PROCEEDINGS OF SCIENCE

A new design for simulation and reconstruction software for the JEM-EUSO mission

Thomas C. Paul[*]
Department of Physics and Astronomy, Lehman College, City University of New York, 250 Bedford Park Boulevard West, Bronx NY 10468 USA
E-mail: thomas.paul@lehman.cuny.edu

for the JEM-EUSO Collaboration

The Extreme Universe Space Observatory (JEM-EUSO) is designed to detect the highest energy particles in the Universe by observing the fluorescence and (reflected) Cherenkov light produced when these ultrahigh energy cosmic rays (UHECR) traverse the Earth's atmosphere. Unlike existing cosmic ray observatories, JEM-EUSO will view the atmosphere from above, as the instrument will be attached to the International Space Station. Here we describe a new general-purpose software framework to facilitate detailed simulation and reconstruction of UHECR events observed by JEM-EUSO. This new software is based on open-source codes developed over roughly a decade by a collaborative effort of several particle astrophysics and high energy physics experiments, and is particularly suited to the needs of large collaborations. We explain the machinery used to manage user contributions, organize an abundance of configuration files, facilitate multi-format file handling, and provide access to time-dependent information on detector and atmospheric properties. We also discuss the strategies employed to ensure stability and maintainability in the face of a large number of user contributions. Finally, we show example applications using this new software package.

The 34th International Cosmic Ray Conference,
30 July- 6 August, 2015
The Hague, The Netherlands

[*] Speaker.

1. Introduction

One of the endemic challenges faced by large, geographically dispersed scientific collaborations is how best to share and organize software tools for data analysis. Data analysis may include rather detailed problems like simulation of complex instruments, involved algorithms required to convert raw data into reconstructed events, as well as tools for handling large data samples. The JEM-EUSO Collaboration comprises over 300 scientists from 15 nations, and faces the same software challenges as other ambitious “big science” programs.

The JEM-EUSO mission is designed to observe ultrahigh energy cosmic rays from space via measurement of the fluorescence light produced when cosmic ray induced air showers excite atmospheric nitrogen and Cherenkov light reflected from the Earth’s surface or from clouds. An overview of the project and the scientific objectives are described elsewhere in these proceedings [1, 2].

In the early design phases of JEM-EUSO, an analysis framework known as the EUSO Simulation and Analysis Framework (ESAF) [3] was developed to address the complexities of large-scale collaborative work on data simulation, reconstruction and analysis for the JEM-EUSO mission. At around the same time, the Pierre Auger Collaboration [4] was addressing the same sorts of challenges. Like the JEM-EUSO Collaboration, the Auger Collaboration is large and geographically dispersed, with over 500 researchers from 18 countries. The Auger analysis framework [5], dubbed simply Offline, has been used for simulation and reconstruction for the Auger Observatory since the first physics results were published in 2004. At the time of writing the software comprises some 340 000 lines of code and 35 000 lines of configuration information, representing a roughly 100 person-year investment according to the Constructive Cost Model [6].

The Offline software includes the latest fluorescence and Cherenkov light-yield models [7], atmospheric models and interfaces to many air shower simulation packages [8]. As Auger analyses using fluorescence measurements are quite mature at this point, the simulation algorithms in Offline have been well vetted with real data. The framework provides many utilities and conveniences, to be discussed later, which have been exercised for over a decade by a large collaboration conducting data analysis. Furthermore, parts of the Offline framework have been adopted by other collaborations, including CODALEMA [10], TUNKA [11], HAWC [12], LOFAR [13] and NA61/SHINE [14], allowing for mutually beneficial collaboration among scientists working on different experiments. The Offline framework is freely available upon request and is distributed under an open source BSD license [15].

It is natural, then, to consider whether there are potential advantages to adopting portions of the Offline software for use by JEM-EUSO, particularly the overarching framework, the utilities and the algorithms related to fluorescence light simulation. An investigation into this possibility commenced in 2013. The use of Offline for analysis of data from the recent EUSO-Balloon pathfinder mission is described elsewhere in these proceedings [16], as is the use of Offline for simulation of the LIDAR and infrared camera [17] which together compose the JEM-EUSO atmospheric monitoring system. Here we describe the overall Offline design and present a few examples.

2. Framework

The $\overline{\text{Off}}\underline{\text{line}}$ framework comprises a collection of physics algorithms contained in *modules*; a *RunController* which commands the modules to execute in a particular sequence; a read/write *Event Data Model* from which modules read information and to which they write their results; a *Detector Description* which provides an interface to conditions data, such as detector calibration, performance and atmospheric conditions; and a *CentralConfig* which directs the modules and framework components to their configuration data and which tracks provenance. These are discussed in more detail below.

Simulation and reconstruction tasks are factorized into sequences of processing steps which can be simply pipelined. Physicists prepare processing algorithms in *modules*, which they register with the $\overline{\text{Off}}\underline{\text{line}}$ framework via a one line macro. This modular design facilitates comparison of algorithms and building a variety of applications by combining modules in various sequences. Control of module sequences is implemented with a *Run Controller*, which directs module execution according to a set of user provided instructions. We devised an XML-based language as one option for specifying sequencing instructions; this approach has proved sufficiently flexible for the majority of our applications, and it is simple to use.

The $\overline{\text{Off}}\underline{\text{line}}$ framework includes parallel hierarchies for accessing data: the *detector description* for retrieving conditions data, including detector geometry, calibration constants and atmospheric conditions; and an *event data model* for reading and writing information that changes for each event.

The detector description provides an interface from which module authors can retrieve the conditions data. Data requests are relayed to a back-end comprising a registry of so-called *managers*, each of which is capable of extracting a particular sort of information from various data sources. The manager mechanism is configurable and relieves authors of the physics code from having to deal with the details of selecting and decoding the correct data source. The general structure of the detector description machinery is illustrated in figure 1.

The event data model contains raw, calibrated, reconstructed and Monte Carlo information, and serves as the backbone for communication between modules. The event is instrumented with a protocol allowing modules to discover its constituents at any point in processing, and thereby determine whether the input data required to carry out the desired processing are available. $\overline{\text{Off}}\underline{\text{line}}$ is also equipped to read formats employed by the most popular air shower simulation packages [8].

The $\overline{\text{Off}}\underline{\text{line}}$ framework includes a system to organize and track data used to configure the software for different applications as well as parameters used in the physics modules. The *Central Config* configuration tool points modules and framework components to the location of their configuration data, and connects to Xerces-based [9] XML parsers to assist in reading information from these locations. We have wrapped Xerces with our own interface which provides ease of use at the cost of somewhat reduced flexibility, and which also adds functionality such as automatic units conversion and casting to various types, including commonly used container types.

The *Central Config* keeps track of all configuration data accessed during a run and stores them in an XML log file, which can be subsequently used to reproduce a run with an identical configuration. This allows collaborators to exchange configuration data for comparing results. The

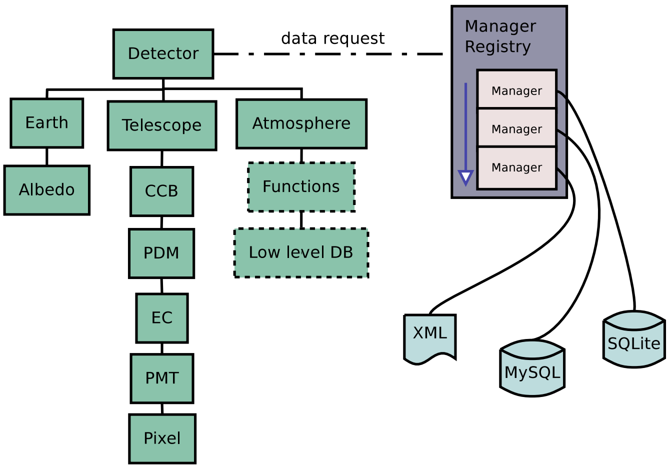

Figure 1: Simplified depiction of the machinery of the detector description. The user interface (left) comprises a hierarchy of objects describing the various components of the detector, where the Atmosphere and the Earth surface or intervening clouds (which reflect Cherenkov light into the telescope) are considered to be part of the detector. The objects shown under the Telescope are arranged with the hierarchy characteristic of the focal surface: the focal surface comprises 18 Cluster Control Boards (CCBs), each of which gathers signals from 8 Photo Detector Modules (PDMs). Each PDM is subdivided into 9 Elementary Cells (ECs), each of which contains 4 multi-anode Photomultiplier Tubes (PMTs), each of which in turn contains 64 pixels. At each level in the hierarchy, detector characteristics can be requested. Data requests are relayed to back-end managers which can fetch data from different sources. Different data formats (eg. flat files, databases, or ROOT [18] files) can be accessed from the back-end, allowing developers to choose the best technology to store different sorts of non-event data, and to provide that information in a unified way so that algorithm developers need not deal with the minutiae of reading and decoding different data sources. Managers are organized in an "chain-of-responsibility," so if an upstream manager cannot answer a request, it is passed to a downstream manager for another try.

logging mechanism is also used to record the versions of modules and external libraries which are used for each run.

Syntax and content checking of the configuration files is implemented using W3C XML Schema validation [19]. Schema validation is used not only for internal configuration prepared by framework developers, but also to check the contents of physics module configuration files. This approach reduces the amount of code users and developers must prepare, and supports very robust checking.

3. Quality control, build system and dependencies

As in many large software projects, each low level component of the framework is verified with a small test program, known as a *unit test*. We have adopted the CppUnit testing framework [20] to help with implementing these tests. In addition to unit tests, a set of higher level acceptance tests has been developed which is used to verify that complete physics applications continue to function as expected, within some tolerance, during ongoing development. We employ a BuildBot system [21] to automatically compile the $\overline{\text{Off}}\underline{\text{line}}$ software, run the unit and acceptance tests, and

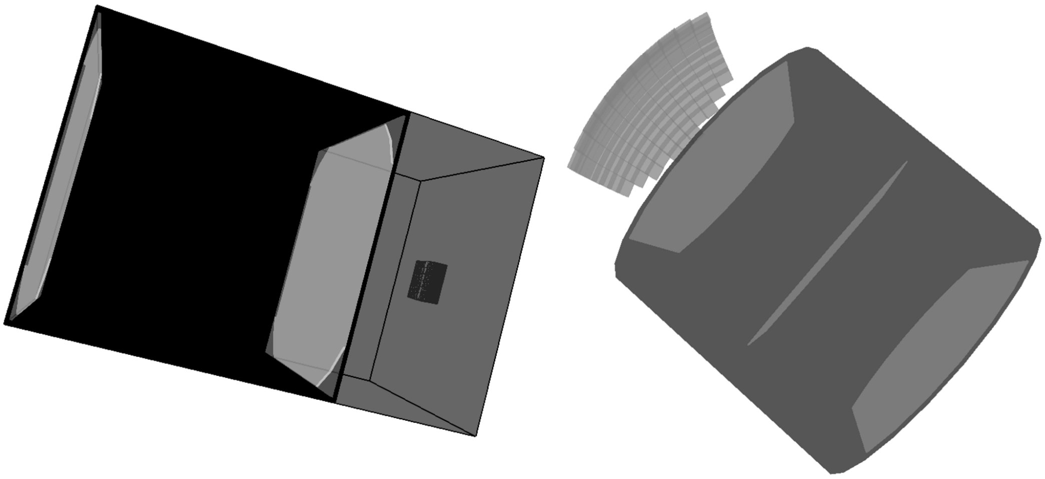

Figure 2: Configurations EUSO-TA (left) and JEM-EUSO (right), as generated in Geant4. EUSO-TA comprises 2 Fresnel lenses (the lightly shaded regions at the ends of the box) and 1 PDM (the small sqare to the right). In the example on the left, a laser shot from the TA Central Laser Facility [30] has been simulated, and one can see the resulting vertical track on the PDM. JEM-EUSO comprises 3 lenses and 137 PDMs.

email developers in case of problems. The BuildBot runs each time the software repository is modified.

The Offline build system is based on the CMake cross-platform build tool [22], which has proven adequate to manage this project. In order to ease installation of Offline and its external dependencies, we have adopted a tool known as APE (Auger Package and Environment) [23]. APE is a dependency resolution engine co-developed by the Auger and HAWC Collaborations. It automatically downloads a vetted combination of external packages required by Offline, builds them in whatever native build system applies for each package, and sets up the user's environment accordingly. APE is freely available, and used by the NA61/SHINE Collaboration, as well as Auger and HAWC.

4. Examples

As mentioned previously, these proceedings contain descriptions of some applications using the Offline framework [16]. We briefly discuss a few additional examples here.

A primary goal Offline was designed to address is the requirement for the degree of flexibility required to support a large variety of applications. As an example, we consider the simulation of several EUSO-like instruments. Aside from the simulating the full baseline JEM-EUSO design, it is of course necessary to simulate the pathfinder instruments which will precede it, including the successful first EUSO-Balloon mission [24], the ongoing EUSO-TA project [25] and the mini-EUSO and long duration balloon flight currently in the planning stages [26, 27].

As mentioned above, the configuration machinery employed by Offline is based on XML and XML Schema. This standard provides a robust mechanism to include markup from external sources

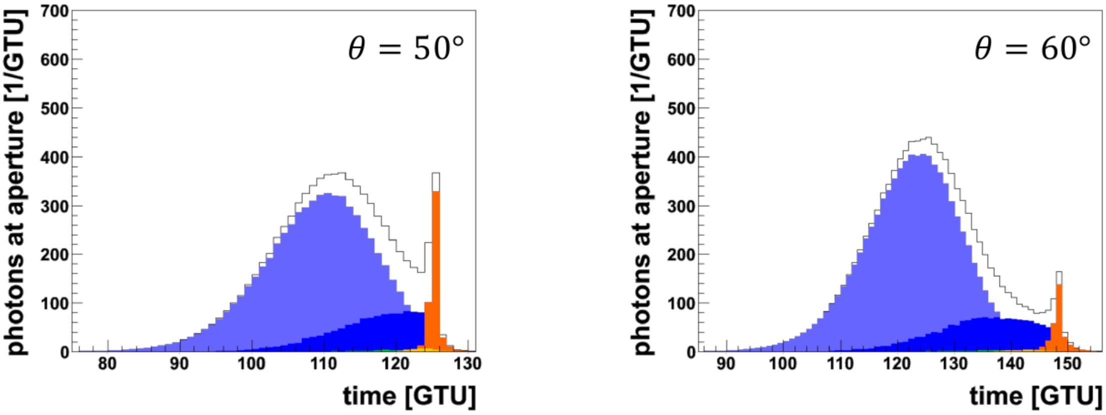

Figure 3: Examples of simulations of a 10^{20} eV air shower at two zenith angles ($\theta = 50°$ and $60°$). The total number of photons arriving at the JEM-EUSO aperture as a function of time is indicated by the solid line. The light shaded (blue) region shows the contribution from fluorescence light, while the dark shaded region shows the contribution from scattered Cherenkov light. The bump at the right end of the time trace arises from Cherenkov light reflecting off the Earth surface. A Gate Time Unit (GTU) = 2.5 μs.

in another file, such that a single XML configuration file can be constructed from a collection of other files. This suits our needs, as each of the EUSO-like projects mentioned above comprises different configurations of the same (or similar) components. For instance, both EUSO-TA and EUSO-Balloon contain 2 Fresnel lenses and a single PDM. The long duration balloon flight will likely be configured differently, with more than 1 PDM. The baseline JEM-EUSO design calls for 3 lenses and 137 PDMs. Thus a straightforward mix-and-match scheme can be used to assemble the appropriate XML description for each project. As an example, we show in figure 2 the Geant4 [28] detector geometries constructed for EUSO-TA and JEM-EUSO. The underlying C++ code for the two cases is the same; only the single configuration file which imports the appropriate lens and focal surface descriptions needs to be altered.

In figure 3 we illustrate an $\overline{\text{Off}}\underline{\text{line}}$ simulation of two air showers as seen at the JEM-EUSO aperture [29]. As mentioned previously, the modules for computing light generation and propagation developed by the Auger Collaboration have been well vetted with data; this work was co-opted for the simulations shown here.

5. Conclusions

The challenges of developing large-scale collaborative analysis software have become common with the onset of complex experiments operated by large, geographically dispersed collaborations. Here we have discussed adaptation of the $\overline{\text{Off}}\underline{\text{line}}$ software package originally developed for the Auger collaboration to the needs of the JEM-EUSO collaboration. This work extends an already fruitful cross-pollination among a number of different experiments which have adopted this software. At the time of writing, the $\overline{\text{Off}}\underline{\text{line}}$ package has been used by the JEM-EUSO collaboration for simulation and reconstruction the EUSO-Balloon and EUSO-TA pathfinders as well as simulation of the baseline JEM-EUSO telescope, with preliminary results reported elsewhere in these proceedings.

Acknowledgment: This work was partially supported by US NSF grant PHY-1205854, NASA grant NNX13AH52G, by the Italian Ministry of Foreign Affairs, General Direction for the Cultural Promotion and Cooperation, by the 'Helmholtz Alliance for Astroparticle Physics HAP' funded by the Initiative and Networking Fund of the Helmholtz Association, Germany. The Spanish Consortium involved in the JEM-EUSO Space Mission is funded by MICINN & MINECO under the Space Program projects: AA2009-06037-E/AYA, AYA-ESP2010-19082, AYA-ESP2011-29489-C03, AYA-ESP2012-39115-C03, AYA-ESP2013-47816-C4, MINECO/FEDER-UNAH13-4E-2741, CSD2009-00064 (Consolider MULTIDARK) and by Comunidad de Madrid (CAM) under projects S2009/ESP-1496 & S2013/ICE-2822.

References

[1] A. Santangelo *et al.*, "The JEM-EUSO Program," these proceedings (ICRC2015 ID:0694).

[2] A. Olinto *et al.*, "The JEM-EUSO Science," these proceedings (ICRC2015 ID:0735).

[3] C. Berat *et al.*, "ESAF: Full Simulation of Space-Based Extensive Air Shower Detectors," *Astropart.Phys.* **33** (2010) 221.

[4] A. Aab *et al.* [Pierre Auger Collaboration], "The Pierre Auger Cosmic Ray Observatory," [arXiv:1502.01323].

[5] S. Argiro, S. L. C. Barroso, J. Gonzalez, L. Nellen, T. C. Paul, T. A. Porter, L. Prado, Jr. and M. Roth *et al.*, 'The Offline Software Framework of the Pierre Auger Observatory," Nucl. Instrum. Meth. A **580**, 1485 (2007) [arXiv:0707.1652 [astro-ph]].
P. Abreu *et al.* [AUGER Collaboration], "Advanced functionality for radio analysis in the Offline software framework of the Pierre Auger Observatory," Nucl. Instrum. Meth. A **635**, 92 (2011) [arXiv:1101.4473 [astro-ph.IM]].

[6] B. Boehm, **Software engineering economics**, Englewood Cliffs, NJ:Prentice-Hall, 1981. ISBN 0-13-822122-7.

[7] M. Bohacova *et al.* [AIRFLY Collaboration], "A novel method for the absolute fluorescence yield measurement by AIRFLY," Nucl. Instrum. Meth. A **597**, 55 (2008) [arXiv:0812.3649 [astro-ph]].

[8] R. Engel, D. Heck and T. Pierog, "Extensive air showers and hadronic interactions at high energy," Ann. Rev. Nucl. Part. Sci. **61**, 467 (2011).
S. J. Sciutto, "AIRES: A System for air shower simulations. User's guide and reference manual. Version 2.2.0," astro-ph/9911331.
T. Bergmann, R. Engel, D. Heck, N. N. Kalmykov, S. Ostapchenko, T. Pierog, T. Thouw and K. Werner, "One-dimensional Hybrid Approach to Extensive Air Shower Simulation," Astropart. Phys. **26**, 420 (2007) [astro-ph/0606564].
H. J. Drescher and G. R. Farrar, "Air shower simulations in a hybrid approach using cascade equations," Phys. Rev. D **67**, 116001 (2003) [astro-ph/0212018].

[9] `http://xerces.apache.org/`

[10] D. Ardouin, A. Belletoile, D. Charrier, R. Dallier, L. Denis, P. Eschstruth, T. Gousset, F. Haddad *et al.*, "CODALEMA: A cosmic ray air shower radio detection experiment," Int. J. Mod. Phys. **A21S1**, 192-196 (2006).

[11] B. A. Antokhonov, D. Besson, S. F. Beregnev, N. M. Budnev, O. B. Chvalaev, A. Chiavassa, O. A. Gress, N. N. Kalmykov *et al.*, "A new 1-km**2 EAS Cherenkov array in the Tunka valley," Nucl. Instrum. Meth. **A639**, 42-45 (2011);
`http://heplx3.phsx.ku.edu/ riceuser/ku-tunka/`

[12] A. U. Abeysekara *et al.* [HAWC Collaboration], "The HAWC Gamma-Ray Observatory: Design, Calibration, and Operation," arXiv:1310.0074 [astro-ph.IM].

[13] http://www.lofar.org/

[14] N. Angrily *et al.* [NA61 Collaboration], "NA61/SHINE facility at the CERN SPS: beams and detector system," JINST **9**, P06005 (2014) [arXiv:1401.4699 [physics.ins-det]].
O. Wyszynski, A. Laszlo, A. Marcinek, T. Paul, R. Sipos, M. Szuba, M. Unger and D. Veberic, "Legacy code: Lessons from NA61/SHINE offline software upgrade adventure," J. Phys. Conf. Ser. **396**, 052076 (2012).
R. Sipos, A. Laszlo, A. Marcinek, T. Paul, M. Szuba, M. Unger, D. Veberic and O. Wyszynski, "The offline software framework of the NA61/SHINE experiment," J. Phys. Conf. Ser. **396**, 022045 (2012).

[15] http://www.opensource.org/

[16] J. Eser *et al.*, "Laser direction reconstruction in the EUSO-Balloon experiment," these proceedings (ICRC2015 ID:0860).
B. Panico *et al.*, "Analysis of EUSO-Balloon data with Offline," these proceedings (ICRC2015 ID:1155).

[17] L. del Peral *et al.*, "JEM-EUSO space mission infrared camera simulation nad reconstruction implementation in Offline," these proceedings (ICRC2015 ID:1292).

[18] https://root.cern.ch/drupal/

[19] http://www.w3.org/standards/xml/schema

[20] http://sourceforge.net/projects/cppunit/

[21] http://buildbot.net/

[22] http://www.cmake.org

[23] https://svn.auger.unam.mx/trac/projects/ape/

[24] P. von Ballmoos *et al.*, "General overview of EUSO-Balloon mission," these proceedings (ICRC2015 ID:0725).

[25] M. Casolino *et al.*, "EUSO-TA. A ground precursor to JEM-EUSO telescope on the Telescope Array site: status and perspectives," these proceedings (ICRC2015 ID:1309).

[26] M. Ricci *et al.*, "Mini-EUSO: A pathfinder for JEM-EUSO to measure the UV Earth background from ISS," these proceedings (ICRC2015 ID:0560).

[27] L. Wiencke *et al.* "EUSO-Balloon mission to record extensive air showers from near space," these proceedings (ICRC2015 ID:0816).

[28] https://geant4.cern.ch

[29] S. Falk *et al.*, "Atmospheric influence on space-based observation of high-energy cosmic rays," proceedings of the 24'th European Cosmic Ray Symposium (in preparation).

[30] T. Abu-Zayyad *et al.* [Telescope Array Collaboration], "The surface detector array of the Telescope Array experiment," Nucl. Instrum. Meth. A **689**, 87 (2012) [arXiv:1201.4964 [astro-ph.IM]].
H. Tokuno, Y. Tameda, M. Takeda, K. Kadota, D. Ikeda, M. Chikawa, T. Fujii and M. Fukushima *et al.*, "New air fluorescence detectors employed in the Telescope Array experiment," Nucl. Instrum. Meth. A **676**, 54 (2012) [arXiv:1201.0002 [astro-ph.IM]].