DAQ			
	OD				GM	
HV						spare
	spare		CAPT			for experts
	for experts					

GM's run quality $\longrightarrow$ Captain