Software Status

Murrough Landon – 7 March 2003

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

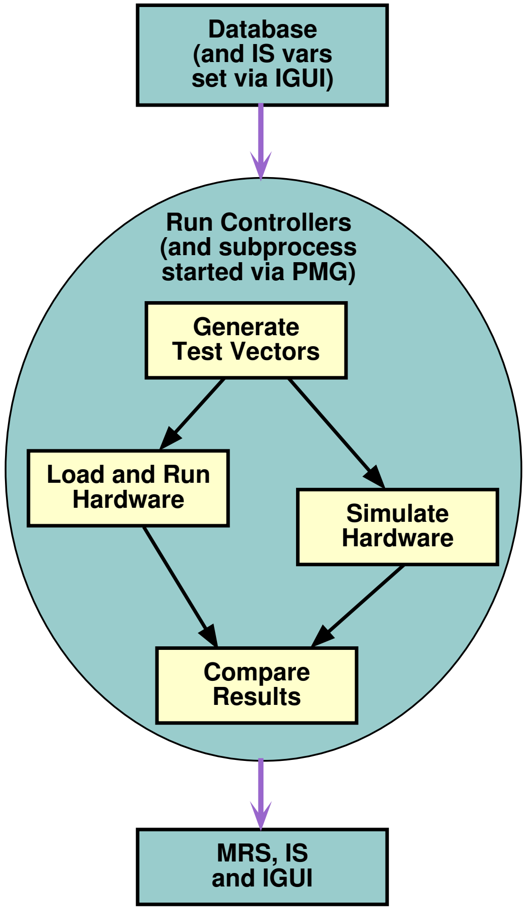
Overview

- Reminder of the aims
- Recent developments (whats there)
- Whats missing
- Overall status
- Progress, plans and people
- Supported test plans

Reminder of the Aims

Slice Test Procedure

- Choose a test to run and a hardware configuration
- Generate test vectors (if necessary)
- Load the hardware with test vectors
- Simulate expected output of the selected configuration
- Run the system, collect data, compare and report



Recent Developments: Overview

Since the first release

- The first release (October) was limited to CPROD/DSS tests at RAL
- Main focus since then has been adding other modules and (for the moment) their standalone test programs
- Aim has been to develop these standalone tests in keeping with the distributed run control and module services philosophy
- Considerable progress with software for CPM and JEM: module services, test vectors, simulation (CPM)
- CMM software development has been waiting for release of the first module from tests by RAL engineers – but should restart very soon
- Recent major update of module services API
- Also a lot of database related developments
- No new releases, but nightly build (including docs) available via the web

Recent Developments: Database (1)

Better support for run types

- Users (only the developers at the moment!) dont want to shutdown and restart the DAQ to change between run types
- Also dont want to have to change lots of parameters in different places in the IGUI at the same time
- So I have added run type objects to our database (Online DB just uses run type names)
- A run type brings sets of options for each module type (and may have special settings for individual modules) together with options for multistep sequences, the test description file (which controls the test vector generation) and possibly which modules expect to read/write test vectors
- The run type can be chosen from a list of available run types presented in the L1Calo panel of the IGUI

Recent Developments: Database (2)

Firmware

- The database can now contain descriptions of the firmware to be loaded into each type of module (or individual modules)
- So far this is not yet in use?

TTCvi

- Since we are not yet using the new TTCrx/TTCdec cards with I2C support, the database now allows the TTCvi to find the TTCrx settings (clock phase,delay) to be configured via TTC broadcast commands

Recent Developments: Database/Run Control

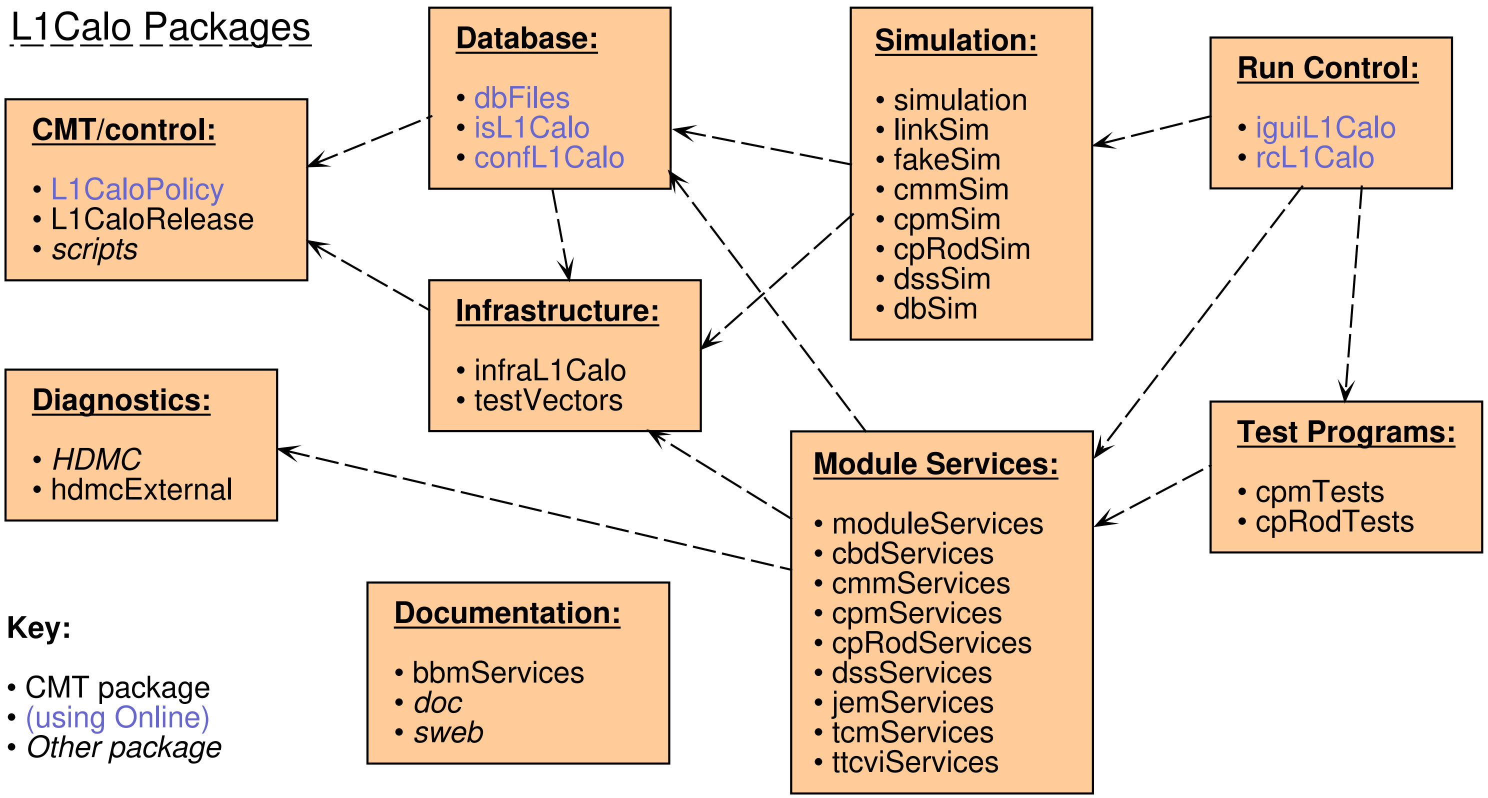
Better support for multistep runs

- Sequencer program drives run control through a set of pause/resume transition
- Calibration objects now support scans of single calibration parameter at a time (for various different parameters)
- A scheme load a succession of test vectors at pause/resume transitions still needs to be developed (what to do about generation and simulation in this case?)

Cleanup

- Some internal cleanup and reorganisations were done in the large database package as part of recent developments. More could be done in future, eg follow DataFlow approach and automatically generate some of the code

L1Calo Components and Packages



Whats missing? (1)

General framework

- *NB several items also appeared on this slide in Bham*
- Implementation of L1A generation (requires DSS GIO card)
- Use of the ROS and monitoring framework
- Inclusion of HDMC into CMT environment

CP subsystem

- *NB a lot of CPM work done since the Bham meeting*
- CPM module services: bit more use of DB (menu, firmware)
- *But little progress on the CMM – waiting for module*
- CMM module services (partly done)
- CMM simulation (partly done – needs updating?)

Whats missing? (2)

JEP subsystem

- *NB a lot of progress compared with this slide in Bham*
- JEM module services: some tidying up and better integration with run control?
- Test vectors: physics test vectors are now used, what about specially crafted ones to test boundaries, random data etc?
- *But the following items remain the same...*
- JEM simulation: someone has to take over student project and complete it – this is becoming critical
- CMM: jet and energy algorithms must be added to CP simulation
- CMM module services: little change from CP version?

Whats missing? (3)

PP subsystem

- *NB little change of this slide from Bham meeting*
- Database: add PPM (esp calibration data) and PPROD
- PPM module services: Asic/MCM test code developments should be a good starting point? But a lot more work is needed to enter this in the module services framework
- PPM simulation: still in progress?
- ROD module services and simulation: extend CP/JEP ROD work...
- Test vectors: analogue pulses exist, what about digital data to check BCID, jet summing, etc?

Whats missing? (4)

System environment

- We are currently using RedHat 7.2, Qt2.3, Online SRT 0.0.17, gcc 2.96
- CERN has now certified RedHat 7.3 with gcc 2.95.2 (gcc 3.2 also available)
- Online software releases include (CMT) 00-17-01, 00-18-00, 00-19-00
- New ROS software and this years test beam will use online-00-19-00 on RedHat 7.3
- Qt (used by HDMC) is now 3.1.1 (some API changes)
- Its never a good time to change, but we will have to do this – and fairly coherently across all our sites

Rough summary

Activity	Core	ROD	CP	JEP	PP	Slice
Database	90%	90%	80%	80%	50%	70%
Test vector gen	90%	90%	80%	50%	30%	50%
Simulation	90%	90%	90%	40%	20%	50%
Load hardware	90%	90%	80%	80%	20%	70%
Run control	80%	N/A	N/A	N/A	N/A	80%
Read out data	50%	90%	N/A	N/A	N/A	10%
Display results	60%	90%	60%	50%	10%	40%
Diagnostics	90%	90%	80%	90%	20%	40%
Monitor h/w	70%	20%	20%	20%	10%	10%
Calibration	40%	N/A	40%	20%	10%	20%
Event dump/disp	50%	20%	20%	20%	0%	10%

Timescales...eg CPROD tests

CPROD tests

- Working under run control since September. New firmware variants added recently

Tasks and Timescales (1)

Core and infrastructure

Activity	Duration	Date
Test/debug new database features	1 week?	March?
Multistep runs with test vectors	1 week?	March?
Test and use L1A generation via DSS	2 weeks?	April?
“allModules” package	1 week?	March/April?
System upgrades	2 weeks?	April????
Use ROS (2 RODs)	1 month?	May??
Use monitoring framework	1 week?	May??
Report hardware status via the IGUI	1 week?	April?

NB above assumes virtually no more ROD testing required!

Tasks and Timescales (2)

CP/JEP Subsystems

Activity	Duration	Date
CPM services and test programs	1 month?	April?
CMM services and test programs	1 month?	April?
CMM simulation (inc JEP variants)	1 month?	May?
JEM services and test programs	1 month?	April?
JEM simulation	2 months?	May/June??
CP/JEP hardware monitoring to IS	1 month?	May?
CP/JEP calibration/monitoring	1 month?	June??
CP/JEP subslice tests	months?	July???

NB above assumes virtually no more ROD testing required!

JEM simulation ideally needs a joint working visit to sort it out

Tasks and Timescales (3)

PP Subsystem

- Depends on availability of suitable person(s)...
  (able and willing to make a few working visits to the UK)
- Wait until Asic/MCM tests satisfactorily completed?

Activity	Duration	Date
PPM module services	2 months?	July?????
PPM simulation	2 months?	July?????
ROD simulation for PPM	1 month?	July?????

Timescale problems

Whats good

- Working together: day (or a few days) in one place tackling a problem. Eg Mainz/Stockholm visit in January was a good catalyst for developments

Whats necessary

- Meetings (but video conferences are hard work)

Whats not good

- Widely scattered developers: easy to be distracted to other problems, no chats over tea/coffee, tendency not to pick up the phone often enough, etc

People

Some hope...?

- Dave Kant (QM system manager) started working on ATLAS part time in January
- Full time (but short term) ATLAS post at QM may soon be filled...?

DIG training

- Another DIG training session is planned (very soon?)

Supported Test Plans

What we envisage the software will (and wont) do

- Configure an arbitrary system of known modules in many crates via run control – given a configuration created in advance by a database expert
- Generate test vectors to be loaded into an arbitrary subset of those modules – given suitable programs to generate the test vectors
- Simulate expected outputs from arbitrary input test data
- Start (and stop) single runs reading event data (via **one** DSS or one ROS) and comparing with expected data
- Start a multistep run reloading test vectors at each pause/resume transitions
- Perform multistep calibration runs scanning a **single** parameter for all modules of a given type (or maybe a subset of modules configured by a database expert)
- Nothing else! (Unless notified well in advance)
- To be on the safe side, why not prepare a detailed test plan?