Schedule, Database and Run Control Status

Murrough Landon – 14 March 2002

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

Overview

- Schedule
- Database
- Run Control
- IGUI Panels
- Documentation
- CMT Status
- Online Group News
- HDMC Issues

Schedule

- JEM0 under test for some time, CMM and CPM basic tests just starting
- Real software is needed really soon now!
- Initial interactive tests with HDMC using already prepared parts files (for CMM, CPM at least)
- We must establish a realistic timescale for full DAQ type software

Timescale

Optimistic estimates?

- Database work: 2-4 weeks
- Test vector scheme: $>$2 weeks?
- Module services for CPM, CMM: 2-4 weeks each?
- Simulation of CMM: $>$2 weeks?
- Various integration tests: 2-4 weeks
- System management: 1 week? [Now done?]
- Not likely to be ready before May??

Database: Trigger Menu

- Trigger menu database defined about 12 months ago
- Work on a GUI trigger editor abandoned in expectation of a complete offering from the CTP group
- Existing classes are used in the offline simulation
- Would like to keep them the same – probably via an interface to the CTP developments?

Database: Hardware Configuration

- Online software hardware configuration schema has been extended
- Module subclasses defined for all our modules
- No extra attributes (yet) but used to create correct run time classes
- Schema defined for the firmware configuration of each type of module with versions of firmware to be loaded into each device.
- Q: is one configuration per **type** of module OK, or should we specify for each individual module?
- Schema defined for describing the cabling between modules (including provision for “octopus cables”) and bundles of cables between pairs of modules (eg LVDS connections)
- Already used satisfactorily in setting up the simulation

Database: Run Types

- Initial attempt to define schema to describe run types
- Provision for global run type dependent settings
- Also run type dependent settings for each bf type of module
- This scheme may need extension/modification for complete description of tests and test vectors

Database: IS Information

- Schema defined for several IS data structures: general run parameters, per module parameters
- Needs extension to include module status
- Online software can automatically generate C++ and Java code from the database schema
- Java code is already used in new IGUI panels
- C++ code will be included in complete database API...

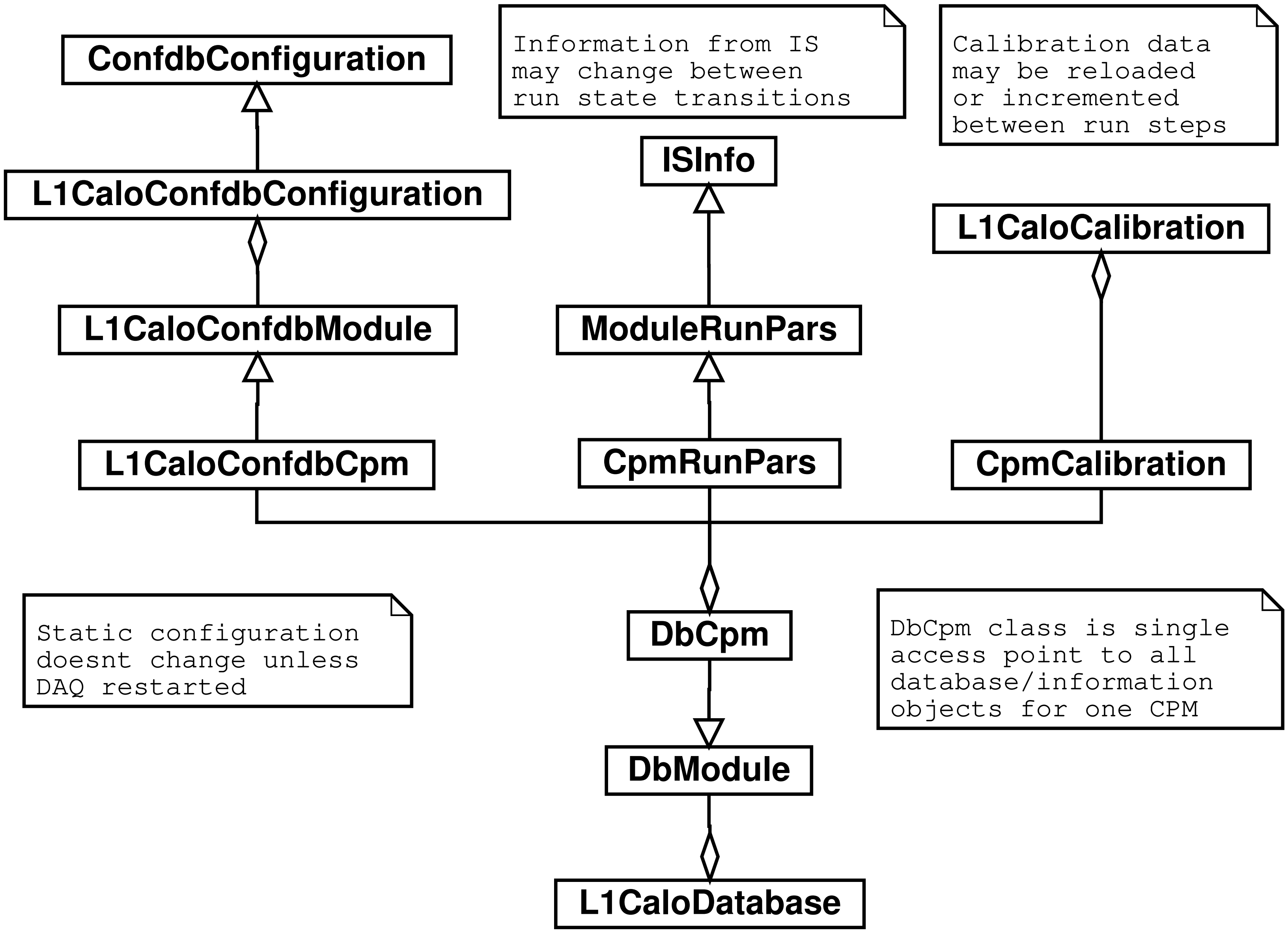
Database: Calibration Data

- Long standing schema defines some calibration data for CPM
- Needs update to newer Online database style
- Also needs completion of CPM data and extension to other modules
- Most modules have little calibration data – only the PPM is a serious challenge

Database: Putting it all together

- Module services package would like:
  - One database object per module, giving access to
  - One database object per submodule
  - These objects should provide hardware and firmware configuration data, calibration data, run parameters, etc
- That information will actually come from a variety of different objects
- Need to provide a thin facade collecting them all together
- NB although configuration information will be constant, calibration data and run parameters may change, eg during multistep test or calibration run
- Module services needs to be careful about when it can rely on which sort of information

Combined Database Information for CPM



Run Control

- Little work done on run controllers for some time
- Updates for incremental changes in database
- Separate database handling into standalone (reusable) class
- Modifications to integrate with module services
- Sequencer program to drive multistep tests and calibrations needs development (presently just a proof of principle)
- “DSS kicker” program to drive DSS during tests needs to be moved into run control/database framework.

IGUI Panels

- The Online software Integrated GUI (IGUI) allows for extension by subdetectors
- L1Calo panels added for general run parameters (eg for sequencer program), individual module parameters, skeleton for future module status
- Uses IS variables generated from the IS database schema
- Java code uses the “reflection API” to (mostly) automatically generate panels using self description information in the schema so it should be quick to add new attributes to existing IS variables
- Also fairly quick to add new panels for new IS variables

Documentation

- Simulation package is now well documented. Reference documentation generated via thorough use of Doxygen. Separate “hand written” user guide with examples.
- Module services package following soon?
- Earlier draft versions of run control and database ideas now need to be updated and completed.
- L1Calo website: many pages updated. Eg packages pages, HowTos for CMT, Online software, descriptions of scripts in the scripts package etc.

CMT Status

- CMT now being used for several L1Calo packages: dbFiles, confL1Calo, rcL1Calo, isL1Calo, iguiL1Calo. Also control packages L1CaloPolicy, L1CaloRelease and HdmcExternal to define HDMC as an external package.
- Follows example of Online group working model. Some scripts adopted from the DataCollection group.
- Follows example of Online group working model.
- All works fine so far.
- But still have to work out how to write CMT style make fragments to cope with Qt. This is required for HDMC.
- Move other packages (simulation, testVectors, moduleServices)
- Later also HDMC?

News from the Online group

- Current release is 0.0.16.
- Additional functionality in `is_generator` provided for me in a special patch (will be in the next release).
- Next release will be with CMT (also SRT).
- They are reviewing all their user requirements documents. These are mostly very short and snappy – quick to write similar things for our packages?!
- Any volunteers to be involved in their reviews? (I have been asked to take part in review of two components).
- An Online task force to discuss the requirements for the conditions database had a CERN based meeting last week.

HDMC Issues

- Perennial question of HDMC library organisation. No hurry?
- But sensible to move module services (L1CalDaq directory) to separate package for easier linking with Online software.
- Useful common parts: TTCrx, FPGA loading, ...
- Is A32 required in the PP crates? VME64x??