Atlas Level-1
Calorimeter Trigger

DAQ Status, UK

Status

and

Direction

Bruce M. Barnett

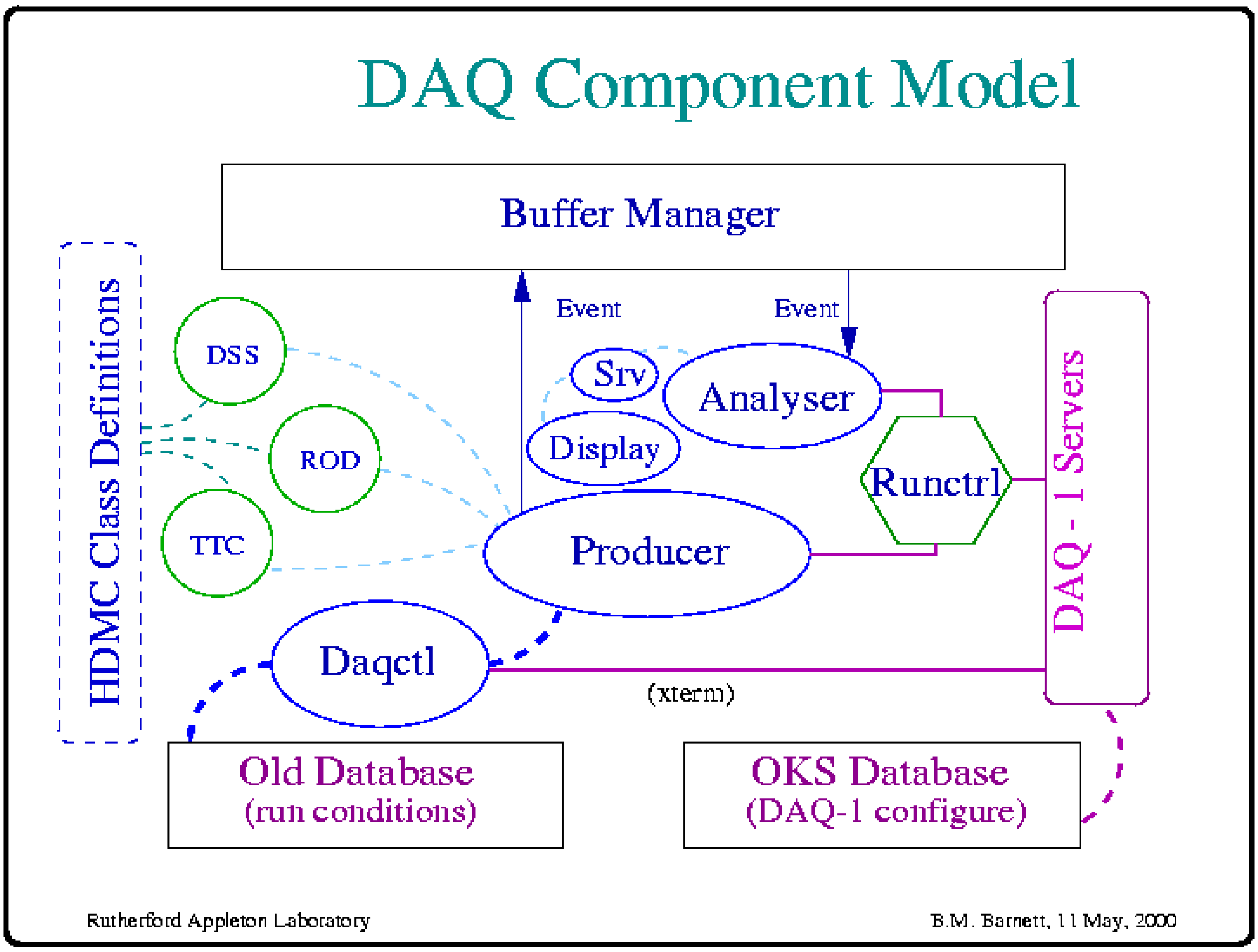
Overview

- Overview
- Goals
- Review of Components
- Buffer Manager
- Producer
- Analyser
- Database
- DAQ-1 Integration
- ROD Tests
- Future

Goals

- Replacement DAQ
  - The previous DAQ was based on last generation methodology and tools. It had grown to the point of being unmanageable and needed replacement.
- Integrated framework
  - With a parallel need for diagnostics, the opportunity exists to achieve an integrated approach to diagnostics and daq for the level-1 calorimeter project. This approach should achieve the most efficient use of resources, and the smoothest integration path. (inter level-1 and inter ATLAS DAQ).
- Extendable
  - If the OO design is robust, the code developed for short term tests should be adequate for long term use and is expected to be extendable in that direction.
- Experience
  - At worst, it will act as an invaluable prototype.
- Vehicle for Tests
  - The code will provide the vehicle for tests, initially of isolated modules and later of their connectivity, (ROD,DSS, TTCVI) then of their interaction, finally (CPM, CMM) in their slice test environment.

Review of Components



Buffer Manager

- PBM
  - Functionality
    - In a stand alone DAQ partition, one needs somewhere to put the data in a way that allows multiple process access at a subsequent stage. The traditional Portable Buffer Manager satisfies this need.
  - Personality
    - The PBM Needs an OO façade. This was prepared before the last meeting in a preliminary form (for both source and sink functionality). The paradigm is of an event store which provides linear sequences of memory to readout methods of hardware-classes.
  - Progress.
    - Not much has been done since July on this front. Additional attention is still required to:
      - event formatting at the producer end.
      - event retrieval by the analyser

Producer

- Event Format
  - To bring us into conformance with some standard, it was decided to wrap data in an ATLAS standard format (DAQ-NOTE 50, ATL-DAQ-98-129)
  - Wrapper code has been provided by H.P. Beck and its suitability for direct use (and incorporation) is awaiting study.
  - Ideas on how to adapt to the DAQ event format have been proposed. The ROD fragment is the lowest level fragment that is defined. For data from modules other than RODs, the idea is to define pseudo-fragments that are in appearance ROD fragments (as that is the lowest level that is defined by the above document.)
- Class Definitions (HDMC Toolkit).
  - A presentation at the software meeting outlined extensions to HDMC to allow logical grouping of module registers to permit the construction of high level classes suitable for their functional (as opposed to elemental) representation. Work for CPM and ROD is progressing well.

Analyser

- Functionality
  - Grab the data from the PBM and decode it. This requires several capabilities.
- Event Dump (low level)
  - Scott T. (Birmingham) has developed (is developing) an event dump which expects a pointer to an (ATLAS) event and analyses the structure thereby recognising isolated event fragments of various types.
- Event Analysis
  - How does the data inter-correlate. No progress.
- Display: l1histos.
  - l1histos
    - This ROOT package from Tara S. takes groups of histograms defined in the old-daq-database and creates equivalent ROOT histograms. The package is stable (awaiting correction of an identified bug).
  - Histogram Database
    - To aid in the parsing of the old database, thereby gaining independence from much old code, new classes to parse the old database were added. (Horse/Cart problem).

Database

- HDMC files:
  - Allow representation of module register content as well as system architecture (what modules, where) and bit definitions.
  - HDMC needs to define a database interface conformant to DAQ-1 (IS) which interacts appropriately with this file based mechanism.

- DAQ (Producer)
  - is able to use HDMC parts files to create modules and associated classes in lists appropriate to the run state model of DAQ-1. If the HDMC interface is extended to access a DAQ-1 database, the move to its use from within the producer (or analyser, in fact) should be effortless.

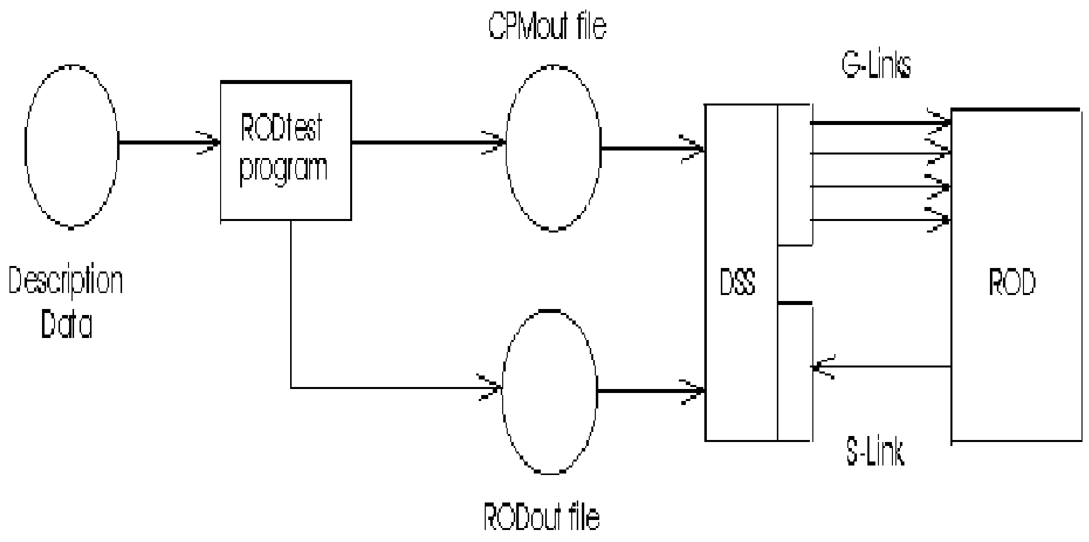
- Python and IS
  - Murrough L. has written a Python interface to a primordial IS based database. This may be a first step of the HDMC integration mentioned above.

DAQ-1 Integration

- Prozaq
  - Which has been described earlier is a basic framework which uses the DAQ-1 GUI and state model and constructs “controllers” appropriate to a level-1 partition. Progress in incorporation of Analyser or Producer code into this framework has been. Not.
- DAQ-1
  - meanwhile has matured from version 008 to 010, the latter containing event-monitoring support and some enhancements in the handling of run controller states which are performance enhancing.
- Next Steps.
  - Integrate Producer into prozaq DAQ-1 run controller
  - Upgrade to 010

ROD Tests

- Test Vectors
  - Tests of each system require input and output vectors which emulate missing hardware or complement the available natural phase space.
  - In the case of the CPM and ROD, Bill S. (Birmingham) has specified a method of generating files containing such vectors, so that the DSS module may be loaded with CPM-like output data and the DSS input may be compared with expectation:
    - http://www.ep.ph.bham.ac.uk/user/stokes:



Future

- Complete basic DAQ work, DAQ-1 interaction
- For Each Module {
  - design HDMC basic part file, configuration file.
  - define HDMC module content parts file (I have n modules of type newModule, with addresses 0x00xxx... respectively).
  - define register associations (Together and HDMC “register inheritances” from the basic part file)
  - generate code functional classes, adhering to daq state-model API (load, configure, run, unload, de-configure)
  - build classes into system

  }
- Define test vectors appropriate to each module, spanning the set of basic, connected and interactive functionality.
- Define test vectors for each sub system …
- Define test vectors for each system ...