Requirements on the Conditions Database interface to T/DAQ

ATLAS TDAQ/DCS Online Software - Lisbon Group

European Laboratory for Particle Physics
Laboratoire Européen pour la Physique des Particles
CH-1211 Genève 23 - Suisse

ATLAS TDAQ/DCS Online Software

Requirements on the ConditionsDB interface to T/DAQ

Document Version: Draft
Document Date: 2002-9-19
Document Status: Draft

Abstract

This document describes the requirements on the usage of the ConditionsDB interface for the Trigger/DAQ system. It addresses the requirements associated with the detector front-end, the LVL1, the DAQ and the HLT systems. It deals mainly with calibration, alignment and robustness information but also discusses the conditionsDB interface to the Detector Control Systems and the detector and trigger configuration.

Institutes and Authors

CERN: Chis Bee, Saul Gonzalez, Murrough Landon, Thorsten Wengler, Mihai Caprini, Levi.Lucio, Igor Soloviev,
CPPM: Francois Touchard,
FCUL: A. Amorim, J. Lima, L.Pedro,

1. Introduction

The Conditions Data are the data necessary for reconstruction and analysis of raw data. These includes Calibration, Alignment and Robustness data, as well as data coming from DCS that will be relevant for event processing.

The Conditions Database should provide an integrated interface to all information that is not stored in a per event basis but that can change from time to time. This includes:
- Calibration and alignment information together with changes in configuration imposed by the functioning and roboustness of the detector;
- Common access interface to DCS information;
- Support for the versioning of the configuration geometry and detector description databases (history mechanisms required to deal with revisions).

The online environment is both the provider and the client of the information on calibration, alignment and robustness. The trigger systems have to cope with changes in condition information within a reasonable delay. The dependencies of the particular trigger on the fine details of the calibration information and the ability to correct for miscalibration offline without trigger bias determine the acceptable delay. The importance of this aspect of the ATLAS database system to the TDAQ is clearly stated in section "Future Work" 10.3.3.2 of [HIGH-LEVEL TRIGGER, DAQ AND DCS [1]

This document has been organised taking into account the information related to:

1- Calibration, alignment and channel status of the detector

2 - Access of general users to the Detector Control System information

3 - Management of versions associated with the ATLAS configuration databases.

The ConditionsDB has[6] evolved during the last years, departing from the BaBar implementation based on ObjectivityDB. It has been redesigned by CERN/IT as common layer for several LHC experiments and it has been recently re-implemented using comercial and Free software RDBMS. Fig. 1 illustrates the main concepts of the ConditionsDB.

The present interface or invent specifications only define a hierarchical directory-like structure to store objects that are contained in folders where they are indexed by time intervals and revisions (versions). it does not address either the nature of the stored objects or, in particular, their schema or their access methods. It was assumed that the object serialization itself was under the control of the underlying ODBMS, as was the case for the ObjectvityDB implementations. The ATLAS offline working model consists of the definition of the ConditionsDB objects inside the ATHENA framework by their transient representation and ATHENA-defined serialization procedures. The efforts common to all experiments in the LCG forum have redefined this policy by making the database schema accessible outside of any experiment-specific framework.

Figure 1 illustrates a general overview on how the Conditions Database works. Each conditions data item is uniquely identified in three dimensions: data item name, version and validity.

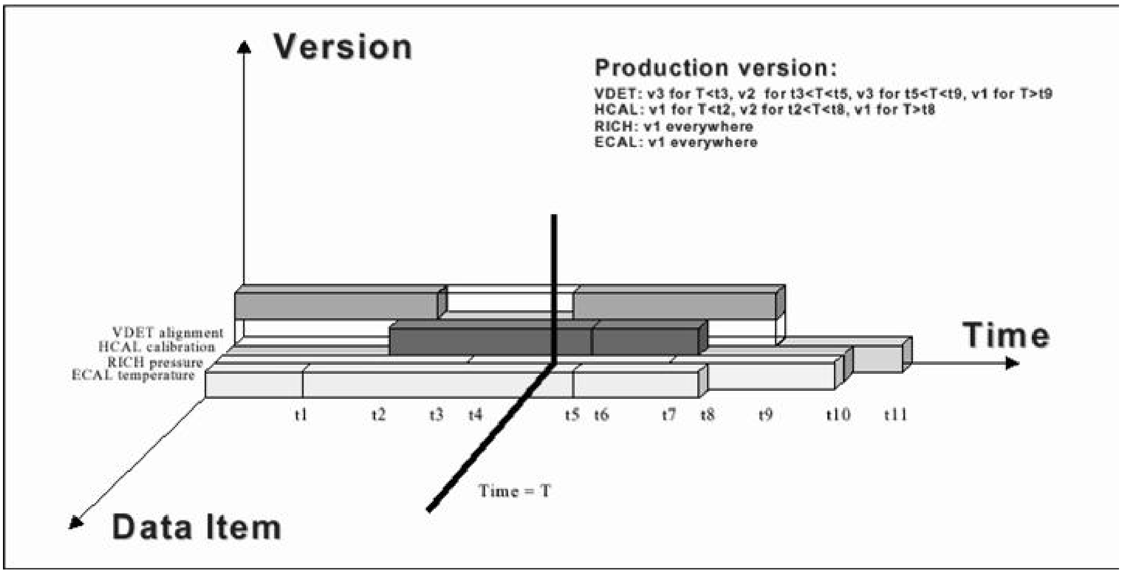

Figure 1: The concepts involved in the present ConditionsDB interfaces

1.1 Purpose of the document

This document presents the Requirements for the ConditionsDB interface of the ATLAS TDAQ/DCS Online Software. It will be the basis for the design and implementation of the ConditionsDB interface in the context of the ATLAS Trigger DAQ/DCS system.

1.2 Glossary, acronyms and abbreviations

1.2.1 Glossary

Alignment:

Calibration:

ConditionsDB:

Roboustness:

1.2.2 Acronyms and Abreviations

DCS: Detector Control System - Related to Slow Control

EF: Event Filter

FSM: Finite State Machine

HLT: High Level Triggers

LVL1: First Level Trigger

LVL2: Second Level Trigger

ROB/ROS: Read Out Buffler/Read Out System

ROD: Read Out Crate

ROI: Region Of Interest

1.3 Referencies

[1] HIGH-LEVEL TRIGGER, DAQ AND DCS Technical Proposal, CERN/LHCC/2000, - 2000-03-31

[2] The Computing Model from TDAQ to Physics Analysis - Requirements and Strawman Design, - 0000-00-00

[3] Conditions DB Interface Specification, Stefano Paoli - 0000-00-00

[4] ATLAS Databases Basics, R.D. Schaffer - 0000-00-00

[5] ATLAS High Level Triggers, User requirements on Conditions Database, S. Gonzalez and F. Touchard - 0000-00-00

[6] Conditions DB - User requirements and Analysis Document (http://wwwinfo.cern.ch/db/objectivity/docs/conditionsdb/urd/urd0.6.pdf), Stefano Paoli - 2000-11-20

[7] ATLAS HLT - User Requirements on Conditions Database, S. Gonzalez, F.Tiuchard - 2002-04-08

1.4 Context of the online interface to ConditionsDB

The online interface to ConditionsDB provides services associated with the detector front-end, the LVL1, the DAQ and the HLT systems.

The interface to the ConditionsDB will provide time management for the different conditions data like:

1. Calibration, Alignment and Robustness
2. Detector Control Systems data for event processing
3. Detector and trigger configuration

The conditions data information needs to be analysed in terms of the:

1. information sources
2. information clients
3. update control

One can classify the users of the ConditionsDB related to the TDAQ world as:

1. ROD

2. LVL1
3. DataFlow
4. LVL2
5. EF
6. (interface to) Prompt Reconstruction
7. (interface to) Offline Computing

The requirements on information provision and clients in the above subsystems can be decomposed into the information domains associated with the several ATLAS subdetectors.

1.5 General capabilities of online interface to ConditionsDB

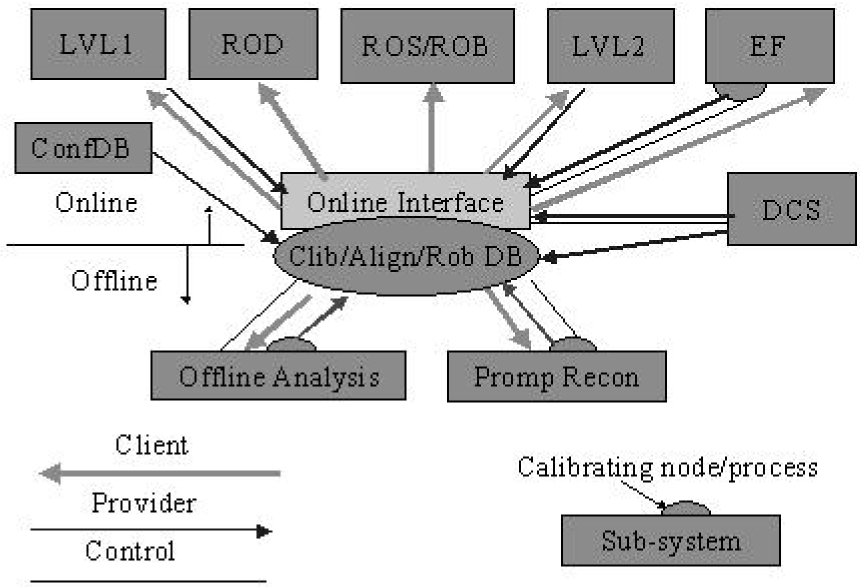

Figure 2:

1.6 General constraints on online interface to ConditionsDB

1.7 General assumptions and dependencies

1.8 User characteristics

Specific Constraints, Assumptions/Dependencies, Use Cases and Requirements

2.2 **Constraints**

CO 01: Time availability
The ConditionsDB online interface should make the conditions information accessible during and outside data taking sessions.
Author: António Amorim

CO 02: Platform availability
The ConditionsDB on-line interface should make the conditions information accessible in all computing platforms involved in the ATLAS/TDAQ including real-time systems.
Author: António Amorim

2.3 Assumptions and Dependencies

AD 01: Assumption
The online system is a provider and a user of Calibration/Alignment/Status information. As a provider, most information should come from reconstructing events but one has also to account for changes notified by the DCS system (ex: machine conditions)
Author: António Amorim

AD 02: Dependence from the conditionsDB common interface
The ConditionsDB interface is defined as an LHC common project. It provides the interface model for the online environment. It does not provide object schema information and must be therefore complemented by another component that must be common to the entire collaboration.
Author: António Amorim

2.3.1 Calibration, Alignment and Robustness Assumptions and Dependencies

AD 03: Dependency from information update rate and frequency.
The model of an ideal detector assumes that the calibration and alignment tasks are going to be performed during an initial stage and will be valid for a long period afterwards. Determining the expected flow of calibration updates and their rate requires a model for the behaviour of the detector which is critically dependent on the corrections to the ideal behaviour. These effects are generally hard to predict with reasonable accuracy. Due to the high performance nature of the system where a huge number of nodes access very large objects in very long time collections, the architecture of the system might be largely determined largely by performance requirements.
Author: António Amorim

2.3.1.1.4 Muon Spectrometer Assumptions and Dependencies

AD 04: Update rate and frequency associated with the wire position in the Muon Detector
Much of the requirements depend on the accuracy with which the several TDAQ components have to know the position of the wires. Is it sufficient to know the chamber position and distortion of the chambers, or do you need to know the wire positions more accurately than average? The answers are different for LVL1, LVL2 and event filter. Hence, the amount of data depends on this specification of the needed accuracy, and the frequency of update depends on it as well. # give the total number of correction values. There will be some overhead due to the organisation into smaller units. If only chamber positions and deformations are used:# chambers 1192 # parameters per chamber O(10) 3 positions, 3 rotations, 12 chamber distortions Total # values O(few 10 k) I do not expect that information on individual wires be needed at the TDAQ level (# wires 371 k). The monitoring of the chambers works quasi-continuously. I can imagine that the corrections are produced much more frequently than they are transmitted to TDAQ. If TDAQ would get each new calibration, my guess is that the frequency would be O(hour). # For the time being we have the LVL2 and event filter in mind, plus the obvious offline. For "large" displacements also the graphics would have to be updated. For LVL1 any dependence on alignment (except for the initial one) would be painful, because it would mean to reload the decision matrices or even to recalculate them. My guess is that these matrices have to be evaluated with some tolerances, which cover the likely movements of the chambers. The decision about multi-muons would have to be adapted (di-muon trigger has low threshold and would contribute high rate, if fake double tracks are not removed). For RODs it depends on how much processing is done there?
Author: Hansl Traudl

2.3.1.1.5 Inner Detector Assumptions and Dependencies

AD 05: * Update rate and frequency associated with the Inner detector
The inner detector contains the pixel detector, the SCT, and the TRT. * There are a total of 1744 pixel and 4088 SCT modules, each of which has its own set of alignment constants. We would expect 6 degrees of freedom per module, giving 35k constants (=280 Kilobytes of doubles) to describe the detector positions at

each moment in time. Statistically, it may be possible to determine these positions about once an hour, though it may not make sense to do so. The distortions of the modules measured as part of their assembly/QA - say 10 parameters per module = another 500 Kilobytes. The latter should be constant or very slowly changing. Note that we expect the 'database view' of the alignment constants to be simply the absolute position and rotation of each module internally in the alignment processing, but that will likely evolve with time, and we don't want to build in particular assumptions in the reconstruction. In the TRT, there are around 2 million 'cells' (segment of a straw), that could be aligned and calibrated about once per hour. This involves the determination of 5 constants (residual + 4 R/t relationship parameters) for each segment, leading to a 10 Mbyte database size, potentially once per hour. Summary of datasizes SCT + pixels : 280 kilobytes TRT : 10 MBytes Both potentially could be updated about once per hour (more frequently than once per run).
Author: Richard Hawkings

2.3.1.2 DCS Assumptions and Dependencies

AD 06: Dependence from Detector Control system
The DCS SCADA environment must provide data to be used by processes computing the calibration and alignment data either online or offline.
Author: António Amorim

2.3.1.2.4 DCS interacting with Muon Spectrometer Assumptions and Dependencies

AD 07: Detector control from the Muon Detector
The MDT precision chambers for the MUON detector will be monitored by ~1200 ELMB each using the available 60 channels. These values must be monitored every 1-2 min and correpond to more than 70000 values in each full copy of the data. Update objects will contain only a small fraction of this number at each time.
Author: Helfried Buchart

2.3.1.2.5 DCS interacting with Inner Detector Assumptions and Dependencies

AD 08: Detector control from the Inner detector
FSI system data: The FSI laser interferometer alignment system (SCT only) has its own independent DAQ and data processing. A lot of raw data is processed but the output is ~500 length measurements which can be used to determine the position of reference points on the SCT structures, potentially a few times per hour. This processed information should then be stored in the standard conditions database (order 50 kilobytes/hour). Slow controls data: The SCT slow controls/DCS monitors ~< 10000 values inside and around the detector volume (temperatures, pressure, humidity). This is used mainly for control and monitoring of the SCT itself, and will be written into the slow controls database (how is this linked to the offline conditions database ?). The logging frequencies are not yet well known (some of the data is used for realtime control and has to be updated very quickly, but this will be done via interlocks). The power supplies have an additional tens of thousands of monitoring channels for the control of the various power lines to each module. Presumably the numbers for the pixels and TRT are rather similar. It is not clear to me how much of this information needs to be accessible to the offline reconstruction- at least the information on bias voltage and status for each module is needed Some of the environmental temperature information may be potentially useful for the alignment processing, and a subset of the environmental information is needed to interpret the FSI information. A small number of global inner detector quantities will also be monitored - magnetic field, temperatures and pressures. Some of these data will also be needed by the reconstruction. Beamspot monitoring: The LHC machine will tell us where they think the beamspot is (from their beam orbit monitors). At LEP this was ~ 20 words updated every two minutes. A monitoring process should also do this from tracks- update every minute ? (or even faster - plenty of statistics). This could be done with a well defined subset of events.
Author: Richard Hawkings

2.3.1.3 LVL1 Assumptions and Dependencies

AD 09: Update rate and frequency associated Level 1 trigger
The level 1 calo trigger requires a calibration data volume to be downloaded into front end modules is about 16Mb. We don't expect to update this more than once a day. Max every few hours we hope. In addition to the latest calibration, there will be the pool of predefined trigger menus (small data volume). For these the ConditionsDB should store the actual trigger menu chosen. Many different trigger menus will have simultaneous periods of validity, but only one at a time will be used.
Author: António Amorim

2.3.3 Detector and trigger configuration Assumptions and Dependencies

2.3.3.8 ConfDB Assumptions and Dependencies

AD 010: Dependence from Configuration Databases
The online and detector configuration databases must provide snapshots that will be stored in the Conditions DB and made available to the whole collaboration trough a single interface.
Author: António Amorim

2.4 Use Cases

2.4.0 EF Use Cases

UC 01:
When a new Run starts, all processing tasks receive quasi simultaneously events to be filtered and request access on conditions data[5]
Author: Francois Touchard

UC 02:
A new version of Conditions Database is available. All existing processing tasks must be modified.[5]
Author: Francois Touchard

UC 03:
A part of the detector is not functioning satisfactorily (e.g. there are some noisy or dead channels). All existing processing tasks must be notified.[5]
Author: Francois Touchard

UC 04:
The whole condition data set is read at run start, before the first event reaches the processing task.[5]
Author: Francois Touchard

2.4.1 Calibration, Alignment and Robustness Use Cases

2.4.1.7 EF Use Cases

UC 05:
A new value of calibration or alignment parameter has been calculated in the EF context. The conditions Database is updated. This update may occur at the end of the run or on the fly. The operation is logged.[5]
Author: Francois Touchard

UC 06:
A dedicated calibration run is performed and analysed by dedicated tasks running in the EF Farm. At the end of the run, the Conditions Database is updated with new parameters.[5]
Author: Francois Touchard

2.4.2 Data for event processing Use Cases

2.4.2.7 EF Use Cases

UC 07:
A subset of the conditions data is read during a run, when it is requested by the processing task to reconstruct the event.[5]
Author: Francois Touchard

2.4.3 Detector and trigger configuration Use Cases

UC 08:
The information involved is directly related to calibration, alignment and status/robustness. Its sources are most often applications that monitor the event stream, although they must incorporate information provided by the DCS databases.
Author: António Amorim

2.4.3.4 LVL2 Use Cases

UC 09:
L2 trigger state changes from RUN to FINALIZE. L2 process updates position in conditionsDB.[7]
Author: Francois Touchard

2.5 User Requirements

2.5.1 Calibration, Alignment and Robustness User Requirements

UR 01:
It must be possible to access C&A objects, by specifying not only time intervals but also intuitive data periods like data set number, run number and experiment number.
Author: António Amorim

UR 02:
It must be possible to have several updates of the C&A objects being applied to the reconstruction during continuous data-taking.
Author: António Amorim

UR 03:
It must possible to have the schema of all "C&A" classes compiled into the event reconstruction applications maybe including persistent/transient splitting.
Author: António Amorim

UR 04:
It must be possible to have multiple versions of the "same" C&A object valid for overlapping time intervals.
Author: António Amorim

UR 05:
It must be possible to define tags for given coherent sets of C&A objects (that we could call revisions) and define the revision to be used in the event reconstruction job.
Author: António Amorim

UR 06:
The interface must allow the direct and efficient association from each event "id" to the set of all C&A objects to be applied to this event once the revision is chosen. Due to the use number of events, direct associations managed by the DBMS, if at all possible, must be considered critical
Author: António Amorim

UR 07:
The C&A objects schema must be directly accessible to the processes in the lower trigger levels through a dedicated

client/server interface. Database schema to IDL conversion should be simple.
Author: António Amorim

UR 08:
Due to the huge number of processes that will access simultaneously the Condition databases, automatic cache server's processes must be implemented. The navigation from event "id" to C&A "id" must allow the cache server to know if the required objects are already included in its repository.
Author: António Amorim

UR 09:
It must be possible to include, in an homogeneous system, the configuration databases for the control data collecting applications, including also the threshold settings for minimal change in any parameter that force an update of the databases
Author: António Amorim

2.5.2 Data for event processing User Requirements

UR 010:
It must be possible to associate an "id" to each monitored channel in the detector.
Author: António Amorim

UR 011:
The different channels must be grouped into the containers for which the same type of information is stored. ex: All channels ecal_HV/barr/mod, etc, may have information on Voltage, Current and so on..
Author: António Amorim

UR 012:
The number channels in the containers and their primary id's should be able to change slowly in time as part of "in container" schema evolution.
Author: António Amorim

UR 013:
It must be possible to store very frequent changes in the monitoring values of a small part of the channels in a given container. EX: every 10s, 1% of the Muon trigger chambers HV might drift enough to force recording of the change for later efficiency studies
Author: António Amorim

UR 014:
It must be possible to browse the databases to retrieve directly the history of one or several parameters for a given channel. Examples include month to month monitoring of parameters that can change once every several seconds. This possibility must also be allowed to the outside institutions that have detector hardware responsibilities even if they do not have an extremely good connection to the experiment site
Author: António Amorim

UR 015:
For a given time it must be possible to retrieve the available channels in one container and their associated parameters.
Author: António Amorim

UR 016:
The system must cope with long periods of data taking (SLC data only grows and tends to grow a lot!) and allow splitting of data into different databases/files/servers for different periods, managing the user requests transparently thus hiding this additional complication.
Author: António Amorim

UR 017:
It must be possible to have graphical user interfaces to browse the data both plotting it and displaying it numerically in tables. These interfaces should be integrated with the data analysis tools.
Author: António Amorim

UR 018:
The SLC collects data at all times. It must aim to 7D/24h availability independently of the running periods. Robustness implies that the applications storing data in these databases must automatically store data in the backup database servers if the main servers are not available
Author: António Amorim

2.6 None Functional User Requirements