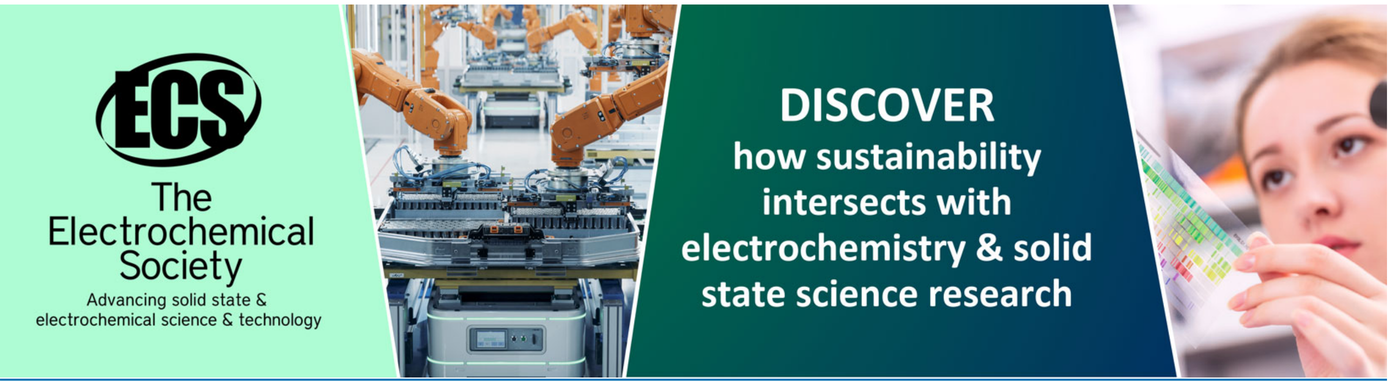
Journal of Physics: Conference Series

The ATLAS detector digitization project for 2009 data taking

To cite this article: J D Chapman *et al* 2010 *J. Phys.: Conf. Ser.* **219** 032031



Journal of Physics: Conference Series **219** (2010) 032031 doi:10.1088/1742-6596/219/3/032031

The ATLAS Detector Digitization Project for 2009 data taking

J D Chapman[1], K Assamagan [2], P Calafiura [3], D Chakraborty [4], D Costanzo [5], A Dell'Acqua [6], A Di Simone [7], G Lima [4], Z Marshall [8], A Rimoldi [9], I Ueda [10], S Vahsen [3], D Wright [11], Y Zhou [12]

[1] Cavendish Laboratory, JJ Thomson Avenue, Cambridge, CB3 0HE, United Kingdom
[2] Brookhaven National Laboratory, PO BOX 5000 Upton, NY 11973, United States of America
[3] Lawrence Berkeley National Laboratory and University of California, Physics Division, MS50B-6227, 1 Cyclotron Road, Berkeley, CA 94720, United States of America
[4] Physics Department, Northern Illinois University, DeKalb, IL 60115, United States of America
[5] Dept of Physics and Astronomy, University of Sheffield, Hicks Building, Hounsfield Road, Sheffield, S3 7RH, United Kingdom
[6] CERN, CH-1211 Geneva 23, Switzerland
[7] INFN Roma Tor Vergata and Università di Roma Tor Vergata, Dipartimento di Fisica, Via della Ricerca Scientifica, IT-00133 Roma, Italy
[8] California Institute of Technology, Physics Department, 103-33, Pasadena, CA 91125, United States of America and Columbia University, Nevis Laboratory, 136 So. Broadway, Irvington, NY 10533, United States of America
[9] INFN Pavia and Università di Pavia, Dipartimento di Fisica Nucleare e Teorica, Via Bassi 6, IT-27100 Pavia, Italy
[10] The University of Tokyo, International Center for Elementary Particle Physics, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[11] SLAC National Accelerator Laboratory, Stanford, CA 94309, United States of America
[12] Insitute of Physics, Academia Sinica, TW - Taipei 11529, Taiwan

E-mail: `chapman@hep.phy.cam.ac.uk`

Abstract. The ATLAS digitization project is steered by a top-level PYTHON digitization package which ensures uniform and consistent configuration across the sub-detectors. The properties of the digitization algorithms were tuned to reproduce the detector response seen in lab tests, test beam data and cosmic ray running. Dead channels and noise rates are read from database tables to reproduce conditions seen in a particular run. The digits are then persistified as Raw Data Objects with or without intermediate simulation of the exact data acquisition format depending on the detector type. Emphasis is put on the description of the digitization project configuration, its flexibility in handling events for processing and in the global detector configuration. Other options available, including detector noise simulation, random number service, metadata and details of pile-up background events to be overlaid, are also described. The LHC beam bunch spacing is also configurable, as well as the number of bunch crossings to overlay and the default detector conditions (including noisy channels, dead electronics associated with each detector layout). Cavern background calculation, beam halo and beam gas treatment and pile-up with real data are also part of this report.

1. Introduction

The ATLAS Experiment[1] is a general-purpose experiment at the Large Hadron Collider at CERN[2]. ATLAS has been designed to collect data from 14 TeV proton-proton collisions as well as 5.5 TeV per nucleon pair heavy ion (Pb-Pb) collisions. During proton-proton collisions at the design luminosity of 10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ proton bunches will cross every 25 ns. This will yield an average of 23 collisions per bunch crossing.

Digitization sits between the GEANT4 [3][4] simulation[5] and the reconstruction[6] in the the ATLAS production chain. Large-scale ATLAS production is done on the World-wide LHC Computing Grid (Grid)[7]. In ATLAS, digitization refers to the process of converting GEANT4 simulated hits in active volumes of the detector to Raw Data Objects (RDOs), which act as an input to the reconstruction. This includes the propagation of charge to readout electrodes, the front end electronics simulation and the conversion of these responses into RDOs. The contributions of minimum bias, cavern background, beam halo and beam gas interactions (collectively known as pile-up) can be simulated during the digitization step. This is discussed further in Section 3.1. Level one (hardware-based) trigger simulation can also be performed during the digitization step. Alternatively, the level one trigger can be simulated during the reconstruction step using the trigger inputs simulated during digitization.

The real ATLAS detector writes out data in various bytestream formats; before this can be input into the reconstruction it must be converted into the appropriate RDO format. Some digitization algorithms simulate this step, others produce RDOs directly. In all cases algorithms exist to convert RDOs into bytestream format and vice-versa. The formats are similar, the main difference is that RDOs have separate transient and persistent representations in order to allow more efficient storage.

2. Core Digitization Framework

The core digitization framework is entirely PYTHON based. This PYTHON code controls the C++ based code which carries out the actual digitization process and organises the pile-up of background Hits. It is highly configurable, while ensuring sensible default values and resolves any conflicting settings. The inputs to the digitization are Hit Objects produced by the ATLAS GEANT4 simulation[5]. They are normally read in from one or more POOL[8][9] files. The outputs of the digitization are RDOs, which can be written out to a POOL file if required. Digitization can be run stand-alone or as part of simulation or reconstruction jobs.

2.1. Digitization Configuration

The digitization steering package exists entirely in the PYTHON layer and configures how the digitization will be performed before the event loop starts. This configuration is highly flexible, but also ensures that sensible default values are given for each configurable property of the job.

In the configuration of a digitization jobs, the user may specify the number of events to digitize, the number of leading events to skip in the input file, the input hit file(s), and the output file. Digitization and the persistification of RDOs may be enabled or disabled by sub-detector. In order to ensure consistency, the detector layout version is, by default, read from the input Hit file metadata. However, the user is free to override the default if required.

Digitization options also include the following:

Detector Noise Simulation: Detector noise simulation can be turned off in the inner detector, calorimeter or muon spectrometer or any combination thereof. This is useful for data overlay jobs where noise is taken from real data events (see Section 3.2) and for studies using a noise-free detector.

Random Number Services: The type of random number engine to be used in all digitization algorithms can be specified (Ranlux64, the default, or Ranecu[10]). Each algorithm has one or more random number streams. Random number seeds can be initialized from a text file or set in job options. The user may also specify an offset from the default values of the seeds, to be used in all streams.

Metadata: By default, input Hit file metadata is used to configure the physics list (used to set the sampling fraction of the calorimeters) and detector layout version, but if necessary this can be overridden.

Pile-up Background Events: The overlay of minimum bias, cavern background, beam gas and beam halo events can all be configured separately. In each case the mean number of events (if any) per bunch crossing to be overlaid and a collection of files containing the events to be overlaid onto the signal events can be specified.

Beam Properties: The initial and final bunch crossings to simulate before and after the main event and the bunch spacing in nanoseconds can all be configured. Another flag is used to specify if the events being digitized are due to cosmic rays or beam interactions. (The configuration of some sub-detector readouts is different in the case of cosmic runs.)

Detector Conditions: There is a default detector conditions (including, e.g., dead electronics and noisy channels) version associated with each detector layout. A non-default version may be specified for use in digitization. The conditions may be specified by sub-detector and may be used to recreate the conditions of a particular data taking period.

After a sanity check to make sure that at least one sub-detector has been switched on, the input and output streams are initialized. The ATLAS GeoModel (the ATLAS detector geometry) is initialized using the detector layout and conditions versions read from the Hit file metadata or specified by the user. Setting the detector layout version to be different from that used in the simulation is possible, but considered to be an expert action not for normal production. Once the detector description is in place, the magnetic field service is initialized. The magnetic field is required for digitization, because it affects charge propagation from the active regions to the readout surfaces.

At this point, caches for pile-up events are created and configured with the appropriate collection of Hit files as well as the number of events to be overlaid per bunch crossing. These caches are controlled by an overall pile-up manager service. A second pile-up service is created to hold information about the time window within which interactions can affect the response recorded by each sub-detector. During the initialization stage this information can be combined with the bunch spacing to calculate the number of bunch crossings which should be simulated for each sub-detector for each event.

Subsequently, the sub-detector digitization algorithms are configured and added to the sequence of algorithms to be run in the job. The output stream is configured with a list of the collections of RDOs, hits and truth information which are to be written out. Digitization algorithms exist for the following sub-detectors:

Inner Detector: Beam Conditions Monitor, Pixel Detector, Semiconductor Tracker, and Transition Radiation Tracker.

Calorimeter: Liquid Argon and Tile Calorimeters. Separate algorithms also exist to simulate the formation of trigger towers in the calorimeters, which serve as inputs to the level one trigger.

Muon Spectrometer: Cathode Strip Chambers, Monitored Drift Tubes, Resistive Plate Chambers and Thin Gap Chambers.

A digitization algorithm also exists for LUCID (situated at ± 17 m from the interaction point), although currently this is not included in default production. Algorithms for other Forward Detectors will be added as required. If requested, the level one trigger simulators are added to the algorithm sequence, provided that the digitization of the relevant parts of ATLAS have been switched on. Currently, the default mode of Monte Carlo production is to run the level one trigger simulation as part of the reconstruction step rather than as part of the digitization step.

As the digitization algorithm for each sub-detector is configured, the names and seeds for the random number streams it requires are added to a list. If seeds are to be read in from a file, the default list of stream names and their seeds are replaced by the file contents. Once all algorithms have been configured, the list is used to configure the random number service. Separate random number streams are used for each sub-detector digitization algorithm and give the same result independent of what is used for the other sub-detectors[1].

Much of the job configuration information, along with the detector layout version, is written to the output file as digitization metadata. The run number provided in the simulation metadata is used to establish a validity range for the digitization metadata corresponding to the current run only. At this point the digitization job is fully configured and the event loop begins.

2.2. Detector Defect Simulation

The strategy for detector defect simulation varies between sub-systems. A common approach is to add defects recorded in a conditions database. Changes to these defects over time are handled via conditions database tags. Defects include deformations of readout volumes (e.g. Silicon wafers in the Inner Detector), dead channels and noisy channels. Detector misalignments are set during the Geant4 simulation step by the Geometry database tags used. Generally dead-channels are set to reflect conditions in the actual detector, although some sub-systems also maintain the ability to generate random dead and busy channels within their sub-detector. In some cases dead channels are blocked during the digitization step, so no RDOs are produced for those channels, other sub-systems simply veto RDOs from channels flagged as dead or busy early on during the reconstruction step instead.

The digitization code allows random noise and cross-talk to be switched on and off in the Inner Detector, Calorimeters and Muon Spectrometer separately. As explained in Section 2.1, the random number generators and seeds used are controlled to ensure reliable performance.

3. Event Pile-up

A triggered bunch-crossing in ATLAS will contain many inelastic, non-diffractive proton-proton interactions in addition to the hard scattering which triggers the detector readout. The effects of beam gas and beam halo interactions, as well as detector response to long-lived particles, are also important. Long signal integration times mean that most sub-detector responses are affected by interactions from neighbouring bunch crossings as well. However, it would take too long to simulate each event in such detail at the Geant4 stage.

3.1. Solution 1: Event Overlay During Digitization

One solution is to simulate the hard-scattering interactions, minimum bias interactions, beam halo events, beam gas events and cavern background events separately. In the digitization stage the Geant4 hits are overlaid to simulate the required number of bunch-crossings at the required luminosity and bunch spacing. The resulting collection of Geant4 hits is then passed to the digitization algorithms.

Minimum bias events are generated using Pythia [11]. The mean number of minimum-bias interactions per bunch crossing is 23 at the design luminosity of 10^{34} cm^{-2}s^{-1} with 25 ns bunch spacing. In the simulation, it depends linearly on luminosity and bunch spacing. The number

[1] Here "digitization algorithm" does not include the calorimeter trigger tower simulation algorithms, which require the corresponding calorimeter digitization to be performed. Similarly, the level one trigger simulation requires the simulation and digitization of the expected trigger inputs to give meaningful results.

of minimum-bias interactions per bunch crossing is Poisson-distributed. Thus, some bunch crossings may have many more than the average number of interactions. Cavern background events are generated using a GEANT3/GCALOR-based program which generates particle fluxes in an envelope around the muon spectrometer. Beam gas includes the residual Hydrogen, Oxygen, and Carbon gasses in the ATLAS beam pipe. Events are generated using Hijing[12]. Beam halo is the background resulting from interactions between the beam and upstream accelerator elements. The flux from upstream (in the tunnel and collimators) is provided by the LHC Machine Division[13][14].

3.2. Solution 2: Data Overlay

Minimum bias, cavern background, beam halo and beam gas backgrounds can be obtained from the same "zero bias" trigger used to understand detector electronic noise. Simulated hard-scattering interactions are then digitized without simulating detector noise. The resulting RDOs are then overlaid on top of real data events which passed "zero bias" triggers. In this way the detector noise and background interactions will match those seen in the real experiment by construction.

The "zero bias" trigger data needed for this type of event overlay can be selected at random from the filled-bunch crossings. The sub-detectors should be read out with as little zero-suppression as is possible and with the Higher Level Trigger in "pass-through" mode. Bunch-by-bunch luminosity information can be used to correctly weight the event sample for pile-up studies. In principle one needs as many zero bias events as Monte Carlo generated events, but in practice zero bias events can be reused with independent Monte Carlo sets. During data taking, zero bias events will be sampled at all times, because detector and cavern conditions are likely to vary with time. The rate must be such that even short luminosity blocks have sufficient statistics: it is likely to be on the order of $1 - 2$ Hz.

4. Performance

There has been a lot of work on improving the performance of the digitization in terms of CPU and memory usage. Digitization of a single interaction (without pile-up) takes $20 - 30$ kSI2K seconds on average[5]. The mean time taken per event increases linearly with the number of pile-up events overlaid on the hard-scatter event. For 10^{33} cm^{-2}s^{-1} luminosity pile-up the mean time per event is 2.3 times that for a single event; this increases to 160 times for 10^{34} cm^{-2}s^{-1} luminosity pile-up events.

Memory leaks in the digitization have been reduced to only a few kB even for 3.5×10^{33} cm^{-2}s^{-1} luminosity pile-up jobs. This has been achieved by nightly monitoring of jobs using Valgrind and Hephaestus[15]. The size in memory of pile-up digitization jobs does increase over time due to the caching of background events. There has been recent progress towards the goal of reducing the memory footprint by reducing the size of hit containers, but these changes are not available for use in production jobs at the time of writing. The goal is to reduce virtual memory usage for 10^{34} cm^{-2}s^{-1} luminosity pile-up jobs from 3.4 GB to 2 GB before large scale Monte Carlo production at this luminosity is required. Simulation of pile-up with luminosities up to 3.5×10^{33} cm^{-2}s^{-1} can be run on standard Grid queues. Recently we have been able to run 10^{34} cm^{-2}s^{-1} pile-up in special queues on the Grid.

5. Conclusion

We have presented the current status of the ATLAS Digitization, part of the ATLAS Monte Carlo simulation project. The software package has been prepared for first collision data in 2009. The digitization has been made configurable to cope with uncertainty in machine performance, detector conditions, and cavern conditions. A description of the current event pile-up simulation has been given as well as the longer term goal of overlaying "zero-bias" trigger data on top of digitized hard-scattering interactions. An assessment of the performance of the current digitization software has also been given.

References

[1] The ATLAS Collaboration, Aad G *et al* 2008 *JINST* **3** s08003
[2] Evans L 2007 *New J. Phys.* **9** 335
[3] Rimoldi A *et al* 2008 *ATL-SOFT-PUB-2008-002*
[4] Rimoldi A *et al* 2008 *ATL-SOFT-PUB-2008-004*
[5] Marshall Z 2009 *CHEP09 Proceedings*
[6] Arshak T 2009 *CHEP09 Proceedings*
[7] Bird I (ed) *et al* 2005 *CERN-LHCC-2005-024*
[8] Dullmann D *et al* 2004 *Computing in High Energy and Nuclear Physics 2004 (CHEP2004)*
[9] Chytracek R *et al* 2005 *IEEE Trans. Nucl. Sci.* **52** 2827-2831
[10] James F 1990 *Comp. Phys. Comm.* **60** 329-344
[11] Sjstrand T and Mrenna S and Skands P 2006 *JHEP* **05** 026
[12] Gyulassy M and Wang X N 1994 *Comput. Phys. Commun.* **83** 307
[13] Azhgirey I and Baishev I and Potter K M and Talanov V 2003 *LHC Project Note* **324**
[14] Azhgirey I and Baishev I and Potter K M and Talanov V 2003 *LHC Project Report* **567**
[15] Calafiura P 2009 *CHEP09 Proceedings*