The Status of the Simulation Project for the ATLAS Experiment in view of the LHC startup

I. Ueda1, A. Dell'Acqua2, M. Gallas3, A. Di Simone3, Z. Marshall4, J. Boudreau5, Y. Zhou4, V. Tsulaia5, J. Chapman6, A. Rimoldi8, M. Asai9, D. H. Wright9, J. G. Rocha de Lima10

1The University of Tokyo, International Center for Elementary Particle Physics, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
2CERN, CH-1211 Geneva 23, Switzerland
3INFN Roma Tor Vergata and Università di Roma Tor Vergata, Dipartimento di Fisica, via della Ricerca Scientifica, IT-00133 Roma, Italy
4California Institute of Technology, Physics Department, 103-33, Pasadena, CA 91125, United States of America
Columbia University, Nevis Laboratory, 136 So. Broadway, Irvington, NY 10533, United States of America
5University of Pittsburgh, Department of Physics and Astronomy, 3941 O’Hara Street, Pittsburgh, PA 15260, United States of America
6Institute of Physics, Academia Sinica, TW - Taipei 11529, Taiwan
7Cavendish Laboratory, JJ Thomson Avenue, Cambridge, CB3 0HE, UK
8INFN Pavia and Università di Pavia, Dipartimento di Fisica Nucleare e Teorica, Via Bassi 6, IT-27100 Pavia, Italy
9SLAC National Accelerator Laboratory, Stanford, CA 94309, United States of America
10Physics Dept, Northern Illinois University, Dekalb IL 60115, United States of America

ueda@icepp.s.u-tokyo.ac.jp

Abstract. The Simulation suite for ATLAS is in a mature phase, ready to cope with the challenge of the 2009 data. The simulation framework, which is integrated to the ATLAS framework (Athena) offers a set of pre-configured applications for the full simulation of ATLAS, combined test beam setups, cosmic ray setups and old standalone test-beams. Each detector component has been carefully described in detail and monitored for performance. The few pieces of the apparatus (forward and very forward detectors), inert material and services (toroid supports, support rails, detector feet) that are still missing are about to be integrated in the current simulation suite. Detailed descriptions of the ideal and real geometries for each ATLAS subcomponent allow optimization studies and validation. Small scale productions are monitored daily through a set of tests for different samples of physics events, and large scale productions on the Grid verify the robustness of the implementation as well as possible errors only visible on large statistics. The conditions used in the simulation process are now stored in the output file as metadata, and can be utilized to process the data properly. A fast shower
simulation suite has also been developed in ATLAS and performance comparisons are part of the overall evaluation.

1. Introduction
The Large Hadron Collider (LHC) at CERN was inaugurated in 2008 and the first collisions of protons are expected in 2009. ATLAS [1], one of the general-purpose experiments at LHC, began operation in 2008. In order to study detector response and the effectiveness of proposed physics analysis strategies, a detailed simulation has been implemented that produces data in a format identical to that of the real detector output.

The simulation is carried out through three main steps. The first step is event generation, where particle collision events and background events are simulated. The second step is the simulation of the interaction of the particles passing through the detector with the detector material and the recording of the particles' energy deposits in the sensitive detector volumes. The third step is the digitization of the deposits to simulate the detector response including the front-end electronics. In addition, pile-up process that overlays multiple events from particle collisions and background can be included before the digitization step. The details of event generation step are discussed in a separate article in this volume [2]. The details of digitization and pile-up are discussed in [3], and not discussed in this article.

The description of the complex geometry of ATLAS is decoupled from the simulation and implemented in a way not specific to the simulation. All of the detector simulation, digitization and reconstruction applications use the same geometry. A variety of detector setups have been prepared and the user can choose an appropriate geometry at runtime. The recent implementation of the missing pieces of the apparatus and improvements in the simulation of inert materials give more accuracy to the detector simulation.

Validation of the ATLAS simulation software is carried out at a few stages. The software before release is tested on a daily basis to ensure the stability and to find bugs and incompatibilities between modules. Computing performance of the software is validated made with every release. A small scale production is made and the results are validated before a large scale production. The validation activities are discussed in detail in [4].

2. The Atlas Simulation Framework
The simulation framework, integrated in the ATLAS software framework named Athena [5], runs based on dynamic loading and action on demand, keeping jobs as light as possible in memory. The framework is written in C++ and a Python interface adds interactivity and also flexibility for run-time configuration using job options scripts written in Python [6]. The three steps can be run either individually as separate jobs, or sequentially in a single job with a chain of job options scripts configured in the Athena framework.

The output of the simulation chain is recorded in an object-based format, and a set of converters is provided to translate the simulated data into a format identical to the output of the ATLAS data acquisition system and vice versa. Thus, both the simulated data and the real data from the detector can be processed by the same ATLAS trigger and reconstruction software. The metadata, a set of options used in the simulation step, are also stored in the output file and used in the subsequent digitization step to automatically configure the job with the consistent options with the simulation step.

3. Event Generation Software
Various event generators are available within the ATLAS simulation suite: the single particle generator based on G4ParticleGun is used for basic sub-detector response studies, a number of physics event generators to simulate p-p and heavy ion collisions are introduced as external software and
interfaced to the Atlas software framework [2], the beam halo, beam gas, cavern background and cosmic ray generators are used to simulate the background to be overlaid with physics events [3]. The Athena generator interfaces allow all relevant parameters to be included at runtime, permitting a fixed software release to be used to produce different physics configurations. The generated events can be either passed to the detector simulation directly, or saved in a file in standard HepMC format so that simulation jobs can be run separately with different configuration or with different software releases.

4. Detector Simulation Software
Different detector simulations can be run using the same ATLAS simulation framework, sharing the same environment and configuration interface. The Geant4 toolkit [7] is used as the baseline detector simulation in ATLAS. The Geant4 version used in the latest ATLAS software (release 15) is 9.2. Although the baseline is the full simulation that simulates the behavior of the particles passing through the detector in detail, a variety of fast simulations are also available.

4.1. Detector Simulation Configuration
The Framework for ATLAS Detector Simulation (FADS) wraps several Geant4 classes in order to allow selection and configuration without recompilation of any libraries. The wrappers serve a dual purpose: first, they ease translations between the Geant4 and Athena standards of geometry, hits, and particle storage. Second, FADS can catalogue the options provided in libraries and loads only those options selected by the user. Thus the memory usage is minimized for the desired simulation configuration while still providing all possibilities without any recompilation. For example, while the whole ATLAS detector is implemented in the software, the user can choose a set of subsystems to be used in the simulation. This is useful especially for the studies of the response of the subsystems individually. The configuration is setup using Python scripts, and once the configuration is complete, FADS objects are translated into their Geant4 equivalents and loaded.

The configuration of the detector includes misalignments and distortions. Any of the detector elements can be shifted and rotated at run time, so that the single implementation of the detector can be used for simulation with the detector in nominal shape and also for simulation with the detector in a realistic shape. Also, the magnetic field may be enabled and the field map may be selected. Various “physics lists”, which define the physics processes to be used in the Geant4 detector simulation, are prepared and the user can select one of them. The default physics list is determined after comparing the results of the simulation with the different physics lists. The parameters to control the treatment of the particles such as range cut, step limitation, neutron cut may be set as well. With the recent development, the configuration of the choice of the stepping method that controls the transportation of particles, and stepping parameters as a function of the initial particle type, energy, and position within the detector is made possible. Such a configuration can allow more careful stepping of muons in the calorimetry without degrading the total performance of the simulation significantly.

4.2. Treatment of Events
The event generation mode (reading external events, generating events from an external generator, or generating single particles on-the-fly) is determined at runtime. Each of the generated events stored in HepMC format is translated into the standard Geant4 event format prior to the detector simulation by FADS, and each particle is propagated through the ATLAS detector by Geant4. A filtering can be applied to select only certain particles from the generated events to process in the simulation. The typical filtering is made on the transverse momentum and the pseudo rapidity of the particles. The vertex position can be smeared according to the nominal beam size, and possibly be shifted if the user requires. Although the events are generated assuming that the proton beams are aligned with the axis of the detector, the simulation framework provide a possibility to rotate the event shape in order to make it consistent with the small (but non-zero) crossing angle of the proton beams.
While the particles are being propagated through the ATLAS detector, the results of its interaction with the materials in the sensitive region of the detector are recorded as “hits,” containing the total energy deposition, position, and time. The “hits” are written into the output file of the simulation as well as the "truth" information, containing the true trajectories of the particles and decays or conversions of certain particles. The information of the generated particles read into the simulation, including not only the final state stable particles but also all the history from the parton level information, are also recorded as "truth" information in all cases even for those particles that are not passed onto the simulation.

4.3. Fast Simulations
The detector simulation with Geant4 toolkit is used as a baseline especially for studies with detailed detector responses. However, because of the complicated detector geometry and detailed physics description used in the ATLAS Geant4 simulation, it is impossible to achieve the statistics required for physics studies with the available computing resources. Therefore, a number of fast simulations with different level of details have been developed to be used for physics studies that require large statistics. Performance comparisons among the fast simulations and the full simulation have been made as a part of the overall evaluation. The details are discussed in a separate article [4].

5. Detector Description
Detector description is a separate package which is used by the detector simulation, digitization and the other steps of the software chain such as reconstruction. Thus, it is implemented independently of and in a way not specific to the simulation. The basic parameters to describe the physical construction of the detector and time-dependent alignment constants are stored in a database from which a transient geometry representation of the detector is built using the GeoModel toolkit [8]. The geometry is then converted into Geant4 geometry during the initialization phase of the detector simulation.

A number of different detector setups are implemented and available in the package, such as the ideal detector with nominal geometry parameters, the realistic detector with misalignments and distortions, the detector arrangement during the commissioning period with displaced sub-detector parts, the cosmic-ray setups with the cavern under 100m of rock, the combined test beam setups, standalone test-beam setups. Any of the pre-defined detector setups can be chosen with a simple option in the configuration script at run-time. Having a simulation with the detailed descriptions of the ideal and the realistic geometries for each ATLAS subcomponent makes it possible to optimize and validate the reconstruction and analysis software. Results of the simulation with the cosmic-ray setup are utilized for the commissioning of the ATLAS apparatus. The test-beam setups are the basis of performance studies and physics validation of the detector simulation. In the data-taking period of the combined test-beam, the detector layout had evolved rather frequently. Simulation of these different and time-dependent layouts can be handled by specifying the run number.

Each detector component has been carefully described in all details and its memory consumption is monitored. The description of the full ATLAS detector is almost complete and enough to produce meaningful simulation data. Although the forward and very forward detectors are being implemented, they are not usually used during a standard simulation job because particles in forward regions are filtered out in order to reduce the simulation time. Also, inert materials such as supports of the toroidal magnets, support rails, detector feet are being integrated.

6. Validation of the software and Large Scale Productions
In parallel to the development of the simulation software, validation of the software is performed on a daily basis with a set of small scale validation jobs for different samples of single-particle and physics events. The aim of the validation is to spot as soon as possible any non-optimal performance, internal inconsistency, or even inaccurate description of the detectors or physical processes. The validation is performed typically with 300 single particle events for each of electron, muon and pion, and with 10 to 30 physics events for each type of events, the number of events depending on the purpose of each
validation job. As soon as a new release of the ATLAS software is prepared, a routine validation of
CPU time and memory consumptions is performed with a set of similar validation jobs but with a
higher statistics and wider variety of types of physics events, typically with 300 single particle events
for each particle type as above and with 200 events of each type of events. A small scale production of
typically 10,000 events for each type of events is then performed for the validation of the simulated
detector responses.

Once the release is validated, large scale productions start in order to provide data for physics
analyses. Several hundred millions of events have been produced on the LHC Computing Grid in the
last year, which proved the robustness of the ATLAS simulation implementation, and production of
even more simulated data is planned. The validation and the computing performance studies of the
ATLAS simulation are described in detail in [4].

7. Conclusions
The simulation suite for ATLAS is in a mature phase with flexibility and robustness. Various event
generators, the Geant4 toolkit and ATLAS specific fast simulations are integrated into the framework,
and ready to use with simple job options scripts. A number of different detector setups are prepared for
different types of studies. The non-critical but missing pieces of the apparatus and the support
materials are being integrated to complete the detector description.

A large number of simulated events have been produced successfully using the simulation suite,
and even more production is planned. The simulation suite has thus proven its readiness and
robustness, and has been contribution to the preparation of the physics analyses, which will use real
collision data taken later this year.

References
[1] The ATLAS Collaboration, G Aad et al 2008 The ATLAS Experiment at the CERN Large
Hadron Collider JINST 3 S08003
[2] Katzy J et al. 2009 Monte Carlo Generators in Atlas software This volume
[3] Chapman J et al 2009 The ATLAS Detector Digitization Project for 2009 data taking This volume
[4] Marshall Z et al 2009 Validation and computing and performance studies for the ATLAS
simulation This volume
[5] The ATLAS Collaboration 2005 ATLAS computing : Technical Design Report
(Preprint ATLAS-TDR-017, CERN-LHCC-2005-022)
Athena Core software http://cern.ch/atlas-computing/packages/athenaCore/athenaCore.php
[6] Dell’Acqua A, Di Simone A, Gallas M V, Rimoldi A, Tsulaia V, Ueda I, Vahsen S, Marshall Z
2008 ATLAS Simulation readiness for first data at LHC J. Phys.: Conf. Ser. 119
(Preprint ATL-SOFT-CONF-2007-001)
[7] Agostinelli S et al 2003 Geant4 — a simulation toolkit Nuclear Instruments and Methods in
Physics Research A 506 250–303
Allison J et al 2006 Geant4 developments and applications IEEE Transactions on Nuclear
Science 53 270–278
[8] Boudreau J and Tsulaia V 2005 The GEOMODEL Toolkit for Detector Description Proc.
CHEP 2004 353-356, (Preprint CERN-2005-002)

Acknowledgments
The authors want to thank all ATLAS developers in the detector groups that contributed to the
successful and constant improvement of their implementation quality, and all organizations that gave
financial support for the realization of this project. We are also extremely grateful to colleagues from
the Geant4 Collaboration for helping in problem solving and for sharing their experiences with us, in
particular J. Apostolakis.