ATLAS Level-1 Calorimeter Trigger 15 August 2001
Software Note 007 Version Draft 0.4b

L1Calo Trigger Online Software Requirements

Bruce Barnett, Eric Eisenhandler, Norman Gee, Steven Hillier, Murrough Landon, Gilles Mahout, Oliver Nix

1 Introduction

This document defines the user requirements of the online software for the Level 1 Calorimeter Trigger.

We start by setting the context of L1Calo Online Software and its interactions with the L1Calo hardware, various data and other software and hardware systems in ATLAS.

We then discuss the decomposition of our software project into a number of identifiable areas or "themes". These themes roughly correspond to the major groups of use cases for the software project.

We list the overall requirements and constraints for the project. Then for each of the identified areas, we set out its scope and a summary of its particular requirements. This is not done in great detail as each area will be implemented by one or more software packages which are expected to have dedicated documents of their own.

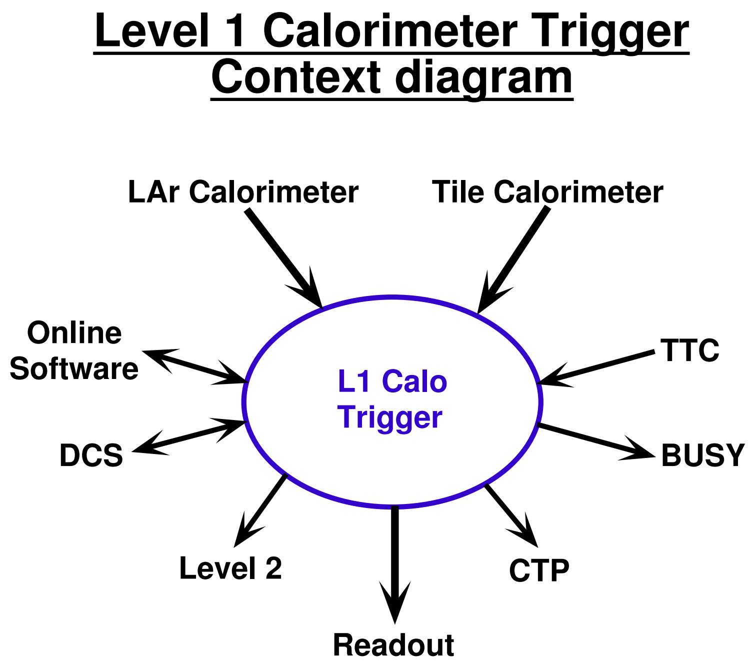
2 Overview of the Trigger

The design of the level 1 calorimeter trigger hardware is described in the TDR [1] and subsequent updates [2]. In this section we very briefly summarise the trigger system and its environment in ATLAS. More information about the trigger may also be found on the web [3].

2.1 Hardware Context

The overall context of the calorimeter trigger is shown in figure 2.1.

The trigger receives data from both LAr and Tile calorimeters. The real time trigger results are passed to the CTP. If these cause a Level 1 Accept (L1A) the region of interest data (RoIs) are passed to Level 2 and detailed readout of the trigger system is sent to the ATLAS readout system (ROS). The TTC system sends clocks, L1As and synchronisation commands to the trigger, while the trigger readout drivers (RODs) can inhibit further L1As via the BUSY signal. The trigger hardware is monitored via DCS. The whole system is configured, controlled and monitored via the L1Calo Online software which is the subject of this document.



L1Calo system context diagram.

2.2 Hardware Configuration

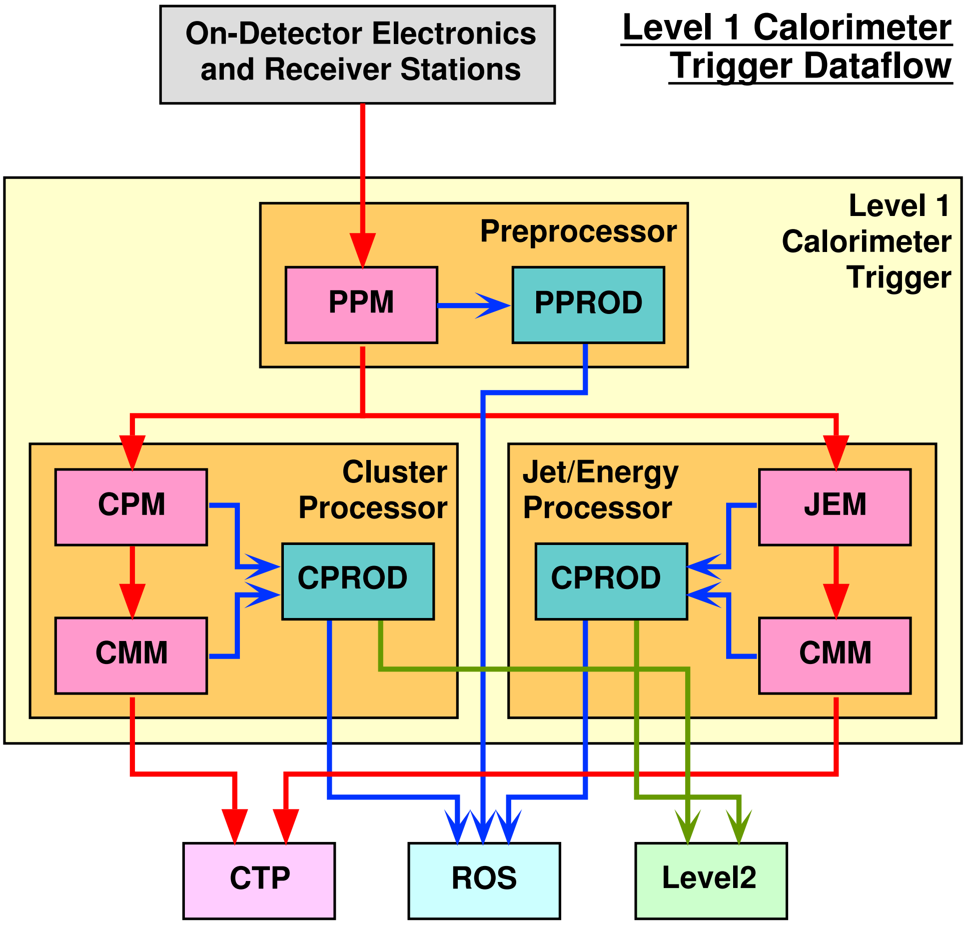
The L1Calo trigger consists of three major subsystems: Preprocessor, Cluster Processor, and Jet/Energy-sum Processor. Figure 2.2 shows the main components of each subsystem and the dataflow through them. The readout can capture data from various points through the system, typically the inputs and outputs of each module, and in some cases intermediate processing within a module. Test vectors can be loaded at many of the points from which real data can be read.

All three subsystems will be implemented in a number of crates of custom electronics. The data read out to DAQ and to Level 2 passes through RODs. In the Preprocessor these are housed in the same crates as the Preprocessor modules, while in the two trigger processors they are in separate ROD crates. There will also be a 6U crate housing a TTCvi and BUSY modules. Each of of these crates will be controlled by a VME single board computer. The computing infrastructure will also include some PC workstations to act as disk servers and to provide graphical interfaces.

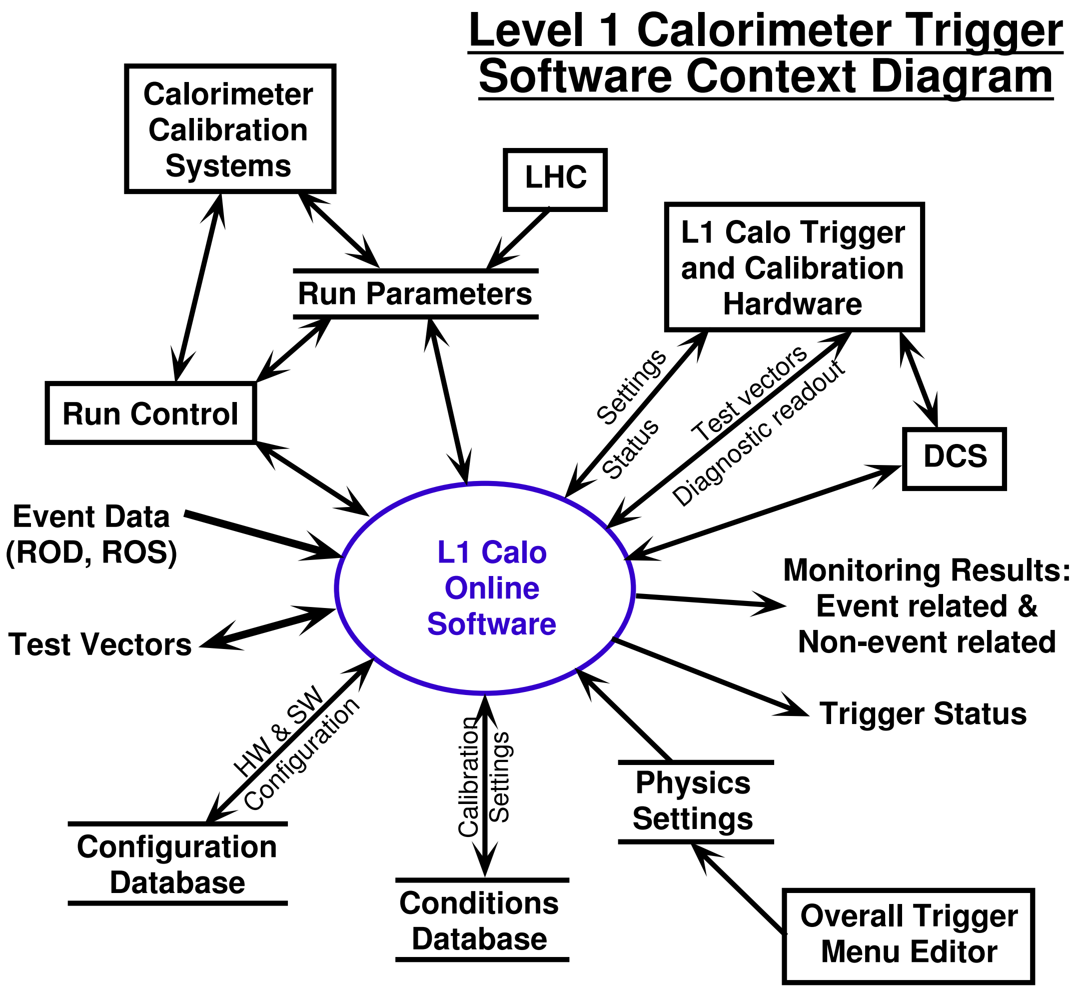
3 Software Context

The context of the online software for the trigger system is shown in figure 3.

The software has many kinds of interaction with the trigger hardware. These range from initialisation and monitoring during normal running to various test and diagnostic activities. We will also want to communicate with DCS which monitors physical properties of the crates and modules. Parts of our software will interact with the ATLAS run control system. The run control system manages all ATLAS detectors, but of particular



Dataflow through the L1Calo subsystems.



L1Calo software context diagram.

interest to us are the calorimeter calibration systems and LHC beam information. Our online software will also deal with a variety of kinds of dynamic data such as events read from the trigger system and the results of analysing it, test vectors of many kinds, and status information. The software also has to use and maintain several databases: the hardware and software configuration, the calibration data, and the collection of trigger menus.

3.1 Software Areas

Considering the software context diagram, we identify the following aspects which must be covered by the body of online software to be developed for the calorimeter trigger.

- Collection of events directly from our RODs or via the standard ATLAS readout chain.
- Analysis and monitoring of event data.
- Collection, monitoring, presentation and archiving of non-event information including periodic checking of hardware settings.
- Collection, distribution and presentation, archiving, reference and trends of histograms (and tables, etc).
- Calculation, distribution and display of status information.
- Derivation, archiving and retrieval of calibration settings both for internal calibrations and those involving the calorimeters and the LHC.
- Generation of test vectors, including simulation of the output from subsequent stages of the trigger.
- Testing of the system by injecting test vectors and verifying the outputs. This may be offered as a service to external systems such as Level 2 and the CTP.
- Interactive display and testing of modules and their internal components.
- Trigger menu processing, ie editing the trigger menu and generating settings files
- Editing and display of the trigger hardware and software configuration
- Loading and verifying hardware settings
- Processing run control commands
- Editing and display of run control parameters
- Interaction with the detector control system (DCS)

Naturally, the scope of our online software does not include any purely offline activities, such as physics analysis or detailed studies of trigger performance. We also exclude the development of firmware from this document. There will, however, be areas where the boundaries become blurred. The derivation and treatment of some calibrations may be one example.

4 Users

Before listing the requirements, we describe the intended users of the L1Calo online software. We envisage the following broad categories of users.

- **DAQ shift operators** are assumed to need little knowledge of the L1Calo trigger, either its hardware or software. Shift operators may however have to perform some frequently required calibration procedures. They may receive error or warning messages if the L1Calo software detects problems in the hardware. They may have to run high level diagnostics.

- **L1Calo experts** are assumed to have a good knowledge of the L1Calo trigger hardware. They are expected to be familiar with the user interfaces of the software, including aspects intended only for experts. They are not expected to know much about the internal details of the software.

- **L1Calo developers** are assumed to have a good knowledge of the internal structure and operation of the L1Calo online software. They may have to diagnose faults in the software itself, as well as extend and improve it.

5 Requirements

This section is divided into requirements for our online software as a whole, and more specific requirements for the various areas listed above.

For the moment, no distinction is made here between what are more formally distinguished as “constraints”, “system properties” or “functional requirements”.

In the following sections, the priority for implementing the various requirements is indicated by the approximate timescale on which they are needed. The following keywords are used:

- **Module Tests**: required for module and subsystem testing

- **Slice Tests**: required for or during the slice tests

- **Final ATLAS**: required after the slice tests

5.1 Project Management

These requirements are not strictly requirements on the software itself, but on the software development process and on the management of the software.

PM01 The software should be developed using an agreed software process.

Timescale: Module Tests

The software process is still being defined. It should probably include (a) use of central repository and version control system for documentation and source code; (b) some quality control or reviews; (c) management of releases; (d) naming and coding conventions; (e) use of common tools.

5.2 Global Requirements

The following requirements are imposed on the whole range of our software. Many are fairly obvious. There is also some overlap with the general requirements of the offline software. However it seems sensible to reiterate them.

GL01 The software shall run on the hardware platforms and operating systems supported by the ATLAS DAQ group as approrpriate.

Timescale: Slice Tests

The currently supported platforms use different "endian" conventions and might in future use different word lengths. Our software must be independent of these differences.

GL02 The software shall interwork with other ATLAS software and comply with established ATLAS standards.

Timescale: Slice Tests

This implies the use of common ATLAS libraries and tools and in particular the ATLAS Online software [4]

GL03 The software shall work on both the complete trigger system and a variety of partial configurations, test setups, etc. Any operations which require components that are missing from a partial configuration shall fail gracefully.

Timescale: Module Tests

Emulation of some critical components may need to be provided.

GL04 The software shall be accompanied by documentation both for L1Calo developers and other users.

Timescale: Module Tests

This should include interactive help where appropriate.

5.3 Event Collection

These requirements relate particularly to collection of events from our RODs via VME, ie "ROD crate DAQ" in some shape or form. We have similar requirements of the main ATLAS readout system (ROS) but we are not responsible for satisfying them.

RD01 It shall be possible to collect events in the absence of the standard ATLAS readout system, ie via VME.

Timescale: Module Tests

RD02 The readout software shall be able to collect synchronised events from multiple PP and/or CP RODs in multiple crates.

Timescale: Slice Tests

For VME readout, this requires suitable firmware support in the RODs.

RD03 The readout software shall make events available for monitoring.

Timescale: Module Tests

RD04 It shall be possible to monitor a fraction of events while the trigger is running at the maximum 100kHz rate.

Timescale: Slice Tests

RD05 It should be possible to collect only events flagged as containing errors by external checking hardware (eg the DSS).

Timescale: Module Tests

This may be more difficult when using the ROS?

RD06 It may be possible to collect diagnostic information, via VME, from non-ROD modules at low event rate and feed this into the DAQ system.

Timescale: Final ATLAS

This requires suitable firmware support in these modules.

RD07 It may be possible to record events.

Timescale: Final ATLAS

Though we dont plan to do this at the moment.

5.4 Event and Non-event Monitoring

This section concerns the two types of monitoring which cover different areas, but which have many requirements in common. The following two sections (5.5,5.6) are concerned with the distribution and display of the results of monitoring activities. There is a certain overlap in these requirements.

The first area of monitoring is the analysis of events or event fragments collected from any part of the readout chain in order to monitor the performance of the trigger hardware and the physics performance of the detector.

The second area is the collection of non-event related data. We expect that the local run controllers or other programs running in each crate will periodically check the status of the hardware and collect status information, eg link errors, PP rate histograms, etc. The local controllers may check that the settings loaded into the hardware at run start are still correct.

A more detailed discussion of what we want to monitor is given in a separate document [5].

MN01 The event analysis and monitoring programs shall use the same interface to obtain events whether the events or event fragments are collected from the ROS (or later in the DAQ chain) or directly from our RODs via VME.

Timescale: Slice Tests

MN02 Non-event related status information shall be read periodically from all major components of the system (eg modules, crates).

Timescale: Final ATLAS

Typical monitoring frequencies are: PPM rates histograms every few ms, CPM and JEM link errors every few ms, verification of loaded parameters every few seconds.

MN03 The results from event and non-event related monitoring, whether in the form of histograms, tables, or other data formats, shall be made available for collection and display by processes running on any suitable workstation.

Timescale: Slice Tests

Eg via the Information Service.

MN04 Any hardware malfunctions detected by monitoring shall be logged and communicated as warning or error messages to the DAQ shift operator.

Timescale: Final ATLAS

Serious errors may force a run control state change.

We will also want to correlate any problems with conditions observed by DCS.

5.5 Distributed Histogramming

We expect that programs running in several crates and PCs will be producing monitoring histograms (and other types of data). In general the bulk of data is best kept locally, but needs to be made available for display at any other workstation.

We may need to maintain a (mostly existing) package ourselves, or we may be able to obtain another package which meets our requirements, eg from Level 2.

DH01 It shall be possible for histograms created on any machine in the Level 1 system to be displayed on any suitable workstation.

Timescale: Slice Tests

DH02 It shall be possible to archive histograms and retrieve them from the archive.

Timescale: Final ATLAS

DH03 It shall be possible to compare any histogram with a reference copy of that histogram stored in the archive and to track changes over time.

Timescale: Final ATLAS

5.6 Status Information

In addition to histograms which can be displayed on demand, we also expect some monitoring programs to produce summary status information. This will be collected in one or more dedicated displays. Some of these may be panels implemented in the IGUI component of the ATLAS Online software. Others may be standalone programs.

SI01 Status information obtained locally from the major components of the system shall be collected and a summary presented by a GUI.

Timescale: Final ATLAS

SI02 Detailed status information shall be presented to L1Calo experts on request.

Timescale: Final ATLAS

SI03 It shall be possible to view status information from any suitable workstation.

Timescale: Final ATLAS

5.7 Calibration

Full descriptions of our calibration, setup and integrity check procedures are given in another document [6]. Some initial suggestions for organising the calibration data are shown in a separate note [7] which is still evolving. Here we just list the main requirements of the calibration software.

CA01 The calibration software shall be able to perform all the types of calibration listed in the calibration document [6]. These include both standalone calibrations and those done in conjunction with the calorimeter calibration systems.

Timescale: Slice Tests

CA02 The results of the calibrations shall be stored in the standard ATLAS database.

Timescale: Final ATLAS

This should include details of how the calibration was performed.

CA03 It shall be possible to generate initial or test calibration files manually.

Timescale: Module Tests

CA04 It shall be possible to analyse trends in successive calibrations with time.

Timescale: Final ATLAS

This is probably a requirement on the offline conditions database – though we may need tools to use it?

5.8 Test Vectors and Simulation

We need to be able to produce, either in advance or on the fly, a large variety of test vectors for different parts of the system. We need both the input test vectors and the expected results from subsequent stages of the processor and readout chains.

TV01 The software shall be able to produce test vectors suitable for loading into all suitable points of the trigger hardware.

Timescale: Module Tests

This implies coverage of the possible phase space.

TV02 It shall be possible to produce test vectors for single elements (eg one FPGA or Module), complete subsystems (eg the whole Preprocessor system) or the complete trigger system.

Timescale: Slice Tests

TV03 Given a set of test vectors loaded into one part of the trigger, it shall be possible to predict the output expected at subsequent stages of the trigger and readout chains.

Timescale: Module Tests

TV04 The simulation of the trigger shall be sufficiently detailed to enable detection and localisation of faults at the level of an individual module or link between two modules.

Timescale: Slice Tests

TV05 The simulation may be available at different granularities.

Timescale: Final ATLAS

Eg for performance reasons?

5.9 System Testing

We need to be able to inject test vectors into the hardware, run the system and verify the correct operation of the system. Some, but not all, of our test procedures will be packaged for use by the Test Manager component of the ATLAS Online software. And some of these will be simple non-destructive tests suitable for diagnostic use during normal running.

ST01 There should be several levels of tests available for each component, from the superficial to the exhaustive.

Timescale: Slice Tests

ST02 There shall be tests available for single components, for collections of the same component, for complete or partial subsystems and for the complete trigger system or for a slice through the complete system.

Timescale: Slice Tests

ST03 For each type of component or subsystem, there shall be at least one test which can be run by the Test Manager.

Timescale: Slice Tests

Presently this implies a command line program returning a Posix 1003.3 compliant result (pass, fail, unresolved or untested) with optional additional explanatory error messages.

ST04 There shall be tests for the external interfaces of the trigger system.

Timescale: Final ATLAS

Eg to the CTP and Level 2.

ST05 It should be possible for the user to define named series of tests which can subsequently be performed as a single operation.

Timescale: Final ATLAS

This requirement may be satisfied by using the Test Manager and Diagnostic Verification packages of the Online software – or future extensions of them.

5.10 Interactive Diagnostics

We need to have interactive access to detailed aspects of the hardware via user friendly graphical interfaces.

HD01 The software should present a graphical view of the available hardware configuration(s).

Timescale: Slice Tests

HD02 It shall be possible to see the detailed structure of each component from the crate down to bit fields in individual registers.

Timescale: Module Tests

HD03 The significance of each bit or bit-field in registers and memories should be separately displayed or explained.

Timescale: Module Tests

HD04 It shall be possible to execute all supported low level actions on the hardware.

Timescale: Module Tests

Ie reading and writing to individual registers and memories.

HD05 It should be possible to execute higher level functions on more complex components such as FPGAs and Modules.

Timescale: Slice Tests

Dedicated GUIs may be required here.

HD06 The interactive diagnostic software should use common software infrastructure with other level-1 and DAQ components where reasonably possible.

Timescale: Final ATLAS

For example the Configuration database and low level VME access libraries.

5.11 Trigger Menu Processing

The trigger menu has to be created coherently for all levels of the trigger, including all parts of Level 1. We will need to provide some tools (a) to edit and display parts of the trigger menu specific to the calorimeter trigger and possibly (b) to convert the trigger menu representation into objects or data structures suitable for us to download into our hardware, including mappings to CTP inputs. If the latter is not completely handled by the trigger editor, we may need to add an extra requirement here. An initial suggestion for the L1Calo trigger menu schema is shown in a draft note [7].

TM01 It shall be possible to edit and display the L1Calo specific parts of the trigger menu.

Timescale: Slice Tests

5.12 Hardware and Software Configuration Database

We need to provide tools (or extend existing tools) to edit and display our hardware and software configuration. For example, we need to add specific Module subclasses to the general Online database schema. We also need a way of describing connections between modules. A draft note [7] will address these issues in more detail.

DB01 The configuration database tools shall be able to edit and display the hardware, firmware and software configuration of the L1Calo system.

Timescale: Slice Tests

This includes parameters appropriate for each type of module; the description of the connections between modules; the collections of FPGA programs to be downloaded into each FPGA; etc.

5.13 Load and Verify Hardware Settings

We have to provide the code that takes database objects and uses them to download all the necessary settings into the modules. This will normally be done under the command of the run control system. Other packages may need particular settings and test vectors for test and diagnostic purposes, but should load the modules using the same software infrastruture.

LH01 It shall be possible to load each type of module with settings and FPGA programs derived from the hardware and firmware configuration, calibration and trigger menu databases.

Timescale: Slice Tests

LH02 It shall be possible to verify that each module has been correctly loaded.

Timescale: Slice Tests

Assumes that settings can be read back from all modules. This is not completely true for the TTCvi.

LH03 It shall be possible to choose particular collections of settings from a pool.

Timescale: Final ATLAS

This is appropriate for the trigger menu. Also probably for calibrations for different run types.

LH04 It shall be possible, for special runs, to load test vectors into selected types of module.

Timescale: Module Tests

5.14 Run Control

We have to implement run controllers in each of our crates. We also need some extra run controllers for synchronisation. More details, including a more detailed list of requirements, are given in a separate document [8] which also briefly addresses the need to provide IGUI panels to allow run parameters to be set and displayed.

RC01 It shall be possible to fully initialise and control the L1Calo system as part of the ATLAS Run Control system.

Timescale: Slice Tests

This implies providing suitable run controllers using the Online Run Control framework.

RC02 It shall be possible to select different run types and run parameters to control the detailed operation of the L1Calo system.

Timescale: Slice Tests

5.15 DCS

We expect that our requirements in the DCS area will be met by the DCS SCADA software and by the DAQ-DCS communication package of the ATLAS Online software.

More detailed requirements in this area may be specified later.

5.16 Access Control

It will be necessary to control access to the hardware at certain times, eg during normal runs. It will also be sensible to ensure that only authorised experts can make changes to vital configuration data.

AC01 Changes to the configuration shall only be made by authorised users and should be logged.

Timescale: Final ATLAS

Standard operating system and file system controls together with a version control package ought to be sufficient for this purpose.

AC02 It should be possible for authorised users both local and remote from the ATLAS experiment to access the trigger system in any of the roles listed in section 4.

Timescale: Slice Tests

The Resource Manager may be used for this – though it only imposes "cooperative locking".

AC03 It should be possible to "lock" the hardware and maybe other resources against accidental access.

Timescale: Slice Tests

The Resource Manager may be used for this – though it only imposes "cooperative locking".

5.17 Logging

We need to ensure that a complete "audit trail" exists for changes made to the system and its state during each physics run.

LG01 A description of the complete configuration of the trigger system shall be recorded for every run.

Timescale: Final ATLAS

We may be able to use the Online Bookkeeper package to fulfil this requirement.

A Glossary

CP	L1Calo Cluster Processor
CTP	Central Trigger Processor
DCS	Detector Control System
DSS	Data Sink and Source Module
GUI	Graphical User Interface
IGUI	Integrated GUI (part of the ATLAS Online Software)
JEP	L1Calo Jet/Energy Processor
L1A	Level 1 Accept
L1Calo	Level 1 Calorimeter Trigger
PP	L1Calo Preprocessor
ROD	Readout Driver Module
RoI	Region of Interest
ROS	Readout Subsystem
TDR	Technical Design Report
TTC	Timing, Trigger and Control System

References

[1] ATLAS Level 1 Calorimeter Trigger TDR
http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/tdr.html

[2] ATLAS Level 1 Calorimeter Trigger TDR updates
http://hepwww.pp.rl.ac.uk/Atlas-L1/TDR/Updates/TDR.html

[3] ATLAS Level 1 Calorimeter Trigger Home Page
http://hepwww.pp.rl.ac.uk/Atlas-L1

[4] ATLAS Online Software
http://atddoc.cern.ch/Atlas/DaqSoft/Welcome/html

[5] L1Calo Monitoring Requirements
http://www.hep.ph.qmw.ac.uk/l1calo/doc/pdf/MonitoringReq.pdf

[6] L1Calo Configuration Database
http://www.hep.ph.qmw.ac.uk/l1calo/doc/pdf/Calibration.pdf

[7] L1Calo Calibration and Setup Procedures
http://www.hep.ph.qmw.ac.uk/l1calo/doc/pdf/Calibration.pdf

[8] L1Calo Local Controllers
http://www.hep.ph.qmw.ac.uk/l1calo/doc/pdf/LocalControllerSpec.pdf

[9] Configuration Data for the ATLAS Level 1 Calorimeter Trigger (ATL-DA-EP-02)
http://edmsoraweb.cern.ch:8001/cedar/doc.page?document_id=111349

Note that some of the above documents are still under development. Those which have been approved have version numbers starting from 1.0, while those that have not been reviewed are labelled as "Draft".