Improving Generators' Interface to Support LHEF V3 Format

Fernando Cornet-Gomez
Universidad de Granada, Spain
DESY Summer Student
Supervisor: Ewelina M. Lobodzinska

September 11, 2014

Abstract

The aim of the project was to modify the Les Houches Event File Interface in order to make it capable of reading the newest version of LHEF format. A partially modified version of the interface was used as an starting point. From there, some features where implemented until the software was fully functional.

Universidad de Granada

INTRODUCTION

This report is giving a dedicated account of my project in the ATLAS (**A** **T**oroidal **L**HC **A**paratu**S**) group[1] at DESY[2]. ATLAS is one of the LHC[3] (**L**arge **H**adron **C**ollider)experiments. It has a rich physics programme, including precise measurements of Standard Model processes as electroweak or top physics. As well, ATLAS research beyond the Standard Model, looking for supersymmetry (SUSY) and other new theories. The 4th of July, 2012, the discovery of a Higgs boson was announced by ATLAS[4] and CMS[5][6] collaborations. The ATLAS collaboration is formed by over 3000 people from 38 countries and 177 universities 1000 students.

The main purpose of the project was to implement new features for what is called Les Houches Event File[7] (LHEF hereafter) interface. This interface is part of Athena[8], an ATLAS control framework for particle physics Monte Carlo simulation.

In particle physics Monte Carlo simulation we have four basic steps:

- Event Generation (Hard Process)
- Parton Showers
- Hadronization
- Underlying Event

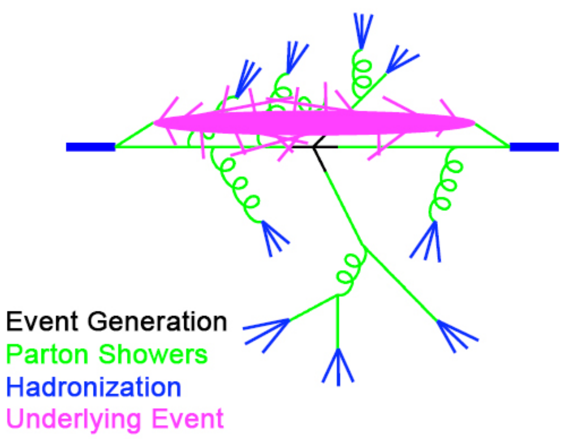


Figure 1: [9] MC simulation steps

These steps can be better understood in Figure 1, where an event draw and the four phases are shown.

In the following sections this framework is going to be explained as well as some formats (HepMC and LHEF) that would be necessary to know for a full understanding of the project. After that, the modified interface will be presented.

THEORY

Athena

As it is mentioned before, Athena is a control framework that is used to run the particle physics Monte Carlo simulation in ATLAS. The whole process can be done using this framework. Athena has six basic steps:

- Initialization: The settings and initial values can be introduced to the framework via the JobOptions. This JobOptions are written in Python.
- Call the generator via the interface and give it the settings.
- Read the event data generated.
- Keep the read data into Storegate. It is writen in HepMC format what will be exlained in the next section.
- Aply some helper classes, for example, a filter .
- Permanent store in a .root file.

The Athena workflow diagram is shown at Figure 2. Depending on the generator, the interface can proceed the steps 2 and 3 in different ways. The interface packages inherit from the common base class, GenModule wich provides the basic functions. However, they have to been modified for each generator: initialize() for the generator configuration, execute() for generation the event, translate it to HepMC format

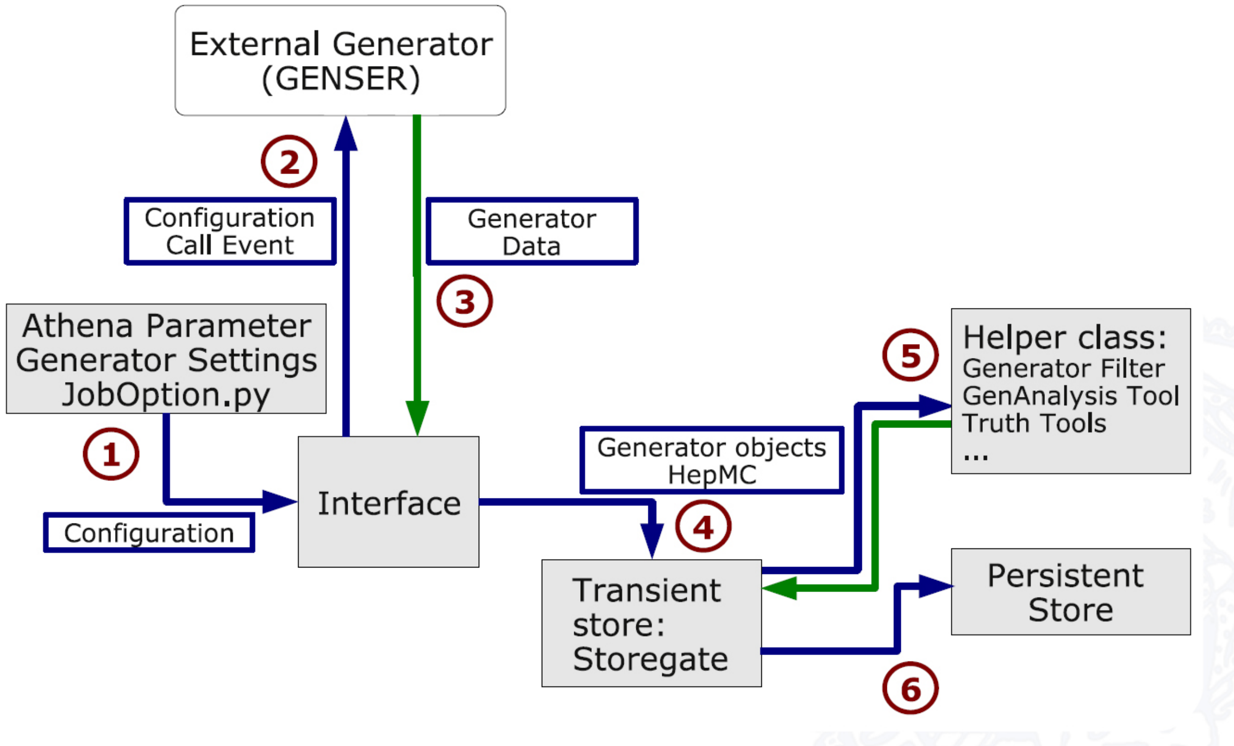


Figure 2: Athena workflow

and store it, and finalize() which call destructors.

As have been mentioned before, for the Monte Carlo simulation an event generator is needed. The generators that Athena uses can be grouped in three categories. The most well-known full generators as Pythia[10] or Herwig[11] (newest version Pythia 8 and Herwig++). With these generators it is possible to run the whole simulation, including parton shower and hadronization.

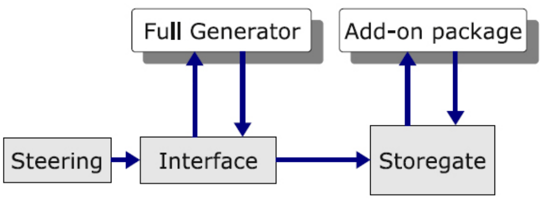


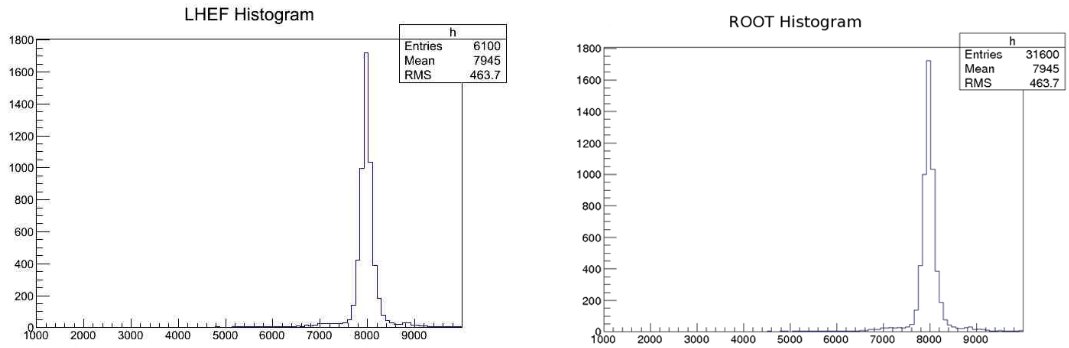
Figure 3: Modified workflow

The second group is formed by the Specific purpose add-on packages to generators like Tauola and Photos. These generators work at Storegate level what means that the workflow change a little bit. As is shown in the Figure 3, these generators read the data in HepMC format from the Storegate, modify the events and keep it back to the Storegate in HepMC format.

The third and last group is formed by the parton level generators. Most of the generators belong to this third group as POWHEG, MC@NLO, MadGraph, etc. The components of this group only simulate the first step, that is to say, the event generation. As a result of running these king of generators we get the Matrix Elements. This file is written in Les Houches Event File format. LHEF format is the international agreement about the way of write the matrix element. Then, a full generator is needed to proceed with the simulation, parton shower and hadronization. To use this kind of generators the LHEF interface that we have been working on it's needed.

Formats

During the report two formats has been mentioned. In the following lines those formats will



(a) LHEF (b) ROOT

Figure 4: Weights histograms

been briefly explained. The HepMC (High energy physics Monte Carlo) is an object oriented event record written in C++. The developing of HepMC is not related with an specific experiment or generator. The main purpose of the format is to make easy the way of storing the data as well as helping in the data generation.

This was the chosen option when the way of keeping the data was discussed. Another solution was proposed. The idea was to associate a label (string) to each event. This idea was discarded because it is much less efficient than the HepMC format. In addition to create a common format was supported by the creators of the POWHEG generator. More information about how actually works this format can be found in the user manual or the HepMC web page[12].

On the other hand, LHEF format have appeared a couple of times. Les Houches Event File receive his name from the French village Les Houches, located in the French Alps. There took place some meetings where this format was created. The main purpose of LHEF is to have a common way of writing the matrix elements. That make the work much easier when reading the data. In the matrix elements we have a lot of information about each event as the number and kind of particles, the mother, the momentum, energy and mass. As well as PDF and cross section information. At the moment there are three versions of the LHEF format. The version two is very similar two the fist one. In the newest version, the third one, the main change is the addition of the possibility to include multiple weights in one event block. These weights can appear in two different ways, this first one and also the more efficient, is displayed all of them in one line. The second possibility is to have one line per weight.

RESULTS & DISCUSSION

The project consists in modify the LHEF interface (LHEF_ i) which at the beginning only was able to read the first version of LHEF. After the month and a half of working in ATLAS group in DESY, the interface is fully functional and capable of reading the LHEF V3. This interface is written in FORTRAN.

Although the format is well defined and clear explained in the manuals, there are at least two different interpretations of it. We have been working with tow different event generators as POWHEG and MC@NLO. The output of these two generators if different in some aspects. At the moment, we are capable of reading the data generated by both of them. The interface gives prioriy to the newest version, and if it find more than one format in the file it will read the one that have been read first. The interface will jump over the rest of the possible formats.

In Figure 4 we can see the histogram of the weights from the matrix elements fig. 4(a) and after going trough the interface and the generator (Pythia or Herwig) fig. 4(b). The two histogram should be, and actually are, equal. If they were not, it would mean that the interface is not just reading and passing the data to the generator but also would be changing it.

As you can see the number of entries is not equal. That is because the way of filling the histograms. From the matrix element only the weights are read and plotted in the histogram. However, after run the generator, Pythia or Herwig keep the weights in a leave (root file) together with the PDF ID, the renormalization factor, factorization factor and other few parameters. So the histogram is filled with this leave and then, only the interval where the weights are, is plotted. If we subtract one from the other, we get, as is shown in Figure 5, a empty histogram.

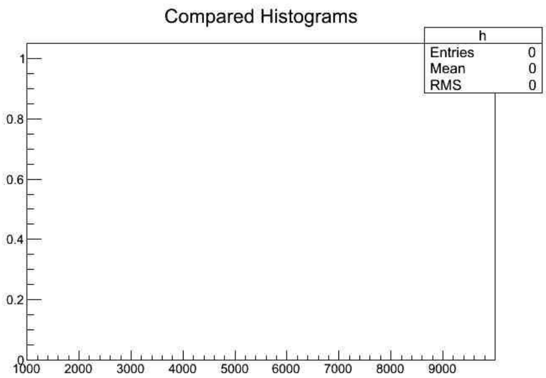


Figure 5: 4(a) minus 4(b)

Acknowledgements

I want to thank in first place, my supervisor, Ewelina Maria Lobodzinska, for helping me and answer all my questions. The DESY ATLAS group for making me feel like home. I also want to thank DESY and all the organizers, for making possible this summer student program and provide financial support. And finally, the whole Departamento de Física Teórica y del Cosmos from Universidad de Granada, in particular Francisco del Aguila y José Santiado for writing for me the recommendation letters for the program application.

References

[1] ATLAS Collaboration, *http://atlas.ch/*

[2] ATLAS Collaboration, *http://desy.de/*

[3] LHC, *http://home.web.cern.ch/topics/large-hadron-collider*

[4] The ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, Phys.Lett. B716 (2012) 1-29*

[5] CMS, *http://cms.web.cern.ch/*

[6] The CMS Collaboration, *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC,Phys. Lett. B 716 (2012) 30*

[7] Les Houches Event File, *arXiv:hep-ph/0609017*

[8] ATLAS Monte Carlo team, *Monte Carlo generators in ATLAS software*

[9] B. Webber, *Monte Carlo Methods in Particle Physics*, University of Cambridge, IMPRS, Munich 19-23 November 2007.

[10] Pythia 8, *http://home.thep.lu.se/ torbjorn/Pythia.html*

[11] Herwig++, *https://herwig.hepforge.org/*

[12] HepMC 2 User Manual, *https://sft.its.cern.ch/jira/browse/HEPMC*