

# Software Discussion Items

---

Murrough Landon – 7 November 2002

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

## Overview

- Issues from the release
- Organisational improvements
- Additional functionality
- Firmware organisation
- Small projects for extra people?

Many of the following slides have suggestions for timescales, priorities and who might do the work. These are my **tentative** suggestions for discussion.

# Issues from the release

---

## The process itself

- How useful was it?
- Did it divert us from other more pressing things?
- How long did it take (apart from the work we needed to do anyway).  
ML estimate: about a week of my time (fiddling with scripts, building and rebuilding, writing release notes and README, a little testing)  
and (guess?) a few days of Bruces time (more serious testing).  
Improvements to scripts may be considered as an investment, testing will always take time each release.

## Other issues

- Should it be public? Is linked from our website?
- Do we need to worry about licensing, copyright, disclaimer, (no) warranties etc?

# Organisational improvements (1)

---

- We need to rethink the test vector directories and files. The test descriptor, or "Bill" files (\*.in) are really source files. They should be stored in CVS and installed into wherever we want to read them. This need not be the same directory we use for generating and simulating test vector files.  
Should the test descriptor files just go into the database package (dbFiles) or a separate one? Where should we install them?

**Time: discussion + day to reorganise?**

**Priority: medium? — Who: ML/SJH???**

## Organisational improvements (2)

---

- The play\_daq script reads play\_daq\_userdefs and play\_daq\_userinit scripts which we keep in ~/l1calo/etc. These should be included in the release too. (The userdefs file is due to be replaced by a ~/.onlinerc file in the next Online release, but we will still want to provide a default one with our release).

**Time: discussion + few hours to sort out properly?**

**Priority: low? — Who: ML?**

- We should move HDMC into the CMT environment (as several packages).

Q: how should we handle VME drivers?

**Time: discussion + few days probably?**

**Priority: medium? — Who: ML(+BB?)**

- Remove Qt dependence from HDMC hardware parts.

**Time: discussion + another few days probably?**

**Priority: medium? — Who: ML(+BB?)**

## Organisational improvements (3)

---

- Several of the module services packages create a private "myHdmc" application adding their module classes to the standard hdmc.

We should probably have a separate package (allModules?) which would use all our other module services packages and make a single myHdmc which has all our modules.

This could also be the place to enable calls to other DaqInterface methods from (an extended?) GUI.

**Time: discussion + day? + future extensions...**

**Priority: medium? — Who: BB?**

- Implement "check targets" for packages where appropriate. NB simulation needs area for large data files: CERN AFS?

**Time: discussion + hours per package?**

**Priority: medium/low? — Who: All?**

## Organisational improvements (4)

---

- We have a cpRodTests package for the programs specific to the CPROD tests, separate from the CPROD code itself. While tests programs that **only** access a single module can naturally live in the package for that module, I think that standalone test programs that access several different module types should be in a separate package to avoid too many crossing "use" relationships.

For example the cpmServices package contains a test program for testing the CPM which has grown and now also uses TTCvi, DSS etc. I think this now belongs in a separate package. (Though a program which only needs a CPM is fine within that package I think).

Im not sure if we want a package per module (eg cpmTests) or per subsystem (eg cpSubsystemTests) or for all standalone tests (eg subsystemTests).

**Time: discussion + day to reorganise?**

**Priority: medium/low? — Who: GM?**

## Additional functionality (1)

---

The general assumption is that the next release, or at least next step in “evolutionary delivery” is centred around tests of the CP subsystem. Including the JEP subsystem may perhaps come at the same time or soon after.

- The cpmServices need to be completed. This (I think) comprises some extra functionality in setting up the CPM(?) and also integration with the database, following recommendations for what to do in each state transition and inclusion in the run control.

**Time: month?**

**Priority: high — Who: GM(+ML?+BB??)**

- The cmmServices need to be completed. This also still needs some extra functionality in setting up the CMM(?). Possibly some little standalone test program? Maybe some more database work?

**Time: month?**

**Priority: high — Who: NG(+ML?+BB??)**

## Additional functionality (2)

---

- The cmmSim package needs to be completed. Also test vectors need to be generated.

**Time: month?**

**Priority: high — Who: NG(+SJH??)**

- Some framework packages could do with a little “sanisation” (packages: testVectors, cpRodTests and others perhaps).

**Time: discussion + days?**

**Priority: medium — Who: BB/SJH+??**

- The ttcviServices need a (temporary) fix to allow setting all TTCrx registers via the TTC instead of via VME and I2C bus.

**Time: day or two?**

**Priority: high — Who: GM/BB/ML??**

## Additional functionality (3)

---

- We should be able to load firmware into the modules following a configuration described in the database. [See later].

**Time: discussion (lots!) + week?**

**Priority: medium/high? — Who: ML/GM/BB??**

- We need test descriptors for larger scale tests.

**Time: week?**

**Priority: high — Who: ???**

- The issue of conflicting settings between the test descriptor files and the database needs to be resolved. This needs further discussion.

**Time: discussion (lots!) + week?**

**Priority: medium?? — Who: ML/BB**

## Additional functionality (4)

---

- The interaction of the run control with module services also needs another look.  
**Time: discussion + day or two??**  
**Priority: medium? — Who: ML/BB**
- The DSS needs to handle LVDS and GIO cards and to cope with many different combinations in a nice way. This may be linked with the above point.  
**Time: discussion + week?**  
**Priority: high — Who: BB(+GM?)**
- Once this is done, we should start using the L1A generation scheme.  
**Time: ???**  
**Priority: high?? — Who: BB/SJH?**

## Additional functionality (5)

---

- We should try to start using the ROS.

**Time: week or two??**

**Priority: medium? — Who: BB+??**

- We should try to monitor events via the standard monitoring framework.

**Time: week?**

**Priority: medium — Who: maybe ML or SJH??**

- We will need to develop procedures for setting up the timings.

**Time: weeks?**

**Priority: high?? — Who: ???**

- We should be able to make multistep runs with sequences of test vectors being loaded.

**Time: week?**

**Priority: medium?? — Who: ML/SJH?**

## Additional functionality (6) – JEP

---

- Existing JEM test software needs to be integrated with the moduleServices framework, to create a jemServices package. Further development may also be required?

**Time: weeks?**

**Priority: medium? — Who: ???**

- The JEM simulation also needs to be integrated with the updated simulation framework. The simulation may also need further development, eg implementation of event readout, integration with database, etc.

**Time: months?**

**Priority: medium? — Who: ???**

- Physics-like test vectors exist for the JEM (I think). Probably other test vectors need to be generated to stress the system and check boundaries etc.

**Time: weeks?**

**Priority: medium? — Who: ???**

## Additional functionality (7) – JEP/PP

---

- Software (and firmware) for JEP variants of CMM and CPROD are required (I think).

**Time: weeks?**

**Priority: medium? — Who: ???**

- Module services, simulation and perhaps more test vectors are also required for the Preprocessor system.

**Time: several months?**

**Priority: medium? — Who: ??? — When: ???**

## Miscellaneous issues

---

- Next Online release (imminent): some improvements and now fully CMT based.  
Check use of gcc 2.95.2 compiler
- CERN will move to RedHat 7.3...
- ...Qt3 (someday): small, but coordinated HDMC migration needed
- gcc 3.1 (someday): proper use of std namespace required

# Documentation!

---

- User guide for the database. [Draft available! (Two long days)]  
**Time: week?**  
**Priority: high? — Who: ML**
- User guide for moduleServices.  
**Time: week?**  
**Priority: high — Who: BB**
- Update simulation guide? (If changes to the framework are sufficient to require this).  
**Time: days?**  
**Priority: medium?? — Who: SJH**
- Update run controller document.  
**Time: day?**  
**Priority: medium?? — Who: ML**

## Firmware discussion: requirements

---

- The database shall describe the firmware to be loaded into each module.
- The description of the firmware shall specify the files to be loaded and contain other information to allow the file to be validated before being loaded.
- It shall be possible to have different firmware descriptions for different modules of the same type.
- It should be possible to have a single description of the default set of firmware to be loaded into all modules of a given type.

## Firmware discussion: use cases

---

- Check that all active firmware programs currently running in the modules FPGAs up to date.
- Load module flash RAM with latest firmware versions.
- Load module FPGAs from flash RAM, including loading of selected alternate versions (if module allows).
- Load module FPGAs directly via VME (if module allows).

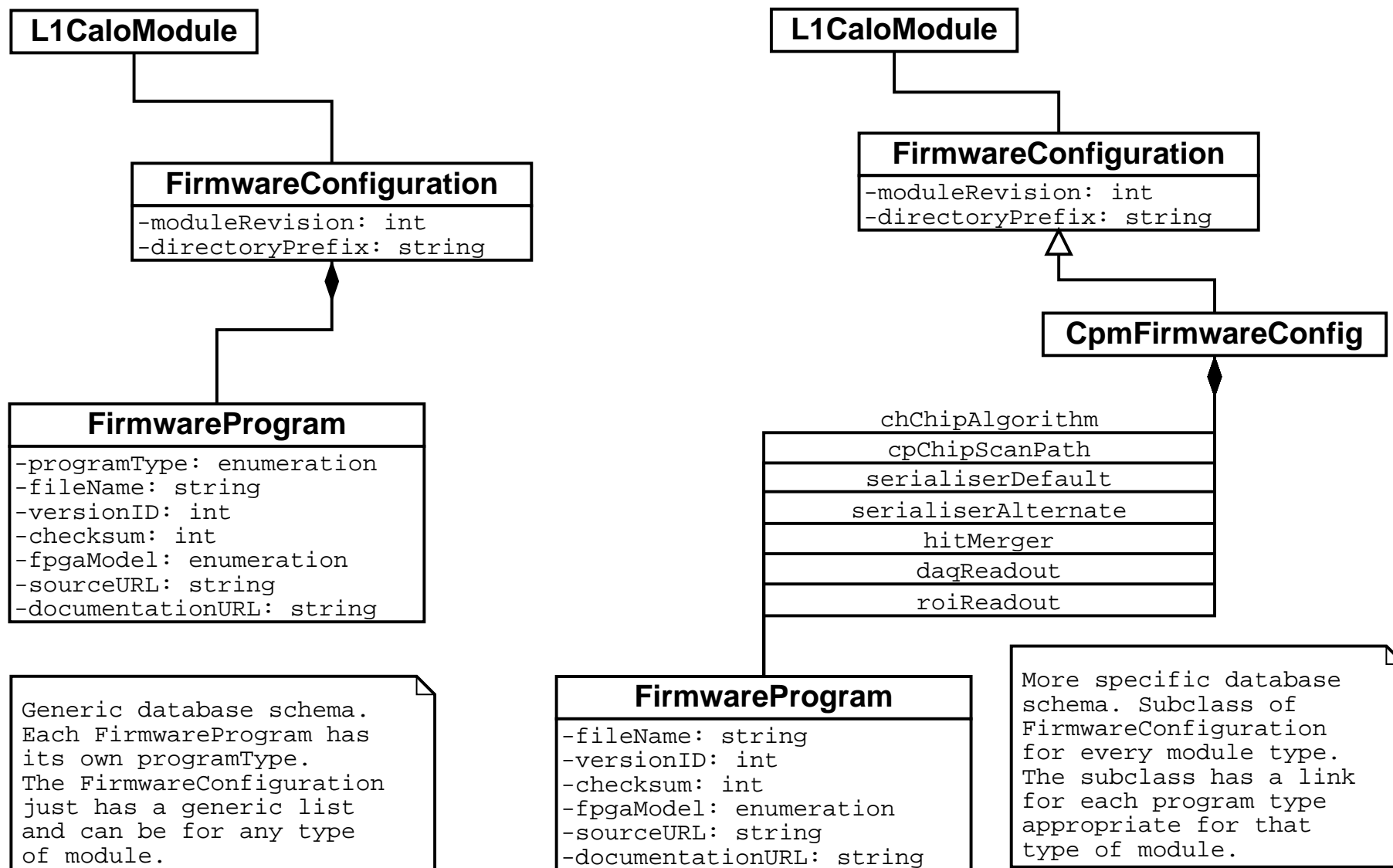
*Do we need different terms for loading flash RAMs vs loading FPGAs from flash RAM?*

# Firmware discussion: initial proposal

---

- Two suggestions for a schema: see diagram... (first draft 14 months ago!)
- Each module object may have a link to a FirmwareConfiguration object.
- The FirmwareConfiguration class describes collection of firmware relevant to one type of module. Normally there is just one FirmwareConfiguration object per module type. Can have different FirmwareConfiguration objects for different hardware revisions or for testing different firmware versions on a test module.
- FirmwareConfiguration contains a list of FirmwareProgram objects.  
FirmwareConfiguration also has attribute(s) to identify the hardware revision of the module to which it applies.
- FirmwareProgram class describes one firmware variant to be loaded into one type of FPGA (Serialiser, CpChip, etc) of a specific brand (eg XCV1000E) on one type of module.
- FirmwareProgram objects have attributes for the filename of the binary to be loaded, its version, checksum, etc.

# Firmware schema: two suggestions



# Small Projects?

---

In case we get some part time people to help on the software...

- Extend CPROD simulation to different firmware variants. [Done?]
- Ditto for the CMM?
- Add simulation of the ROS (ie event building) eg using the datacollection eformat package?
- Customised event dump.
- Event display??
- Monitoring of events and filling histograms?
- DCS: something with PVSS2??