# The ATLAS Trigger Simulation with Legacy Software

**Catrin Bernius**

SLAC National Accelerator Laboratory, Menlo Park, California

E-mail: Catrin.Bernius@cern.ch

**Gorm Galster**

The Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark

E-mail: Gorm.Galster@cern.ch

**Andy Salnikov**

SLAC National Accelerator Laboratory, Menlo Park, California

E-mail: Salnikov@slac.stanford.edu

**Joerg Stelzer**

CERN, Geneva, Switzerland

E-mail: Joerg.Stelzer@cern.ch

**Werner Wiedenmann**

University of Wisconsin, Madison, Wisconsin

E-mail: Werner.Wiedenmann@cern.ch

**Abstract.** Physics analyses at the LHC require accurate simulations of the detector response and the event selection processes, generally done with the most recent software releases. The trigger response simulation is crucial for determination of overall selection efficiencies and signal sensitivities and should be done with the same software release with which data were recorded. This requires potentially running with software dating many years back, the so-called legacy software, in which algorithms and configuration may differ from their current implementation. Therefore having a strategy for running legacy software in a modern environment becomes essential when data simulated for past years start to present a sizeable fraction of the total. The requirements and possibilities for such a simulation scheme within the ATLAS software framework were examined and a proof-of-concept simulation chain has been successfully implemented. One of the greatest challenges was the choice of a data format which promises long term compatibility with old and new software releases. Over the time periods envisaged, data format incompatibilities are also likely to emerge in databases and other external support services. Software availability may become an issue, when e.g. the support for the underlying operating system might stop. The encountered problems and developed solutions will be presented, and proposals for future development will be discussed. Some ideas reach beyond the retrospective trigger simulation scheme in ATLAS as they also touch more generally aspects of data preservation.

## 1. Introduction

To analyse the data taken with the ATLAS detector [1] at the Large Hadron Collider (LHC), many analyses require a corresponding sample of simulated Monte Carlo (MC) data. This means that the production of new MC data needs to be maintained for all data taking periods. While MC data are generally produced with the newest software release, the simulated trigger response needs to be in agreement with when the data was taken by ideally using the same trigger algorithms and selections. This is necessary to reproduce selection efficiencies and trigger response as close as possible to those in the accumulated real data sample. Therefore the accurate re-simulation (e.g. because of improved detector response description, simulation of new physics processes etc.) of in particular the trigger response for data taking periods from several years ago poses a challenge. A solution to this is to use legacy software, the trigger software and conditions data that match the simulated data-taking period dating potentially many years back.

## 2. The ATLAS Monte Carlo Simulation Chain

A simplified view of the ATLAS MC simulation chain [2] is shown in Figure 1. Generated physics processes are put through a detailed simulation of the detector response (*Hits*). They are then available after the digitization processes as simulated raw data which are the input to the trigger simulation, adding the trigger response record to the event data. The MC simulation is completed by further event reconstruction. Data exchange between digitization, simulation and further reconstruction in form of *Raw Data Object (RDO)* [3], a *ROOT/POOL* [4, 5] based data format.

## 3. Considerations and Options for Simulation of the Trigger Response

There are several reasons that will require to re-simulate the trigger response, e.g. an improved description and/or understanding of the detector response, improved software for offline reconstructions in terms of algorithms and methods, increase of the MC sample size for future studies and the introduction of new event generators.

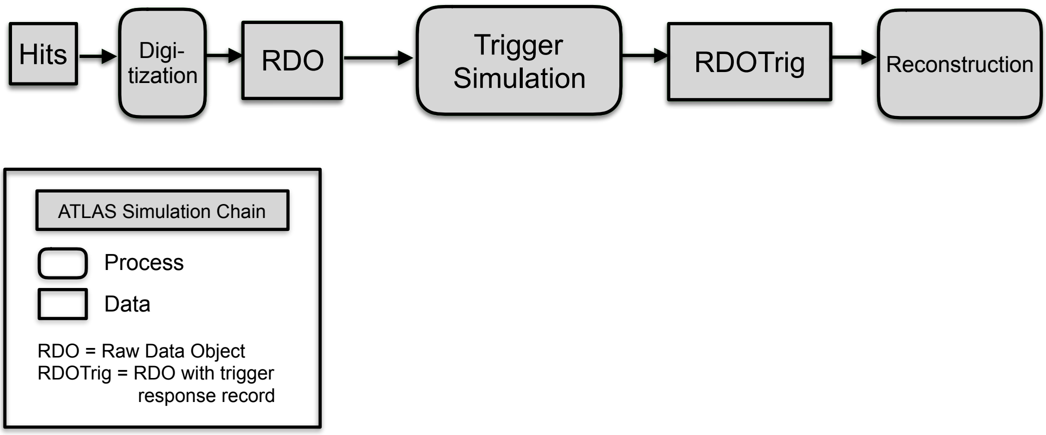

**Figure 1.** Simplified view of the ATLAS MC simulation chain. The physics processes that are generated are first put through a detailed simulation of the detector response (*Hits*), then digitized. The data is then passed on to the trigger simulation and further reconstruction. For the entire chain the same common simulation release is used. The data are exchanged in form of *Raw Data Object (RDO)*.

While detector response simulation and event reconstruction should be done with the newest software, the version used for the simulation of the trigger response needs to match that used for data taking. Mismatch between the software versions imposes a number of challenges with the most direct challenge being the data format compatibility with respect to the trigger response simulation. Therefore forward compatibility of format and content produced by new detector simulation or the possibility of the detector response conversion to an older format readable by the old trigger simulation needs to be guaranteed. Similarly backward compatibility or a conversion step need to be considered for the reconstruction that has to be able to read the old trigger response record.

Besides the data format challenges, there are the changes to hardware architectures, operating systems, core components and compiler changes which make it impossible to run the old trigger software "as is" today. The option of porting old trigger selection code to new simulation releases poses various problems as it would be necessary to keep old selection lines, algorithms, their configuration and conditions data operational along with the most recent trigger selections. This is not only problematic as it requires significant maintenance and manpower efforts but because of conflicting requirements in the trigger selections, the preservation of knowledge and the maintenance of the the infrastructure services.

The reuse of unmodified legacy trigger selection code from old releases can provide the most accurate trigger simulation of past data taking periods. The option to rerun legacy code allows for a strategy where the representative releases for a data taking period are chosen when all the required information and expertise is still available. The same selection algorithms and trigger configurations as during data taking can be applied and no or very little maintenance effort for conservation of the legacy trigger selection lines is required. Due to the advantages of this option, it has been further investigated and a simulation chain has been developed.

## 4. Trigger Simulation with Legacy Trigger Selection Software

To realise running the trigger legacy trigger selection code, there are various considerations that have to be addressed.

To be able to run with legacy releases, the long term conservation of trigger software releases, configuration data and conditions data becomes a necessity. In ATLAS, the software distribution which also includes all required external software components, the compilers with their run time environments and software configuration and development tools is done on the *cvmfs* file system [6].

Furthermore the ATLAS simulation chain requires a split into sub-steps which can use different software release versions. In the present simulation chain data are exchanged between the simulation modules as RDO data containing MC truth information and meta data with data processing parameters using a single software release. This guarantees the data compatibility between all steps, which is not given when running the trigger simulation with a different (legacy) software. In this case the data compatibility becomes an issue with respect to forward compatibility (output data from newer detector simulation need to be readable with older trigger releases) and backward compatibility (output data from old trigger release needs to be readable with newer reconstruction release).

Data which are directly read out from the detector hardware are available as *byte stream (BS)* data. This format is based on containers of 32 bit integers with a simple payload structure and described by the ATLAS raw data format [7]. It is tightly coupled to the detector readout hardware with only very few changes over time. Since it is a requirement that all ATLAS software releases can read data in BS format from all data taking periods, backward compatibility is guaranteed. Due to the very simple structure of the data format providing forward compatibility is much easier. The scripts to convert RDO data to BS data are already available in simulation releases and some ATLAS detectors already provide forward compatibility. The BS data format is therefore well suited to reach compatibility between different software release versions. However it does not provide structures to hold MC truth information or simulation meta data like the RDO data. The trigger simulation step only needs the detector raw data as input without any additional MC information and provides as output the trigger decision record in BS format. For the subsequent event reconstruction step the output can be merged with the other event reconstruction input data.

The additionally introduced sub-steps of RDO to byte stream data conversion before the trigger simulation and data merging after the trigger simulation which integrate the trigger simulation with a legacy release in the simulation chain are shown in Figure 2. In the data merging step the trigger decision record is converted from BS format to RDO format and added to the simulated data.

## 5. Use of Virtualisation - An Outlook

In the medium term the use of older releases on new operating systems can be achieved with a compatibility layer. However in the long term, using legacy code "as is" will be impossible on modern computing platforms. New computing hardware technologies, operating system changes and updates to the compilers and core run time libraries require changes to the legacy trigger releases. Virtualization enables the abstraction of the hardware and software run time layer from the underlying computing platform but comes at the expense of computational and resource overhead. Furthermore, external infrastructure services, e.g. for data input and output will most likely undergo important changes on a time scale of 10 years, requiring adaptations for the legacy trigger release. Patch releases to the legacy code should collect all these changes and allow for the integration with the environment external to the virtual machine. Encapsulation of the legacy trigger selection code in a virtual machine image extends the time span for of using legacy trigger selection code.

## 6. Conclusion

The challenges of simulating the trigger response precisely in MC simulation campaigns for data-taking periods dating back many years have been discussed and a strategy how to overcome

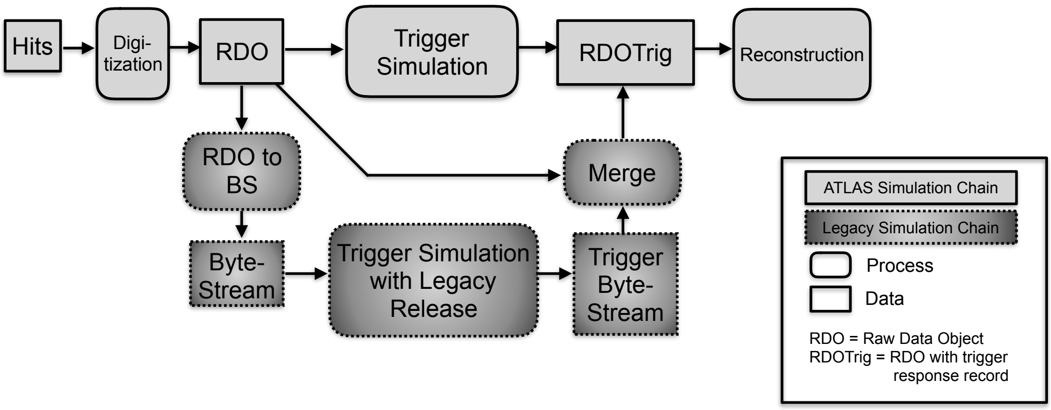

**Figure 2.** Simplified view of the ATLAS MC simulation and legacy simulation chain using a different, legacy software release for the trigger simulation. For the legacy simulation chain the data are converted from RDO to byte stream format and passed through the trigger simulation using a legacy software release. The resulting byte stream with trigger response record is then merged with information from the initial RDO and converted back to RDO format.

these difficulties has been laid out. While detector simulation and reconstruction use the newest available software releases, the simulation of the trigger response is done with older software which was used for this particular data-taking period that is simulated. Issues with data format compatibility between the different simulation steps have been studied and a modified simulation chain has been identified and implemented. Further work is still needed to completely integrate this modified simulation chain in the production workflow for large scale simulation campaigns in ATLAS.

## References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008).

[2] ATLAS Collaboration, Eur. Phys. J. C **70**, 823 (2010).

[3] P. van Gemmeren, D. Malon, for the ATLAS Collaboration, *Supporting High-Performance I/O at the Petascale: The Event Data Store for ATLAS at the LHC*, Proccedings of IEEE Cluster 2010, Grece, September 2010.

[4] *ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization*, Computer Physics Communications, Anniversary Issue, Volume 180, Issue 12, Dec. 2009, P. 2499-2512.

[5] *POOL - Persistency Framework*, [`http://pool.cern.ch`].

[6] *CernVM Software Appliance*, [`http://cernvm.cern.ch/portal`].

[7] C.P.Bee, D.Francis, L.Mapelli, R.McLaren, G.Mornacchi, J.Petersen, F.J.Wickens, ATLAS-DAQ-98-129 (2004), [`https://cds.cern.ch/record/683741`].