LCG GENERATOR

P. Bartalini*, University of Florida, FL 32611-8440 Gainesville, USA.
L. Dudko, A. Sherstnev, A. Vologdin, SINP MSU, 119992 Moscow, Russia.
M. Kirsanov, INR RAS Troitsk, 117312 Moscow, Russia.
S. Makarychev, I. Seluzhenkov, ITEP, 117259 Moscow, Russia.
S. Belov, S. Korobov, V. Uzhinsky, JINR, 141980 Dubna, Russia.
F. Ambroglini, Perugia University, Perugia, I-06100, Italy.
J. Cuevas Maestro, H. Naves, Cantabria University, E-39005 Santander, Spain.
F. Moortgat, A. Ribon, CERN, CH-1211 Geneve 23, Switzerland.

Abstract

In the framework of the LCG simulation project, we present LCG Generator, a sub-project launched in 2003 under the oversight of the LHC Monte Carlo steering group.

INTRODUCTION

The mandate of the LCG Generator project is to collaborate with Monte Carlo (MC) generators authors and with LHC experiments in order to prepare validated LCG compliant code for both the theoretical and experimental communities at the LHC, sharing the user support duties, providing assistance for the development of the new object oriented generators and guaranteeing the maintenance of the older packages on the LCG supported platforms.

Contact persons for most of MC generator packages relevant for the LHC and representatives for all the LHC experiments have been agreed. Other candidate contact persons are currently being approached.

Four different work packages (WP) have been defined in the project:

WP1) Generator services library and new Object Oriented MC generators;
WP2) Storage, event interfaces and particle services;
WP3) Shared generator level event files: production and MC data base;
WP4) Validation and tuning.

The responsibilities in LCG Generator are currently shared in the following way: coordination (University of Florida); generator library (MSU, ITEP, Troitsk, Protvino, CERN); event interfaces (MSU, CERN); MC data base (MSU, ITEP, JINR); Production (CERN, Santander).

On top of the activities developed internally, the coordination with other external projects allows to cover the entire spectrum of topics outlined in the work packages: validation (CEDAR/JETWEB), new object oriented MC generators (SHERPA, THEPEG, PHENOGRID/HERWIG++).

This document describes the status and the development guidelines of the two main activities of the project, i.e. the MC generator services library (GENSER) and the MC generator data base (MCDB).

THE GENERATOR SERVICES LIBRARY (GENSER)

GENSER [1] is the central code repository for MC generators and generator tools. It was the first CVS repository in the LCG simulation project. The sources and the binaries are installed in AFS and the tarballs are made available by the Software Process & Infrastructure group (LCG-SPI). This new library is going to gradually replace the obsolete CERN library for what concerns the MC generators support. In fact GENSER is currently used in the ATLAS Data Challenge (DC2) and is tested by all the other LHC experiments. CMS and LHCb should adopt GENSER in the official production from the next Data Challenges.

Three different solutions are available for the inclusion of MC generator sub-packages in the LCG Generator Library:

1) To develop the MC sub-package in GENSER as an independent module;

2) To fully export the generator package in GENSER reorganising the directory structure according to the LCG policy and defining the corresponding sub-package. CVS tags are used to keep track of the sub-packages versions. (Solution adopted for most of the installed packages);

3) To install the generator package as external software in the LCG environment and to store in GENSER just tests suites and other related code.

For each MC package, an ad-hoc solution is found taking into account the authors directives and the user requirements. Top priority and second priority packages pursued for inclusion in the generator library have been indicated in the report of the RTAG 9 working group [2].

GENSER is fully independent from other large libraries and currently follows a quarterly release scheme. Quick bug fixes and special versions can be produced under request. Most of the MC sub-package versions produced by the authors are installed. Old versions are maintained as long as they are used.

* on behalf of the LCG Generator team.

Status of GENSER

At the moment GENSER is considered to be at the Beta stage. Configuration and build systems for librarian and end users are based on the SCRAM technology [3]; future versions of GENSER will support Makefiles as well.

The current beta version of GENSER (0.2.0) supports both shared and static libraries for two different platforms (rh73_gcc32 and rh73_gcc323) and includes the following MC generator packages:

- PYTHIA [4] versions 6.205, 6.217, 6.220, 6.221, 6.222, 6.223, 6.224 and 6.304;
- HERWIG [5] versions 6.500, 6.503, 6.504, 6.505;
- JIMMY [6] version 4.1;
- ISAJET [7] versions 7.67 and 7.69;
- EVTGEN [8] version alpha-00-11-07;
- EVTGENLHC [9] version 1.0;
- ALPGEN [10] version 1.3.2;
- COMPHEP [11] version 4.4.0;
- LHAPDF [12] versions 1.1 and 2.0;
- PDFLIB [13] version 804;
- PHOTOS [14] versions 202 and 207;
- GLAUBER Xs [15] version 1.0;
- HIJING [16] versions 1.36, 1.37 and 1.383.

GENSER Validation

The event generator validation is divided in two different parts:

1) Basic sanity checks;
2) Physics validation.

The basic sanity checks are currently performed in a standalone way. The code is provided by the authors, beta testers and librarians and it is stored under the test and example directories of the GENSER sub-packages. It will be subsequently integrated with the simple generator level production framework (developed in WP3).

The physics validation will be performed with JETWEB [17], assuming that it will integrate GENSER in a reasonable time scale (i.e. by the third quarter of 2005). The independent similar projects are strongly encouraged to merge with JetWeb contributing to increase the library of validation code (currently supporting only Pythia and Herwig).

GENSER: Future Plans and Collaboration to the Development of the New Object Oriented MC Generators

Work has already started for the preparation of the first version of GENSER that will be declared of production quality (1.0.0) to be released by the end of the year 2004. It will extend the support to the Scientific Linux platform [18] and it will contain most of the top priority and second priority packages indicated in the RTAG 9 document.

From GENSER 1.0.0 the management of the GENSER releases will be improved, providing appropriate documentation for the development, technical and user support duties and the corresponding task assignment. The full compliance to the LCG standards will be guaranteed. A specific GENSER FAQ will be developed.

In order to favour the adoption of the new object oriented MC generators in the experiment simulation frameworks, the LCG Generator project will share some responsibilities on the development and maintenance of the Toolkit for High Energy Physics Event Generation (THEPEG). THEPEG [19] was initially developed by L.Lonnblad at Lund University. It was broken out from Pythia 7 [20] to better factorize the parts which are Pythia-specific from those which are general model-independent components of the toolkit and which can be used by any event generator model.

LCG Generator has set a common milestone with the PHENOGRID [21] initiative for the third quarter of 2005: the first test of ThePEG and EvtGenLHC integration in Herwig++ (and the possible preliminary inclusion of Herwig++ in GENSER).

The next important milestone deals with the inclusion of the first object oriented general purpose MC generator in GENSER. SHERPA [22] has been identified as a possible candidate. Work has already started with the authors and should be finalized in the first quarter of 2005.

THE MONTE CARLO DATA BASE (MCDB)

In this paragraph we review the main ideas behind MCDB and discuss future plans to develop this data base within the LCG framework.

MCDB [23] is a public database for the configuration, book-keeping and storage of the generator level event files. Access from individual users as well as from the LHC and LCG simulation frameworks will be granted.

The goal of MCDB is to improve the communication between Monte Carlo experts and end-users; this tool is most likely to become essential to perform comparisons and combinations in LHC Working groups.

One of the most frequent problems for the experimental high energy physics community is MC simulation of physics processes. There are several publicly available MC generators; however, the correct MC simulation of complicated processes requires in general rather sophisticated expertise on the user side. A physics group in an experimental collaboration ofen requests experts and/or authors of MC generators to produce MC samples for a particular process. Furthermore, it is common that the same physics process is studied by various physics groups needing the same MC event samples. One of the goals of MCDB is to make MC event samples prepared by experts publicly documented and available to various physics groups.

MCDB History

Historically, PEVLIB [24] can be considered the first working prototype of MCDB. PEVLIB was established at CERN on the AFS file system. This database provided

COMPHEP parton level events for CMS, but it was lacking of a dedicated user interface. It was rather built as a set of directories where event samples were stored. Documentation for the samples consisted of ASCII README files located in the same directories as the event files.

The second version of MCDB was set up at Fermilab. This database was split in two independent parts:

1) MC event files, stored via the FNAL tape system ENSTORE;
2) Documentation for the events, publicly available via the Web [25]

The latest version of MCDB was developed and deployed within the CMS collaboration [26]. This database includes web interfaces both for event files (enabling download and upload) and documentation. Its main goal is to store parton level files. The PEVLIB recently migrated to the CMS MCDB.

However, we can identify some potential weakness in CMS MCDB that motivates further developments:

1) The CMS MCDB was designed to store only parton level events. Typically the size of these samples is smaller than 100 M Bytes;
2) Only the most interesting and sophisticated requested processes are contributed to the data base;
3) CMS MCDB does not support a SQL engine; therefore the database can process keyword phonetic queries only. Search based on meta-data is not possible.

The next version of MCDB is intended to be used by both the experimentalist and theorist communities at the LHC, and it is now under development in the context of the LCG Generator project. The development will be driven by the need to solve the problems described above.

The LCG MCDB will provide persistent storage of shared event samples with convenient public interfaces for MC users, MC experts and MC authors.

The main requirements for the LCG MCDB are the following:

1) It should be based on SQL, allowing to keep deeply structured information and to treat sophisticated search queries;
2) It should support storage of both parton level and particle level events;
3) It should allow for dynamic access to the data through simple Web interfaces or via application programming interfaces (API).

The HEPML Meta Data Format

In the LCG MCDB the events will be stored using the HEPML format (developed in WP2), where the information is sub-divided in two parts:

1) The header, that contains the general information concerning the event sample, i.e. author, creation date, collider description, generator specific data, generation cuts, other physical parameters, parser directives etc.
2) The event records, i.e. the variable data of events written in some compact format to one string (particle momenta, colour chains etc.)

The header is stored in a txt file with XML Syntax. The event records are zip-compressed and attached to the header file. The HEPML meta-data format provides the basis to the SQL search in MCDB.

MCDB: Elements and Terminology

The following elements and terminology have been agreed for the LCG MCDB data base:

- *Event sample*, i.e. the data stored in HEPML format;
- *Article*, i.e. the document describing a set of samples. Articles are organized in a hierarchy tree structure. The branches of the trees are defined to be the categories;
- *Author*. The author uploads the shared event samples, provides a formal validation and can add, modify, or delete any information about the samples themselves;
- *User*. The user has access to the MC event samples, to their descriptions and can submit comments;
- *MCDB License*: agreement between an author and the users about the event sample available in MCDB;
- *Moderator*. The moderator manages (adds, removes or modifies) authors and user profiles in MCDB;
- *Administrator*. The administrator takes care of the technical aspects of MCDB.

The interfaces of LCG MCDB are shown in Figure 1.

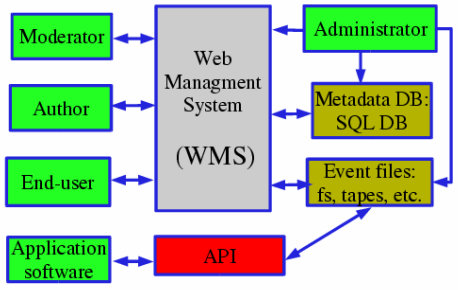

Figure 1: Interfaces of LCG MCDB.

MCDB: Software Evaluation

Most of the operations supported by MCDB (registration and management of user/author profiles, event download/upload, posting/retrieving of articles and comments) will be available to users and authors as well as to the moderator through dedicated Web interfaces. It is therefore essential to choose an appropriate Web Management System (WMS).

Although there are many WMS on the market (WebGUI, Metadot etc.), most of them are created for

developing of Web portals, no specific software fulfilling only the specific MCDB requirements can be identified. Existing toolkits might have to be customized with unknown spin-offs for what concerns the maintenance and the license issues. After a careful study of the options available, we have decided to develop our own WMS based on the Perl scripting language.

After a comparison of different SQL databases we have decided to adopt MySQL [27] as SQL DBMS for LCG MCDB. MySQL offers many advantages for our project with respect to other SQL database systems (Oracle, PostgreSQL, etc.): it is supported by the LCG-SPI, it is freeware and cross-platform, it has an already developed large library of GUI applications, and it turns out to be reliable, stable and well supported.

Status of MCDB

The complete SQL tables' structure has been designed. Most of the internal Perl modules (access to the MCDB SQL tables, parse configuration file, etc.) have been written. An alpha version of the MCDB Web server is running [28] that for the time being supports only the Mozilla browser. Work is in progress to extend to the other most popular Web browsers (Netscape, Internet Explorer etc.).

In the user area, a Web interface for the creation and editing of articles is available. File uploading to hard disk is supported. Work is already started to support CASTOR [29] as well. In the administrative area, interfaces for the management of profiles and articles have been achieved.

MCDB: Future Plans

A first working MCDB prototype should be achieved by the end of the year 2004 (short term milestone). A test will be performed with the full chain of possible MCDB actions through the Web interface.

In this period he main effort will be dedicated to provide MCDB documentation and to complete the definition of the HEPML set of XML tags. Authorization procedures through login and GRID certificates will be enabled as well.

In the mid-term time scale (i.e. during 2005), MCDB will be completed with: search engine; moderator interface, HEPM parser, logging system. In the long-term time scale (i.e. from 2006), more enphasis will be put on the development of dedicated APIs to access MCDB from external simulation frameworks.

ACKNOWLEDGEMENTS

We would like to thank the LCG-SPI team, as well as our contact persons in LHC experiments and MC generator projects: G.Corti (CERN, LHCb), A.De Roeck (CERN, CMS), I.Hinchliffe (LBNL, ATLAS), F.Krauss (Dresden University, SHERPA), L.Lonnblad (Lund University, THEPEG), M.Mangano (CERN, ALPGEN), F.Moortgat (CERN, CMS), A.Morsch (CERN, ALICE), F.Ranjard (CERN, LHCb), P.Richardson (University of Cambridge, HERWIG), A.Ryd (Cornell University, EVTGEN), A.Sherstnev (Moscow State University, COMHEP), T.Sjöstrand (Lund University, PYTHIA), G.Stavropoulos (LBNL, ATLAS), B. Waugh (University college of London, CEDAR/JETWEB), V.Uzhinsky (JINR, GLAUBER XS), M.Whalley (University of Durham, LHAPDF).

REFERENCES

[1] http://lcgapp.cern.ch/project/simu/generator/
[2] J.P. Wellisch et al., Proceedings of the International Conference on Computing in High-Energy Physics: CHEP'03, La Jolla (CA), USA, 2003.
[3] http://lhc-monte-carlo.web.cern.ch/lhc-monte-carlo/
[4] T.Sjöstrand et al., hep-ph/0108264 and hep-ph/0308153.
[5] B.Webber et al., Journal of High Energy Physics (2001) 0101:010.
[6] J.M. Butterworth et al., hep-ph/9601371.
[7] E. Paige et al., hep-ph/0312045.
[8] D.J. Lange et al., Proceedings of the International Conference on Computing in High-Energy Physics: CHEP'98, Chicago (IL), USA, 1998.
[9] http://lhcb-comp.web.cern.ch/lhcb-comp/Simulation/evtgen.htm
[10] M.Mangano et al., Journal of High Energy Physics 0307:001, 2003.
[11] E. Boos et al., hep-ph/0403113.
[12] http://durpdg.dur.ac.uk/lhapdf/
[13] H. Plothow-Besch, Comput. Phys. Commun. 75: 396-416, 1993 (No.3).
[14] E. Barberio, Z. Was, Comput. Phys. Commun. 79: 291-308, 1994.
[15] K. Abdel-Waged et al., Journal of Physics G26: 1105-1115, 2000.
[16] X-N. Wang, M. Gyulassy, Phys. Rev. D44: 3501-3516,1991.
[17] J. M. Butterworth et al., Comput. Phys. Commun. vol. 153/2: 164-178, 2003.
[18] https://www.scientificlinux.org/
[19] http://www.thep.lu.se/ThePEG/
[20] L. Lonnblad, Comput. Phys. Commun. vol. 118: 213-228, 1999.
[21] http://www.phenogrid.dur.ac.uk/
[22] F.Krauss et al., hep-ph/0311263.
[23] L. Dudko et al., hep-ph/0404241.
[24] V.A.Ilyin et al., hep-ph/0101232.
[25] http://www-d0.fnal.gov/~dudko/mcdb
[26] http://cmsdoc.cern.ch/cms/generators/mcdb
[27] http://www.mysql.com
[28] http://mcdb.cern.ch
[29] J.P. Baud et al., Proceedings of the International Conference on Computing in High-Energy Physics: CHEP'03, La Jolla (CA) , USA, 2003.