Level 1 Use of Online Software:

Present Situation; Requirements; Assessment of Online Framework and Tools; Suggestions for Next Steps

Murrough Landon – 14 November 2000

Outline

- Introduction
- Existing Software
- Use Cases
- Requirements
- Assessment of Online Software
- Future Plans

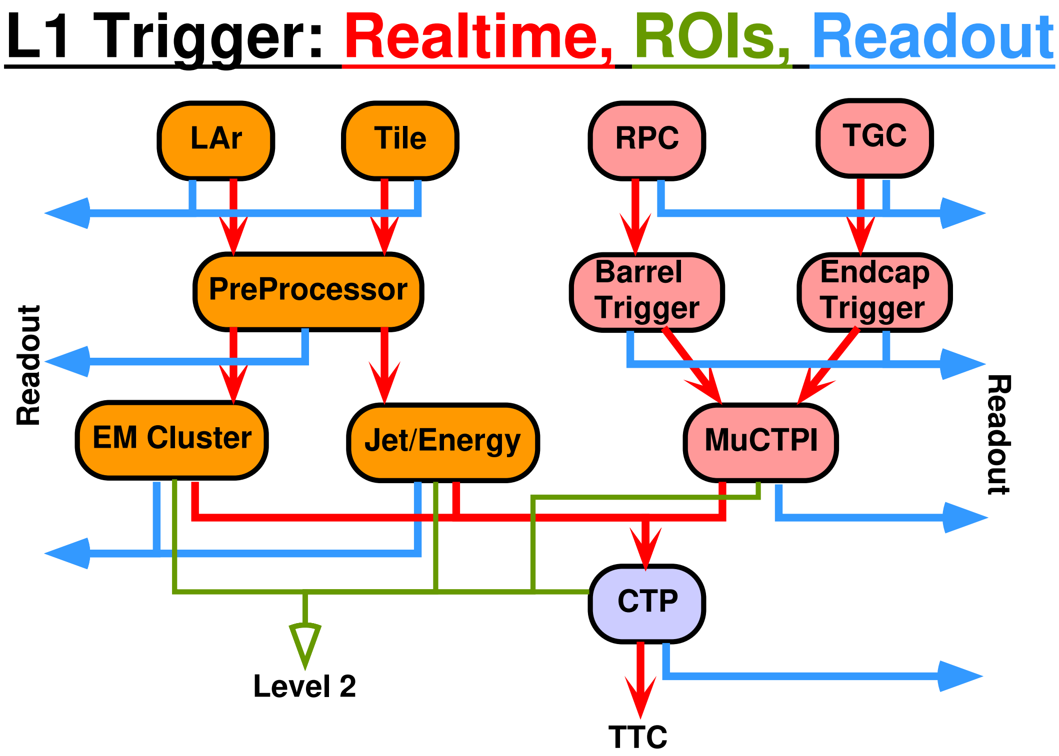
Caveats

- Rapid collation of material...
- Talk slanted towards Calorimeter Trigger

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

Introduction (1)

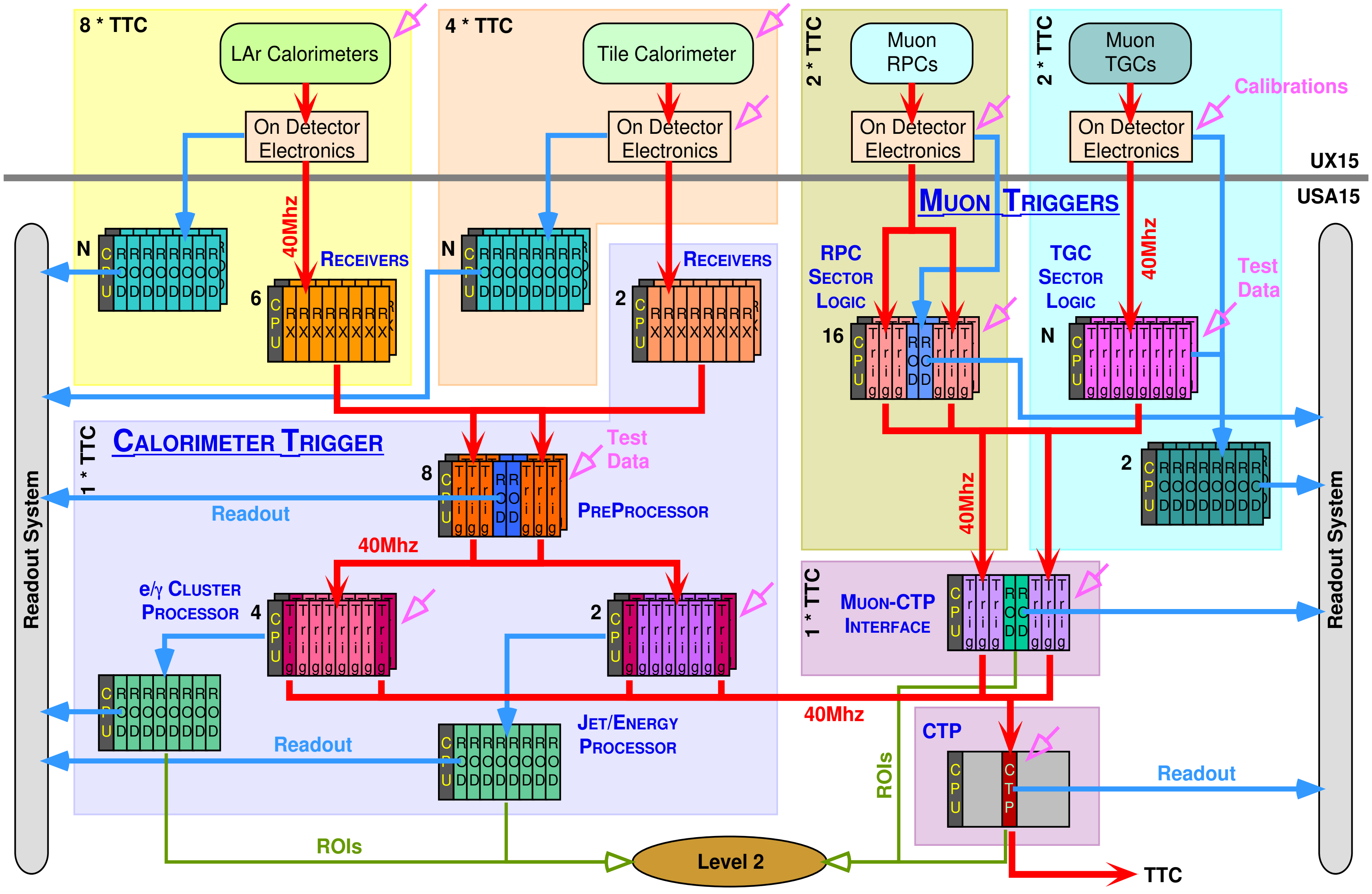
Overview of Level 1 Trigger



- Two separate systems: Calorimeter and Muon Triggers both feeding CTP and thence TTC.
- Both triggers are multistage systems: data sampled and read out at several points along the level 1 trigger decision path.
- Calibration pulses and digital test vectors can be injected at inputs to many of these stages
- Verification of realtime processing of a single channel of one event requires collection of data from several ROD crates (in both Calo and Muon systems).
- Many different subsystems: crates with RODs only, crates with trigger processor modules only, mixed crates.
- Large number of different module types, each with separate sets of configuration parameters.

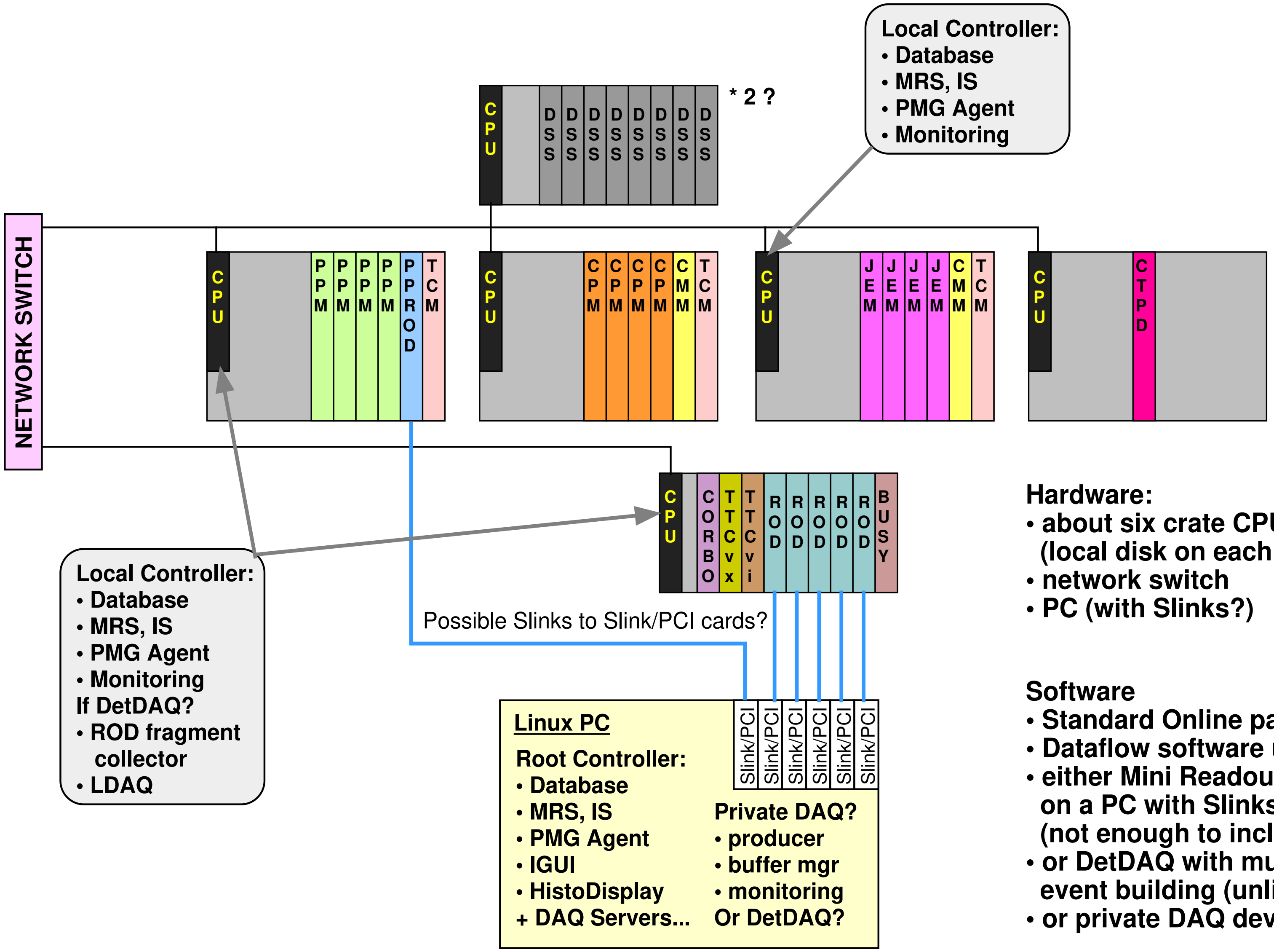
L1 Trigger: Realtime, ROIs, Readout

Murrough Landon
8 November 2000



L1 Calo Trigger: Slice Test DAQ Setup

Murrough Landon
31 October 2000



Hardware:
- about six crate CPUs (local disk on each?)
- network switch
- PC (with Slinks?)

Software
- Standard Online packages
- Dataflow software uncertain:
- either Mini Readout Crate on a PC with Slinks (not enough to include CTP)
- or DetDAQ with multicrate event building (unlikely?)
- or private DAQ development

Introduction (2)

History and Timescales

- Long series of hardware demonstrators
- Level 1 produced its TDR in June 1998
- Now mostly in “fully specified prototype” phase (ie different phase from DAQ/HLT)
- Test new subsystems separately up to mid 2001
- Level 1 complete slice(s) starting mid/late 2001
- Priority: level 1 trigger decision path
- **On successful completion** of at least a minimal set of internal slice tests, join DAQ/HLT integrated slice mid 2002
- Though some ROD-RoIB and ROD-ROS tests early 2001
- Also need integrated tests with calorimeters and muon chambers

Present Situation

Fresh start with new software

- Lab and beam tests leading up to the TDR were performed with a mix of private software developments using a variety of packages.
- Aim for the future: develop **new software** based around the DAQ -1 Online (and Dataflow) components.
- Compare with Level 2 and DCS: no significant replacement or integration of equivalent packages is required.

Existing software

- Calo Trigger: parts of test and diagnostic software toolkit (HDMC) will be used. This provides low level descriptions of each type of module. May be useful to integrate with higher level DAQ configuration database (ie description at system of crates level).
- MuCTPI and CTP: low level software packages also exist. Configuration data is stored in a variety of files.

Worry

- Online components appear to be a good basis for our needs. . .
- But in the dataflow area we need a small scale solution which can collect events from multiple ROD crates.

Use Cases: Summary (1)

NB partial set of use cases from initial analysis. More detailed studies for the whole of level 1 are required.

System Management

- Install N systems, libraries, etc
- Synchronize upgrades on N systems

Startup

- Cold (power on) reset
- Warm reset
- Start normal run
- Stop normal run
- Shutdown

Configuration

- Change configuration database
- Load different (existing) trigger settings
- Generate new settings from new trigger menu

Use Cases: Summary (2)

Calibration

- Wide variety of calibrations to be performed:
- Calo: energy, timing (calo cells and trigger sums), pulse shape (BCID), internal timing
- Muons: HV, ASD thresholds, timing delays, absolute timing, momentum
- Calo trigger calibrations require use of calorimeter calibration systems. Some runs can be driven by calibration data; others need synchronised scan of trigger parameters during calibration run.

Diagnostics

- Check systems by loading test vectors or calibration pulses at the detector and verifying correct processing and final trigger by downstream logic.
- Individual checks of each stage by loading test patterns separately at the inputs of each stage and verifying outputs.
- Individual checks of all links by loading test patterns separately at the outputs of each stage and verifying inputs at the next stage.

Others

- Various other special procedures...

Use Case: Start Run

Activities for Normal Run (load,config,start)

- Ensure database files are up to date; read hardware and firmware configuration databases
- Initialise/check DCS connection (if required)
- Check expected hardware configuration is present; lock hardware resources
- Check firmware (FPGA programs) report correct version; load new FPGA programs if required
- Reset all modules
- If RODs present, ensure BUSY is set; configure BUSY network
- Read all calibration database files for each crate; load calibrations and other settings into modules (maybe only reload if changed since last time)
- Read trigger menu database files for each crate; load trigger settings and FPGA programs into modules (maybe only reload if changed since last time)
- Report/log which calibrations and trigger settings used for this run
- Check all input links are up
- Start event-independent monitoring threads/processes
- Book/zero histograms and counters
- Publish crate and module status information
- Configure TTCvi
- If RODs present, remove BUSY

Use Case: End Run

Activities for End of Run

- If RODs present, ensure BUSY is set
- Stop(?) event-independent monitoring threads/processes (but some monitoring continues until shutdown)
- Save histograms and statistics for that run

Use Case: Calibration Run

Many Variations on a Theme

- Setup as for normal run
- Initialise calibration system and/or load test vectors
- Sequence of run; pause; load new parameters; resume
  NB pause of calibration triggers and/or run controllers in multiple crates
- Monitor and analyse data in ROD firmware or local CPU
- Save new calibration or other settings to database
- Ensure local calibrations updated in offline conditions database

Requirements (1)

General

- Clear need for all the standard Online packages: RC, MRS, IS, Database, PMG, RM, OBK, . . .
- Also require additional packages: DDC, . . .
- Further development of existing packages

Database

- Calibration data partitioned by crate
- Lot of work needed on the schema:
  - many types of module to describe
  - links to detector description
  - trigger menu (level 1, 2, EF)
  - software schema extended to FPGA programs
  - TTC partitions?
- DALs(?) and (user friendly) tools
- Boundary of responsibility: DAQ configuration vs subdetector specific schemas (eg calibration data)?

Requirements (2)

Run Control

- Local controllers in both ROD and non-ROD crates; overall controllers for subsystems?
- Need to cope with many different calibration regimes (more work on use cases is required here)
- Multithreaded: perform local crate monitoring during the appropriate states?
- Muon trigger: DCS connection essential for some functions

Monitoring

- Will have both event monitoring (using monitoring skeleton) and continuous monitoring of rates, errors conditions etc.
- Some monitoring continues after run stop (ie until shutdown of the partition).
- Start and stop some monitoring tasks at any point in a run (debugging, development, tracing problems).
- Monitoring of trigger functionality requires event building across ROD/readout crates.

Requirements (3)

DetDAQ

- Common hardware/software environment at the ROD level
- Various low level libraries or skeletons for:
- VME, PCI, Slink
- Collection of fragments from RODs (ie hooks to install ROD specific code)
- Event formatting, buffering
- Scheduling different tasks
- Communication with local controller
- Retrieval of configuration data from database
- Facilities for monitoring of events
- . . . and of course (low speed) event building across crates

Requirements of DetDAQ on the RODs include:

- As much commonality among types of RODs as possible
- Facilities in the RODs to capture fragments from the same event in different RODs

Assessment of Online Software (1)

Overall

- Existing (and planned) software looks like fulfilling our requirements
- Documentation very extensive - some variability between packages
- Good support and training
- Progress towards fully open source project is very welcome
- Helpful interactions with Online software group
- We have given feedback on all core and some other components
- We have been involved with specifiying DAQ-DCS connection

Assessment of Online Software (2)

Miscellaneous issues and suggestions

- Partitioning: (not) running subpartitions within a single TTC partition?
- Interactions between RC and local triggers for calibrations
- Database schema: organisation and support
- Storing local DAQ configuration/calibration data into offline conditions database
- More user friendly editing tools (XML is promising)
- Version control/logging of changes: log changes made to database files by trigger “experts”? Eg CVS connection from OKS editors?
- Support for distributed histogramming (Calo trigger currently intending to use a ROOT based package from CDF)
- Extensibility of Event Dump to decode subdetector specific ROD fragments
- Utility Event class (C++) to find/iterate over event fragments
- Not all Online component APIs are yet available in Java. . .

Work Done So Far

Database

- Complete(?) list of all parameters required to configure the calorimeter trigger have been documented in ATL-DA-EP-0002.
- A similar list is available for the CTP (ATL-DA-EP-0001) and the MuCTPI: http://home.cern.ch/schwick/muctpi/MuctpiDataBaseItems.html.
- The categories of data to fully describe the trigger menu have been documented in ATL-DA-ER-0002.
- None of the above yet exist as object models though.

Run Control and Calibration

- Calibration procedures are known in some detail
- But not yet all mapped to DAQ/RC states
- Calo trigger: initial discussions have been held with calorimeter representatives to check calibration systems can do what we need (but no detailed use cases worked through yet)

Other

- Started developing suite of test vector generating programs (far from complete)
- Low level code for some prototype modules

Future Plans (1)

New Software: Online

Put flesh on Online skeletons and develop subsystem specific software. . .

- Develop initial work on trigger parameters into object model schema for the database
- Start to implement local controller(s) for the various crates
- Code to initialise and configure different module types
- Firmware (and firmware generation utilities) for programmable modules
- Full set of programs to provide test vectors for diagnostics (agreement on formats across subsystem boundaries)
- Continuous (event independent) monitoring tasks/threads
- Event monitoring tasks: monitoring skeleton or private framework – depending on dataflow software decisions
- Calibration and setup procedures
- Online simulation of the trigger (overlap with offline?)

Timescales

- Minimal set of software required by mid 2001
- Continue to develop and refine up to mid 2002
- Evaluate
- Develop final software for ATLAS

Future Plans (2)

Dataflow

- Tests of single subsystems can use DetDAQ (when available)
- . . . requires implementation of ROD fragment collector and ROD-specific code
- BUT slice tests need multi-ROD crate readout for some checks:
- PC readout crate may not have enough Slink/PCI cards (complete L1Calo slice test involves $\geq$ seven RODs)
- DetDAQ may not provide event building across crates. . . (given ROD fragment collector, would still need other changes)
- May need to continue private (Calo Trigger) DAQ solution (embedded in Online software)

Problem

- Limited effort available. . .

Summary

Main points

- Online software seems to meet our needs
- Some extensions suggested
- Much work needed to implement contents of the Online frameworks. . .
- . . . detailed studies of use cases
- . . . database schema
- Also work required in Dataflow area...

Discussion?

- Are there any cases Online software fails to cover?
  "DAQ world" activities independent of partitions?
  Missing states/commands in RC skeleton?
  Missing/desirable additional Online components?
- Major missing DAQ component is presently DetDAQ
  Trigger really needs multi-crate readout.
- Commonality across independent level 1 subsystems?