

Online Software Status

Murrough Landon – 10 November 2005

`http://www.hep.ph.qmul.ac.uk/~landon/talks`

Overview

- Introduction
- TDAQ updates
- Recent developments
- Database
- Next steps

Introduction

Overview

- Online software handles configuration, control, monitoring, calibration and testing of the trigger hardware
- L1Calo developments are based on underlying TDAQ software
- We have used it for developing prototypes, taking data at the testbeam and will use it for Tile/LAr cable installation as well as final ATLAS
- Some parts (hardware access and simulation) are mature and stable, other parts (database, monitoring) have existing implementations but need reworking or major additions, others (calibration) are only recently under development

Context

- Eric reminded us yesterday of promises in the areas of ROD crate DAQ, user friendliness and documentation
- He omitted databases which were and still are the most critical area

TDAQ updates

New versions

- Shortly after our July meeting we upgraded to SLC3 and a new TDAQ software release (tdaq-01-02-00)
- Another new TDAQ version (tdaq-01-04-01) was released recently and we are in the middle of moving to that
- These TDAQ releases include the new official monitoring tools (GNAM, presenter) and significant improvements in ROD crate DAQ
- We had intended to be using these by now, but that has not yet happened – though ROD crate DAQ is nearly there

Summary of recent developments (1)

9U ROD

- Module services and simulation now all done
- Minimal database support using existing OKS framework

Bytestream decoding

- Bytestream decoding (of raw event data into more usable C++ objects) is now complete for most of the new 9U ROD formats

Run control scripting

- Simple minded scripting facility developed to automate control of the run control with various parameter settings
- Addresses issue of user friendliness – at least for a few well defined procedures. Though detailed control of a complex system will still need expertise

Summary of recent developments (2)

PPM calibration

- Procedures and analysis code for testing of input analogue signals is now very smooth – using VME readout of the PPM and flat file “database”

CMM calibration

- Demonstration of complete chain of CMM backplane timing calibration procedure to storage of new calibration data into COOL

CPM production acceptance tests

- Existing software wrapped up in scripts to automate the acceptance testing of production CPMs

Summary of recent developments (3)

ROD crate DAQ

- We are already using some of ROD crate DAQ, but not all...
- RCD based run controller is implemented and successfully tested at RAL
- RCD based module readout is implemented and unsuccessfully tested at RAL
- Once the latter is working, we can replace existing run controller and crate_readout program with RCD versions
- We still need to repack some of our standalone programs (“kickers”) so we can start them selectively (in the standard, but less flexible way)

Monitoring

- See previous talk...

Database (1)

Where we have come from

- We use and have extended the standard TDAQ configuration database (OKS) for almost everything: configuration parameters, calibration, trigger menu, run types, cables
- The only exception is the PPM whose configuration and calibration data are combined in a custom XML file (one per PPM)
- We have known of a long time that much of this should move into the final ATLAS conditions database – for a long time this was uncertain, but now COOL is well defined and in use by many detectors
- The top level COOL API is implemented on three different underlying database products: Oracle, MySQL and SQLite
- To allow us to continue running all our software at all our sites we decided to start by using the MySQL implementation at RAL
- Unfortunately this had a lot of problems and progress has been delayed

Database (2)

Where we are now

- We can store CMM backplane timings for one channel in COOL...
- ...and retrieve it for comparison with previous history...
- ...but not yet load latest values from COOL to CMM via run control
- We have started discussing how to organise our data in COOL...
- ...but are not yet sure of all the criteria for deciding this

Database (3)

Where we need to get to

- Calibration data stored and retrieved from COOL
- Store production and installation test results and histograms: probably keep large files separately with references from COOL
- Consider moving (some?) module configuration data from OKS to COOL (unless new history mechanism in OKS is sufficient)
- Run types for standard and multistep runs extended and combined with calorimeters – probably also in COOL
- Trigger menu from new trigger configuration database
- Final cabling (and crate/module?) configuration derived from TC installation database

Database (4)

New Online Database Taskforce

- new taskforce to “clarify and focus work in the database group and subdetectors communities relating to online database work for the ATLAS commissioning phase and beyond”
- <https://uimon.cern.ch/twiki/bin/view/Atlas/OnlineDBTaskForce>
- chaired by Steinar Stapnes, reps from commissioning, DB, DAQ, DCS
- series of meetings with subdetectors (ID, LAr, Tile, Muons, Trigger)
- systems represented by online/DB experts and system leaders if possible
- we will be invited to come prepared with written list of conditions and configuration data together with sources, destinations and volumes of data – circulated in advance
- report to 28 Nov commissioning meeting and 1-2 Dec TMB meeting
- considering a workshop in January

Next steps

- Update all sites to latest TDAQ release
- Complete move to RCD run controllers and module readout
- Further developments of production acceptance test software
- Move PPM cable testing software to work with ROD readout
- Start (re)developing monitoring software using new tools
- Database, database, database...

Summary

- Useful progress in many areas...
- ...but slow progress in critical database area
- Still need to fully move to ROD crate DAQ run control
- Still need to move to new monitoring tools
- Still need to document the system better: try using Twiki?