Software

Murrough Landon – 15 July 1999

- Requirements
- History
- Recent Developments
- DAQ-1
- Future Plans
- Timescales
- Hardware Platforms
- Software Process

Requirements (1)

Draft requirements document...still to be completed.

- Standalone Tests/Diagnostics
- Data Acquisition
  - Test Beam Activities
  - Final DAQ in Atlas
- Calibration
- Online Monitoring
- Control
- Offline...

Tests and Diagnostics

- Detailed checks of single VME boards
- GUI to access all bits of registers/memories
- Flexibility for hardware development
- Prepackaged acceptance tests

Data Acquisition

- Many final requirements ought to be satisfied by the DAQ group
- Also need suitable environment for prototype tests (high on functionality, may be low performance)
- Infrastructure for initial implementation of final calibration, monitoring
- “ROD crate” DAQ?
- Still many questions to be resolved

Requirements (2)

Calibration

- Internal connectivity and timing setup
- Check connections to calorimeters
- Setup timing of calorimeter (test) pulses
- Determine BCID parameters
- Energy calibration

Many of the above will require access to and control of the appropriate calorimeter calibration systems.

Online Monitoring

- Check performance of the processor
- Monitor beam conditions
- Hot or dead cells

Control

- Control access to resources (local/remote, user/expert)
- User interface,setup parameters, database updates, etc

Offline

- Move existing offline software to OO framework
- Simulation and reconstruction for calibration and monitoring

Existing Software: Old and New

Data Acquisition

- Many test beam runs for our demonstrator programme
- DAQ system evolved for fast data collection and simple online monitoring
- Homegrown C using many C library packages
- Neither very user or programmer friendly

Diagnostics

- (Mostly) single board tests
- Tcl/Tk GUI to access all registers and memories
- C++ classes to access the hardware
- C++ VHDL-like model of the hardware
- Prepackaged tests of whole boards

Heidelberg Test Software

- Developed for PreProcessor test system
- Fully OO, GUI uses Qt package
- Very flexible, driven by configuration files
- Provides low level hardware access with some functions suitable for high speed tests
- Network VME access
- Still missing some desirable features

Future Directions

Not entirely clear!

Try to use DAQ -1

- We have already looked at DAQ -1 database
- Use DAQ -1 backend packages in our existing software
- Investigate whether proposals for ROD crate DAQ are suitable for our short (or long) term needs
- PRO: develop prototype of final software to run prototype modules
- CON: learning curves, timescale for early prototypes
- CON: may be difficulties working with DAQ group (source code, support)

Continue to evolve existing software

- PRO: quickly cope with continual stream of prototypes
- CON: effort wasted investing in obsolete framework
- CON: more difficult to conduct joint tests
- CON: we will have to make the change eventually
- CON: no real checks that DAQ groups products are satisfactory

More homegrown developments

- PRO: under our control
- CON: may take more effort than we expect
- CON: no gain for conducting joint tests

Timescales

Prototype Programme

- Heidelberg test system: now and for next 18 months
- LVDS tests with DSS module: next month or two
- Prototype (CP/JEP) ROD: autumn 1999
- Prototype CCM: late 1999, early 2000?
- Prototype CPM: mid/late 2000?
- Prototype Jet/Et system: mid 2000?
- Joint tests: variously late 1999 to 2001

DAQ group

- DAQ -1 evaluation: Dec 1999
- Technical proposal: Mar 2000
- Final DAQ: 2003?

Miscellany

Hardware Platforms: CPUs for VME access

- Existing setups mostly LynxOS on 68K or PPC
- DAQ group wont provide their code for 68K modules
- Short term, network access from eg Linux PCs via LynxOS server
- Test DAQ group supported platform: VMIC or SBS Bit3 interface?
- Present setup: multiple VME crates via VICbus
- Final architecture: CPU per crate?

Software Process

- Offline ASP in trouble? SRT to be redesigned?
- DAQ -1: backend group made heavy use of OO design tools and techniques
- DAQ -1: dataflow group used a less formal development process
- Some agreed formalism useful for a dispersed group
- Review process known to be useful for the hardware
- Dont want to discourage people who are new to OO
- CVS very helpful, documentation support vital
- Looking at Software through Pictures CASE tool