PROCEEDINGS OF SCIENCE

LCG MCDB and HepML, next step to unified interfaces of Monte-Carlo simulation

Sergey Belov*
Joint Institute for Nuclear Research
Dubna, Moscow region, Russia, 141980
E-mail: belov@jinr.ru

Lev Dudko
Scobeltsyn Institute of Nuclear Physics of Lomonosov Moscow State University
Moscow, Russia, 119992

Alexander Sherstnev
R. Peierls Centre for Theoretical Physics, University of Oxford
Oxford, OX1 3PU, UK and
SINP, Moscow State University, Moscow, Russia (on leave)

In this article we present a way of building fully automated Monte-Carlo simulation chain. In recent years there was a need for common place to store sophisticated MC event samples prepared by experienced theorists. Also such samples should be accessible in some standard manner to be easily imported and used in experiments' software.
The main motivation behind the LCG MCDB project is to make sophisticated MC event samples and their structured descriptions available for various groups of physicists working on LHC. All the data from MCDB is accessible for end-users in several convenient ways from Grid, on the Web and via application program interface.
HepML (High Energy Physics Markup Language), developed in collaboration of LCG MCDB and CEDAR teams and several MC generator authors, should give a unified XML description of event samples simulated by Matrix Element (ME) generators. By means of HepML, MC generators can automatically create self documented event samples. The other main purpose of HepML is to keep MC generation parameters for further MC generators tuning. It is possible to extend HepML as an XML standard to keep necessary information for the different levels of simulation in HEP, from theoretical model to a simulation of detector responds. HepML provides the possibility to use and develop many standard tools for the comparison, validation, graphical representation of the results and create transparent unified interfaces for the different software in HEP on the modern level of Computer science.
Using MCDB and HepML together gives a possibility of automation of such significant part of MC simulation chain as correct transfer of physics events from Matrix Element generators to Shower generators and then to detector simulation. Such machine-driven manner allows to avoid errors coming from human factor, saving a lot of time and efforts for end users of trusted and verified shared MC samples.

XII Advanced Computing and Analysis Techniques in Physics Research
November 3-7 2008
Erice, Italy

*Speaker.

PoS(ACAT08)115

1. Introduction

Complexity of current experiments in high energy physics requires calculation of scattering processes with many particles in the final state. Monte-Carlo (hereafter, MC) simulation becomes technically more complicated and is exposed to mistakes due to the human factor. Thereby making Monte-Carlo simulation more automated can be considered as an important task.

The LCG MCDB project has been created to facilitate communication between experts/authors of MC generators and users of the programs in the LHC collaborations. The current version of LCG MCDB [1, 2] provides flexible infrastructure to share event samples and keep the files in a reliable and convenient way. It has several interfaces, mainly Web-based, which help to carry out routine operations with the stored samples by users and authors of the samples.

The second motivation behind the project is to create a central database of MC events where stored event samples are publicly available for all groups to use and validate. Often, the same event samples are created by different experimental groups independently several times. If the samples are publicly available and equipped with corresponding and comprehensive documentation it can speed up cross checks of the samples themselves and applied physical models. In many cases it also prevents a possible waste of researcher time and computing resources.

LCG MCDB can particularly be useful in tasks where preparation of an event sample requires specific knowledge of the Monte-Carlo codes/techniques applied, significant computing power, and/or constant interaction authors of the events. For instance, this situation can arise if we use such MC programs as ALPGEN, CompHEP, GRACE, or MadGraph. Even MC generators as PYTHIA or HERWIG sometimes require keeping of event files themselves. Examples of this sort happen in simulations of rare processes and/or with strong pre-selection cuts. In order to simplify tasks mentioned above, LCG MCDB development was initiated some time ago [3, 4, 5].

With the introduction of LHEF event files format [6], we have a common general format for representation of simulated events, which has become a generally accepted tool for the community of developers of MC generators. But LHEF has a limited power to accommodate the events with meta-data (detailed physical description of sample and generation parameters). Keeping the meta-data within event samples are extremely useful for correct processing, verifying and re-usage of the event samples. Now the only project proposed unified meta-data description and storage, is HepML [7, 8, 9].

HepML project is an effort to state a unified extensible way of MC events description and provide program libraries to work with such meta-data. The main goal of the project is to store all possible information from MC generators in XML view, as well as to store generator input parameters and setup. In addition, HepML block is an allowed part of LHEF standard event file header, so keeping events' meta-data as a HepML document inside LHEF header is natural standard way. In the next sections we describe the LCG MCDB and HepML design and ideas in more detail, and briefly portray subsystems and modules of LCG MCDB.

At present, LCG MCDB is a stable software package and ready to use for the LHC community. A dedicated web server is deployed at CERN. There is an application programming interface for program access to MCDB content using HepML. An XML Schema and a program tool to handle (create, validate and parse) HepML documents are publicly available.

2. LCG MCDB as a knowledgebase of Monte-Carlo simulated events

Knowledgebase is a special kind of database for knowledge management. It provides the means for the computerized collection, organization, and retrieval of knowledge [10]. According to the definition one of the specific features of knowledgebase is that it keeps meta-data, i.e. information on data. Usually it is not possible to strictly distinguish between data and meta-data, since the separation depends on situations where the data are exploited. In our concrete case we discriminate between events, as sets of particle 4-momenta (data), and information describing the events as the whole event sample (meta-data). Meta-data form the main contents of MCDB. In this sense, MCDB can hold a path to an event sample only and the sample itself can be located somewhere else. MCDB interfaces provide the means to manipulate with event meta-data.

Comprehensive description of an event sample requires a lot of information, which should be entered to the database. However, in practice, in this specific application area a large part of the information is common for lots of samples. For example, author can re-use any pre-entered information from MCDB, or can create his/her own event description based on already entered information. The second idea behind the current design of MCDB is that MCDB is an area for interaction between two different communities, producers of events and consumers of the events. The latter users are end-users and the former users are "authors". Any physicist who feels his/her sample is worthy to be kept in MCDB can make a request to open a new author account on the MCDB server. It means that MCDB does not assume to have a special team of event producers to prepare events according to end-user requests.

3. LCG MCDB system description in brief

This section briefly describes Web-concerned subsystems of LCG MCDB and software technologies adopted in it. The current version of LCG MCDB is based on the following technologies: WWW, CGI, Perl, SQL, XML, CASTOR [11], and Grid. The main storage of event files is based on tape robots at CERN with access via CASTOR. The MCDB software is organized as a set of Perl modules with the possibility of installing and customizing the software on other sites. All of the MCDB software has been developed from scratch and is available publicly in LCG CVS.

The main unit of MCDB is an article, a document describing one or several event samples. MCDB contents is divided into several categories, i.e. a set of articles concerning a particular type of physical process (e.g. Top physics, Higgs physics) or by theoretical model involved (e.g. supersymmetry, extra dimensions). Each category has its own branch in the main MCDB Web tree graph. The access system in MCDB reminds of a classical system used in the usual Internet forums or newsgroups. There are three different types of permissions to access MCDB. The end-user access is reserved for users who are interested in downloading or making comments to already published event samples or requesting a new event sample. The author access is reserved for authorized users (MC experts) and requires registration on the main Web site. Only registered authors can upload and describe new event samples. The moderator access is reserved for users who manage author profiles and are responsible for general monitoring of information uploaded.

4. HepML and MCDB API to collaboration software

Apart from the MCDB server, LCG MCDB team provides application programming interfaces (APIs) specific to the simulation environments of the LHC collaborations. The main idea of these subsystems is to develop a set of routines for the collaboration software which would give a direct access to event files in LCG MCDB for the MC production on computing farms. In order to allow a possibility for automatic access to event sample description, MCDB team provides an API based on XML representation of event sample meta-data [12]. The current version of the API is developed as a C++ library, which can be added to collaboration software. The XML output from LCG MCDB is based on the HepML specifications (XML Schema).

In the future, some emphasis will be put on the development of extensions of API specific to the automatic uploading of HepML information and event samples to MCDB. This development will be carried out in the context of the HepML project.

5. Conception of completely automated Monte-Carlo simulation chain

In this section we describe a method of making a uniform Monte-Carlo simulation chain based on MCDB and HepML. Events are generated with a Matrix Element MC generator, stored in MCDB, processed with a Shower and Decay MC generator, and then passed to the Detector simulation software. Handling all the data in automatic way allows a user to avoid most of human-related errors and can save researcher's time and keep simulated event files in order.

Firstly, data from a Matrix Element generator are kept in the standard LHE format. Detailed sample's description in HepML form is included to the LHEF header (libraries to write the HepML code are provided). After this step MC generator provides self documented event sample with the full description of simulation inside it. At the moment, CompHEP generator can provides extended information in the form of HepML code inside the standard LHEF header [13].

The next step is to store the sample in a public place. LCG MCDB allows automatic upload and documentation (for several types) of such event files.

Then, the events can be taken via a Grid interface or directly from CASTOR at CERN or through the web interface. Both URLs to the samples and its' detailed description are provided by LCG MCDB in the HepML form using a unique description number (called an article number) as a reference.

After getting the files, they can be transferred to the next generation level, Showering and Hadronization, along with the full meta-data set. Then, stored, for example, in the HepMC format, the data can be processed by Detector Simulation software.

By now, this conception is accomplished in CMSSW within the CMS experiment software environment. Our HepML libraries were adopted to be partially included to CMSSW. This libraries are responsible for parsing HepML response from LCG MCDB.

6. Conclusion

MCDB is a special knowledgebase designed to keep event samples for the LHC experimental and phenomenological community. Now LCG MCDB is ready for use by the community and by other groups of researcher.

In addition to the service, MCDB team has prepared an API [14] for the LHC collaboration software environments. Implementation of the API to the collaboration software environments gives a possibility to use MCDB as a native storage of external event samples in large-scale productions in the collaborations. Subsequent development of the software will rely on further standardization of event file formats and elaboration of the HepML specifications and software.

It is possible to make an automated Monte-Carlo simulation chain, partially based on usage of HepML and LCG MCDB. This way is shortly described in this article and already realized and officially used in CMS experiment.

7. Acknowledgments

This work was partially supported by the RFBR grant 07-07-00365-a. We also acknowledge the LCG collaboration for support and hospitality at CERN. Participation of A. S. in the project was partly supported by the UK Particle Physics and Astronomy Council.

References

[1] S. Belov et al., *LCG MCDB - a Knowledgebase of Monte Carlo Simulated Events*, Computer Physics Communications, Volume 178, Issue 3, 1 February 2008, p. 222 [hep-ph/0703287]

[2] S. Belov, L. Dudko, A. Gusev, A. Sherstnev, *LCG MCDB* `http://mcdb.cern.ch`

[3] M. Dobbs et al., *QCD/SM Working Group: Summary Report*, Les Houches 'Physics at Tev Colliders 2003' [hep-ph/0403100]

[4] P. Bartalini, L. Dudko, A. Kryukov, I. V. Selyuzhenkov, A. Sherstnev and A. Vologdin, *LCG Monte-Carlo data base* [hep-ph/0404241]

[5] L. Dudko, A. Sherstnev, *CMS MCDB* `http://cmsdoc.cern.ch/cms/generators/mcdb`

[6] J. Alwall et al.,*A standard format for Les Houches event files*, Comput. Phys. Commun. 176, 300 (2007) [hep-ph/0609017]

[7] A. Sherstnev, *HEPML: proposal for a structure of partonic events files* `http://agenda.cern.ch/fullAgenda.php?ida=a035826`

[8] S. Belov, L. Dudko, A. Sherstnev, *HepML TWiki Page* `https://twiki.cern.ch/twiki/bin/view/Main/HepML`

[9] A. Buckley et al., *CEDAR HepML Web Page* `http://projects.hepforge.org/hepml`

[10] Wikipedia, *Knowledge Base article* `http://en.wikipedia.org/wiki/Knowledge_Base`

[11] J. P. Baud, B. Couturier, C. Curran, J. D. Durand, E. Knezo, S. Occhetti and O. Barring, *CASTOR status and evolution*, In the Proceedings of 2003 Conference for Computing in High-Energy and Nuclear Physics (CHEP 03), La Jolla, California, 24-28 Mar 2003, pp. TUDT007 [cs.oh/0305047].

[12] S. Belov, L. Dudko, A. Sherstnev, *HepML XML Schema* `http://mcdb.cern.ch/hepml/schemas`

[13] A. Sherstnev et al., *CompHEP Monte-Carlo generator, status report for the version 4.5*, in proceedings of ACAT08 conference, PoS(ACAT08), to be published

[14] S. Belov, L. Dudko, E. Galkin, A. Sherstnev, *MCDB and HepML API libraries* `http://mcdb.cern.ch/distribution`