Various Software Topics

Murrough Landon – 28 November 2001

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

Overview

- Recent work on run control
- Run types and run parameters
- Test vector organisation
- Software packages and CMT
- Schedule...

Run Control (1)

Recent Non Work

- Not much since the joint meeting (where I reported adding use of the parts manager to create hardware modules)
- The above changes are still not in CVS (things are still in flux a bit and I want to tag the first try).
- Use of "`#include hdmc/Dir/file.h`" requires extra link in HDMC directory tree.

Run Control (2)

Calibration Sequencer

- But I have successfully tested a “calibration/test sequencer”.
- This is adapted from an Online software example (part of their internal test suite).
- Sends a sequence of pause/resume commands to the root run controller, ending with a stop command.
- The idea is that the other controllers can take the relevant actions for the next step in the calibration or test run.
- So far only a proof of principle: it just has hard coded waits for each step. What do we actually want? Eg software L1A commands to TTCvi? Control of other external logic?
- So far run separately by hand. Should be started automatically for specific run types by a suitable run controller.
- Should also be controllable via run parameters (stored in IS, set via IGUI).

Run Types and Run Parameters

Run Types

- Physics (and maybe video RAM)
- Calibration and test sequences
- Should various tests, eg loading playback in different places (PPM, CPM, etc) be separate run types?
- Or a single playback run type with sophisticated test vector choice as a run parameter?

Run Parameters

- Not too complicated (GUI code is tedious!)
- Enable/disable modules described in full database?
- Some global modes for modules (eg BCmux)?
- Calibration/test sequence control (n.steps, trigger method, time or n.triggers per step, etc)

Software Packages and CMT

How to organise our packages

- ATLAS has finally decided to adopt CMT. Online group is now trying it seriously.
- I intend to try it seriously (but have not done so yet).
- We do need some conventions for interworking between different packages. CMT (and ATLAS) impose some choices.
- Eg "`#include package/header.h`" style. This implies suitable directory structure (not yet done for my packages).
- CMT also expects consist use of CVS tags. ATLAS suggest `pkg-vv-rr-pp`, which `vv` the major version, `rr` the release, `pp` a patch.
- Online is developing a working model – but dont yet have guidelines for the librarian or how to make a binary release:
  `http://akazarov.home.cern.ch/akazarov/cmt`
- CMT home page (manual and tutorials) are at:
  `http://www.lal.in2p3.fr/SI/CMT/CMT.htm`

Schedule, Workplan, Timescales

Gantt chart

- After many repeated suggestions, I have now tried to enter information about our tasklist and timescale into a tool which produces Gantt charts. (Mac shareware to start with which was quick and easy to get hold of, but isnt very sophisticated).
- Should probably use eg MS Project (does that ensure workers dont work more than 100% of their available time?)
- Preliminary chart is doubtless very unrealistic. It also has many overlaps and gaps, etc.