Future DAQ Software

Timescales: from now to 2001 we have DSS... ROD... CCM... CPM... etc

Possibilities:

- Continue with existing DAQ as long as possible
- Fresh start with DAQ-1 (not in time for ROD?)
- Gradual evolution, replacing a bit at a time
- Some combination of the above

We would like to have a prototype of the final software to accompany the full set of prototype modules.

Existing DAQ

- Historically only on LynxOS
- Just tried porting to Linux:
  - Convert Posix 4d9 calls to Posix1.b standard...
  - ...still some differences with Linux (shm_open)
  - Linux is missing Posix.1b semaphores (fake with SysV)
  - Linux doesnt have all the old BSD terminal calls – replace with Posix
  - Havent tried VME access (no VMIC or Bit3 etc)
  - New Makefile, remove explicit paths, etc
  - Modified code still seems to work on LynxOS!
- So we could carry on with existing DAQ on new platforms...
- ...ideally while replacing bit by bit

DAQ -1 Software

Yet another reminder! DAQ-1 has two major parts: the "dataflow" and the "back end".

Dataflow

- All the real time movement of event data
- One ROB crate controlled by "LDAQ".
- Monitoring at ROB, Event Builder interface, Event Filter

Back End

- Control side: highly distributed via CORBA
- Divided into many components
- Core: Configuration database, Run Control, Message Reporting, Information Service, Process Manager
- Others: Integrated GUI, Test Manager, various others

Use of DAQ-1

Incorporate Back End

- Backend software (not source!) on CD
- Replace config/parameter database in existing DAQ (Would need GUI to alter run parameters etc)
- Replace EMU with MRS/IS
- Reimplement DAQ processes as subclasses of Run Controller skeleton

Develop ROD Dataflow

- We are invited to help define/develop DAQ-1 in the ROD crate. Essentially means the dataflow side, communication protocol with RODs, etc.
- The LDAQ has a simple user interface to control one "crate". With VIC/similar intercrate bus this may be enough.

Considerations

- Would like to have some commonality with test/diagnostic developments, eg Module, Register, VME access classes?
- New modules (ROD) could be C++ class surrounded by C?
- May need to follow old and new paths in parallel for a while
- Should allocate people to well defined tasks