ATLAS non-event data and the conditions database

ATLAS conditions database working group
Editor: Richard Hawkings (CERN)

Abstract

This document presents a general overview of non-event data in ATLAS, in terms of the types of data, the different databases used, and the dataflow around the system. Particular attention is paid to the conditions database, and the use of it made by each of the ATLAS subdetector systems.

1 Introduction

Many types of non-event data will be used in the ATLAS data taking, reconstruction and subsequent processing. These data have many different origins, and are stored in many different types of databases. All data which is accessed offline, *i.e.* by the reconstruction or analysis software, will be stored in the conditions database, which will therefore contain raw or processed copies of much of the data from upstream databases. This note aims to provide an overview of the different types of data and where they are stored, the dataflow into the conditions database, details on the requirements of each of the subdetectors and a catalogue of the data to be produced. The interfaces between the databases are under the responsibility of the Database Coordination Board (DCB) and are described elsewhere [1]. This document builds on earlier work carried out by the ATLAS TDAQ group [2].

2 Data types

This section describes the various types of non-event data that will be produced and used in the various parts of the on and offline system. Note that only data which eventually is used in the conditions database is described here, other types of data (*e.g.* production drawings, acceptance test results) certainly exist and are stored in ATLAS databases, but they are outside the scope of this note. The relationship between the different types of data, and the sources and consumers of this data, is shown in Figure 1.

2.1 Setup data

Setup, or configuration, data is used in all parts of the online system, including the subdetectors, TDAQ, event filter and DCS system. This data consists of all the programs (?), parameters and configuration information necessary to set up and operate the experiment. Much of this data will be input and maintained manually, using various special-purpose tools. Some will be input automatically from the technical coordination (TC) production and installation databases, for example the cabling database giving information on the connections between detectors, processing elements *etc.*

This data will generally be stored by version (*e.g.* sets of configuration files for a particular group of runs, rather than timestamp). A mechanism is needed to record and retrieve the setup used for a particular data-taking run. This could be achieved by copying a 'snap-shot' of the data used for each run into the conditions database, and enforcing an interval of validity (IoV).

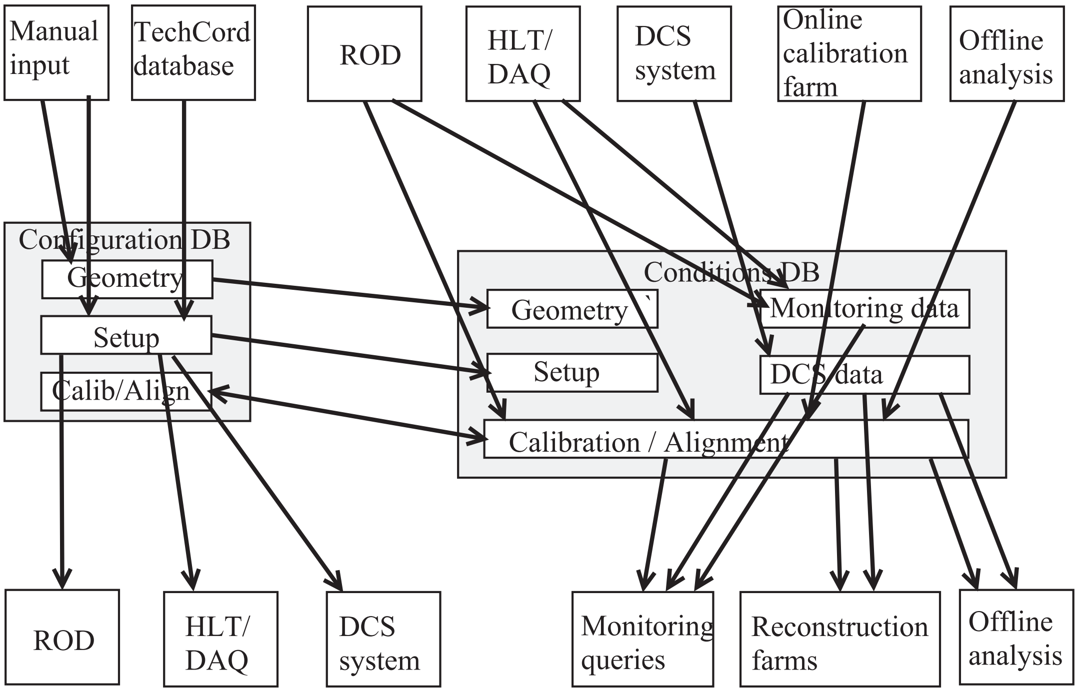

Figure 1: Sources, types and consumers of data in the ATLAS configuration and conditions databases. The top row represents the sources, the middle row the databases themselves with the different types of stored data, and the bottom row the consumers of data. Arrows show the direction of dataflow from one item to another.

2.2 Geometry data

Geometry data gives the physical geometry parameters and location of the components of ATLAS, including survey information. This data is not expected to change, except during the installation phase and when major interventions are performed in the experiment. The main consumers are the high level trigger, the offline reconstruction and detector simulation. The data is generated mainly 'by hand' by subdetector experts, and will be stored directly in the conditions database. For the real detector, a time-stamp method of retrieval is appropriate, whereas for simulated geometries (where different from the real experiment), non time-based versioning may be more useful.

The TC databases also contain a certain amount of geometry data, but this is not expected to detailed enough or in a form useful enough to allow automatic filling of the geometry information from the TC databases.

2.3 DCS data

DCS data is produced by the detector control system (DCS), *i.e.* slow controls. This is a distributed PVSS system where data is stored on the backend nodes (subsystem control stations) using native PVSS tools, and periodically archived to an external system, the conditions database. The data consists of a mixture of digitised analogue readings (*e.g.* temperatures, pressures, high voltages) and 'occurrences' or state transitions, (*e.g.* systems changing mode, switching on or off). This data is accessed by timestamp and is not synchronous with data-taking. DCS data will come from the subdetectors themselves, infrastructure monitoring *e.g.* rack and cavern temperatures) and also from the LHC accelerator. It is used by the high level trigger/event filter farms, offline reconstruction and also for shift crew monitoring. Historical lookup of old DCS data will be done via the conditions database.

Although the data will be stored in the conditions database, it does not fit naturally within the IoV interface model—consisting rather of a stream of 'instantaneous' values that either remain valid or need interpolating until the next stored value. The concept of versioning is unlikely to be useful—DCS data will not be updated with 'better' versions offline. For these reasons, a second type of IoV service interface may be useful for DCS data. For efficiency reasons, it may also be useful to store the data objects as 'chunks' of *e.g.* one hour's worth of data, and have an intermediate layer retrieve these chunks from the conditions database and interpolate to a particular time.

A second data path for DCS data exists, from the DCS to the online DAQ system, via the 'DDC' interface. This is primarily for DCS-DAQ communications and control functionality, and will not be used to archive DCS data via the DAQ system.

2.4 Monitoring data

Monitoring data is data derived from the event stream at various points to monitor the functioning and performance of system components. Monitoring will be performed in the subdetector RODs, in the level 1 and high level trigger, and in the event filter (what about the prompt reconstruction?). The primary use of monitoring data is to verify the data quality and allow the shift crew and experts to take remedial action. Monitoring data (typically in the form of histograms) will also be logged in the conditions database for subsequent retrieval and study. This data is similar to DCS data, in the sense that it has a time stamp but no versioning (unless some of the same tools are used to monitor data quality in re-reconstruction passes.) The main difference is that monitoring data is derived from the event stream and DCS from independent sources, continuing in the absence of data-taking.

The same monitoring tasks may be used to generate calibration and alignment information 'online', to be fed back to the high level trigger and event filter for use in processing the next events (*e.g.* monitoring of the beamspot position), and forward to the conditions database for use in prompt and subsequent reconstruction *e.g.* online determination of calorimeter calibration and tracking alignment constants.

Considerable work has already been done gathering the requirements and data sizes for monitoring data from the various ATLAS subdetectors. The results are available in [3].

2.5 Calibration and alignment data

Calibration and alignment data includes all of the 'constants' necessary to run reconstruction and offline analysis, apart from fixed numbers like the detector geometry. The offline reconstruction software only accesses the conditions database, not the online configuration or TC databases. Therefore, all data required offline has to be available in the conditions database. The data naturally comes with two types of 'tags'—timestamps giving the interval of validity over which it is valid, and version—several versions of the conditions data may be available for the same event data, corresponding to improved understanding and more refined calibration algorithms.

Conditions data is produced in many places—explicitly by online monitoring and calibration tasks, running in the RODs, high level trigger, event filter and possibly special purpose farms, analysing subsets of the event data, and implicitly by other databases filling the conditions database with an archive of their current information (DCS, configuration and geometry databases). Updates will also arrive asynchronously from offline processing tasks, either run by hand or automatically, typically to provide improved calibration for subsequent reprocessing of the event data.

Conditions data will be required primarily by the prompt reconstruction and offline computing farms, but some data may also be 'fed back' to the high level trigger system. Data will also be required to answer queries generated by shift crew operators and subdetector experts. With several versions of the same data being present, careful versioning control will be required.

3 Databases

This section discusses the online configuration database and particular the conditions database. More details about the other databases in ATLAS, and the relationships between them, are given in [1].

3.1 Configuration database

The primary role of the configuration database is to store everything that is needed for the current (and next) run of the ATLAS DAQ. This includes all the relevant setup data, plus a subset of the currently valid conditions data. The configuration database acts as a source for these conditions data, and uses the conditions database as an archive.

A prototype configurations database exists as part of the DAQ-1 project, implemented in the OKS system (?). need more details, technology stable? future evolution ?

3.2 Conditions database

The conditions database stores everything needed for offline reconstruction and analysis of the ATLAS event data stream, *i.e.* all of the calibration and alignment data, geometry information and setup data. It is also used as the archive for DCS and monitoring data, some of which may also be needed in offline analysis.

The primary access to conditions data is through the interval of validity (IoV) interface. An API (application program interface) has been developed by IT division, and several implementations have been proposed or actually developed. The currently relevant ones are a mySQL implementation by the ATLAS Lisbon TDAQ group, and an Oracle implementation by IT. The mySQL implementation is currently in use in several ATLAS test beam environments.

An important concept is that the API and IoV service just store ASCII strings, which can act *e.g.* as XML strings encoding arbitrary pieces of information. The API service provides the string corresponding to the requested timestamp and version tag. The string can be used either to store the actual time varying information (if the size of the information is small) or to store a persistent pointer to some other object, located *e.g.* in another database, perhaps in a different technology. In this way, the IoV service can be used as a 'gateway' to many types of database information stored in mySQL databases, POOL ROOT files, simple text files *etc.*. The conditions database can then be seen as a heterogeneous structure, incorporating different database technologies but accessed through a common interface.

The IT API to the conditions database has some limitations, and is not well suited to *e.g.* the storage of DCS-type data. Some developments are needed in this area.

4 Dataflow

This section presents a view of the flow of non-event data into the conditions database. The dataflow is shown schematically in Figure 1. Some systems (*e.g.* the RODs and HLT/DAQ) act as both sources and consumers of different types of non-event data. Most of the dataflow arrows are self explanatory, but several are not:

- Several types of data (geometry, setup and calibration and alignment) are present in both the configuration and conditions databases. In these cases, the direction of dataflow has to be defined. For geometry and setup data, it is assumed that the configuration acts as the 'master', with data being archived in the conditions database as necessary[1].
- For calibration and alignment information, the flow of data is less obvious. Some calibration and alignment information is needed in the online system—should this be stored in the configuration database and preloaded from the conditions database? How should information updated in the course of a run be propagated to the configuration database and thence to the subdetector hardware?

[1] Geometries used only for Monte Carlo simulation will also be present in the conditions database, but will not be loaded from the configuration database.

5 Subdetector specifics

Need to fill in information from the subdetectors, on how they fit into this scheme. Data volume, any special requirements?

5.1 Inner detector

5.2 LAr calorimeters

5.3 Tile calorimeter

5.4 Muon spectrometer

5.5 Trigger / DAQ

6 Summary and open issues

References

[1] ATLAS databases—requirements and interfaces, ATLAS database coordination board, R. Hawkings, D. Malon, K. Pommes, I. Soloviev, document in preparation.

[2] ATLAS TDAQ DDD - Database definition document (draft), A. Amorin. M. Caprini, D. Klose, J. Lima, L. Pedro, I. Soloviev, F. Touchard, March 2003.

[3] Online monitoring requirements, ATLAS monitoring working group, F. Touchard, H.P. Beck, B. Di Girolamo, S. Kolos, M. Landon, July 2003.