LCG Monte-Carlo Data Base
(LCG Generator Services Subproject)

P. Bartalini[1] *L. Dudko*[2], *A. Kryukov*[2], *I. Seluzhenkov*[3] *A. Sherstnev*[2], *A. Vologdin*[2]
[1]CERN, Geneva, Switzerland,
[2]Moscow State University (MSU), Moscow, Russia,
[3]Institute for Theoretical and Experimental Physics (ITEP), Moscow, Russia.

arXiv:hep-ph/0404241 v1 27 Apr 2004

Abstract
We present the Monte-Carlo events Data Base (MCDB) project and its development plans. MCDB facilitates communication between authors of Monte-Carlo generators and experimental users. It also provides a convenient book-keeping and an easy access to generator level samples. The first release of MCDB is now operational for the CMS collaboration. In this paper we review the main ideas behind MCDB and discuss future plans to develop this Data Base further within the CERN LCG framework.

1. Problem description

One of the most general problems for the experimental high energy physics community is Monte-Carlo (MC) simulation of physics processes. There are numerous publicly available MC generators. However, the correct MC simulation of complicated processes requires in general rather sophisticated expertise on the user side. Often, a physics group in an experimental collaboration requests experts and/or authors of MC generators to create MC samples for a particular process. Furthermore, it is common that the same physics process is investigated by various physics groups neeeding the same MC event samples. The main motivation behind the Monte-Carlo Data Base (MCDB) project is to make MC event samples, as prepared by experts, available for various physics groups.

There are a number of useful aspects that motivate setting up a central MC Database.

1. Correct and reliable MC event generation of the most processes of interest requires significant expertise. Moreover, most MC generators, in particular those calculating higher order perturbative corrections, require a significant amount of computer and human resources. By means of MCDB, samples prepared by experts can be distributed easily and used as many times as needed.
2. Public availability of common event files helps to speed up the validation procedure of the events.
3. A central and public location where well-documented MC events can be found would be very useful. It would also allow rapid communication between authors of MC events and their users.
4. The same MC samples for Standard Model processes can be used for multiple purposes, e.g. to study backgrounds to various new physics processes.
5. Files containing detector and beam-related backgrounds can also be kept in a common location.

Historically, the first MCDB – PEVLIB [1] was established at CERN on the AFS file system. This database provided CompHEP [2] parton level events for CMS, but was lacking of a special interface for users. It was rather built as a set of directories where event samples were stored. Documentation for the samples consisted of ASCII files (README) located in the same directories as the event files.

The next version of MCDB was established at Fermilab. This database was split in two independent parts:

- MC event files, stored via the FNAL tape system ENSTORE [3].
- Documentation for the events, publicly available via the Web [4].

The latest version of MCDB [5] was developed and deployed in the CMS collaboration – CMS MCDB [6]. This database includes web interfaces both for event files (enabling download and upload) and documentation. Its main goal is to store events, only at the parton level, generated by MC experts, to be used by the LHC community. Note that all the files from PEVLIB have been moved to CMS MCDB.

However, we can identify several potential weakness in CMS MCDB that motivate further developments of this project.

- CMS MCDB was designed to store parton level events. This implies that the size of event files should not be too large (typically smaller then 100 Mb).
- CMS MCDB does not support a SQL engine. Therefore this database can process keyword phonetic queries only. Complex search is not possible.
- Support for few important processes is guaranteed at the moment, however the LHC experiments need to use further processes (a multiplicity of several hundreds is estimated for LCG MCDB).

These aspects do not limit CMS MCDB at present. However, we expect that in a few years users will request a more powerful MCDB where these restrictions will be removed.

The next version of MCDB (to be used by all the LHC collaborations) is now under development in the context of the LCG Generator subproject [7], an Application Area activity developed in the framework of the simulation project. The new MCDB has to provide persistent storage of event samples with convenient public interfaces for users from the LHC community and experts or authors of MC generators. The main requirements for the new version of MCDB are the following:

1. LCG MCDB is based on a SQL database, therefore it is possible to keep deeply structured information and to treat sophisticated search queries.
2. The database should store both events with partons in the final state (partonic events) and events after hadronization and all decays (particle events). It also can keep some other types of events.
3. Users have access to the MC event samples and descriptions of the samples via Web interfaces.
4. Users may publicly discuss MC samples.
5. Formal validation of MC samples by experts.
6. An authorized user (author) may add, modify, or delete any information about his(her) own event samples or the samples themselves dynamically by a simple Web interface.
7. Application software should have a programming access to the samples.
8. New methods of event files uploading (in addition to the existing methods in CMS MCDB) should be implemented. These methods would be based on GRID technologies.

2. General conception and terms

LCG MCDB uses ideas and experience of CMS MCDB [5]. We will use the following notations in the note (they are marked out by bold).

Event File (sample) - a file containing particle or partonic events. These files consist the main contents of MCDB.

Article - a document describing a set of samples. The main task of the article is to provide comprehensive information about event samples (connected with the article) supplied by an author. The article is written via the Web interface and is freely available via the Web on the MCDB Web site.

MCDB License - agreement between an author and the end-users about event samples available via the MCDB Web site.

In the LCG MCDB conception we subdivide all the information about events into two types:

Event Meta-data - information which describes all events in a sample *in general* (e. g. beam description, physics parameters, applied cuts). The meta-data store SQL tables of MCDB.

Event Data - the events themselves (the event data are stored into event files).

These types of the information do not have clear limits. For instance, number of particles in a partonic event sample can be considered as meta-information. But, after hadronization, the number of particle will change from one event to another and this parameter, certainly, does not belong to the meta-data. However, a particular sample always can be subdivided into meta-information and event information. The meta-data provides a basis to the SQL search in MCDB.

There are 5 different access ways to MCDB in the LCG MCDB conception. Each way has specific rights and restrictions. We will describe these access methods in terms of “users“. These “users” can be both real persons and software modules.

End-User - a user who works with MCDB (search, read, download) via Web interfaces. For the downloading of event samples the end-user should register on the MCDB Web site and accept MCDB License.

Author - an authorized user, which can create and modify his/her own articles and upload new event samples.

Moderator - an authorized user which manage (add, remove, modify) authors profiles in MCDB. Moderators may be appointed

Administrator - system administrator of MCDB.

Application software - a part of the experimental software, which has an access to event files in MCDB via special API’s.

3. Main LCG MCDB components

Main components of LCG MCDB are presented in the figure 1.

Event files are kept on a mass storage file system, a plain or a distributed file system.

All meta-data are written to SQL tables of Event Meta-Data DB. This database will work under control of one of SQL DBMS (MySQL, PostgreSQL, or Oracle).

WMS (Web Management System or Content Management System) provides an access of

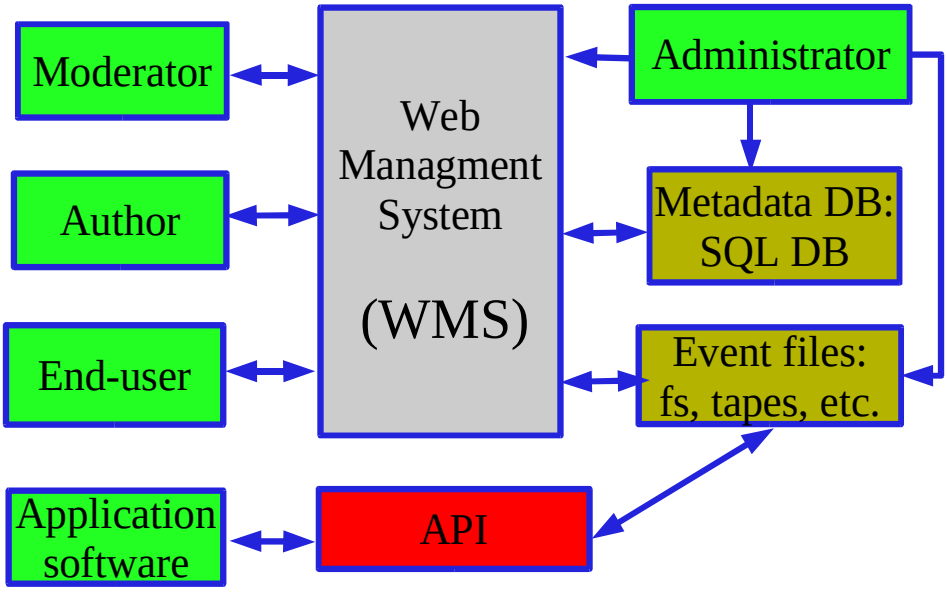

Fig. 1: The CERN $\bar{p}$ complex.

end-users, moderators, and authors to read and write articles on the LCG MCDB Web site.

API (Application Programming Interface) is a library which provides a direct programming access of application software (for instance simulation software specific to a collaboration) to the event samples.

4. MCDB interfaces

All interfaces of LCG MCDB are shown in the figure 2. End-users, authors and moderators work with MCDB via the Web interfaces. All these interfaces are a part of WMS.

The end-user Web interface consists of several parts:

- Search form. Users mine data from LCG MCDB by filling a Web form. SQL requests based on the user requests are formed and sent to Event Meta-Data DB.
- Registration. A new end-user can register by filling out a special Web form. To successful register, an user should accept the MCDB License, which defines conditions of working with event samples in LCG MCDB. After that (s)he obtains a login name/password to access to the MCDB download area.
- Download samples. A registered end-user can download the samples to local machines.
- Comments on articles. A registered end-user can add comments on articles. A new comment is published on the Web. MCDB will have possibility to inform the article author automatically about the new comment by email.
- Web navigation. Articles are organized in a hierarchy tree structure on the MCDB Web site. The branches of the trees are defined to be the categories. End-user can navigate through this hierarchy.

Each author has own profile in WMS and should pass an authorization mechanism in WMS, if (s)he would like to work in WMS. After authorization, the author can work with his/her own articles (create, modify, publish or interdit the access to articles in the LCG MCDB

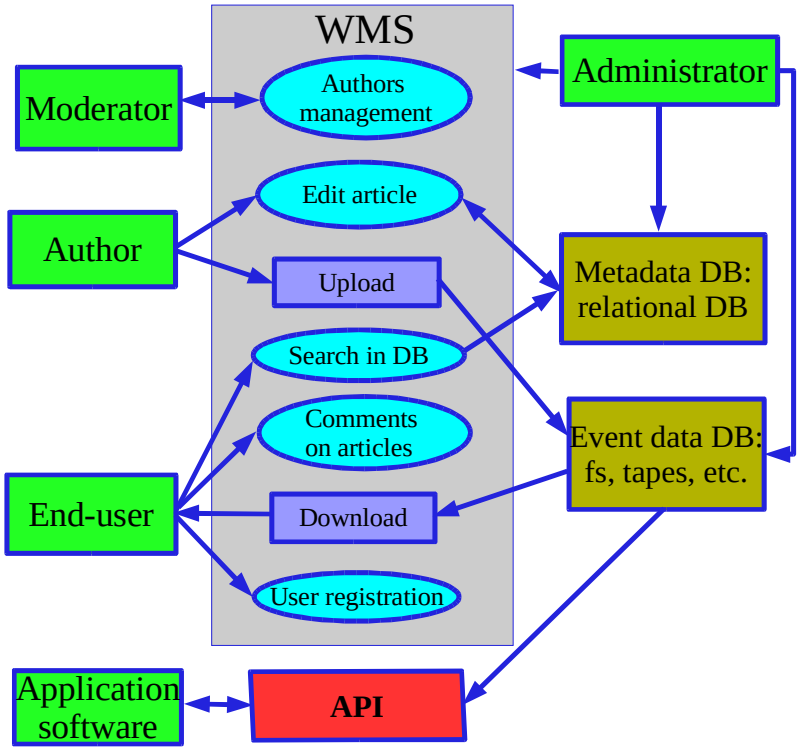

Fig. 2: MCDB interfaces to users and software

Web site). The author can upload new event files via the Web interface. The LCG MCDB Web site also has a special Web form to register new authors.

The moderator Web interface allows to manage (create, remove, modify) the author's profiles and articles.

Administrator supports WMS, Event Meta-Data DB, and event files.

The application software has an access to event samples in LCG MCDB directly (for instance, by ftp or gridftp). The API organizes such connections. By default the API can download a full event sample. Application software can request a particular number of events.

4.1 WMS technical description

This section describes the future internal structure of WMS. We plan to develop our own variant of WMS. Although there are many WMS on the market, most of them are created for developing of Web portals (for example, for on-line stores or entertainment industry). No specific software fulfilling the only the MCDB requirements can be identified. Toolkits may have to be customized.

e can not find a specific software with all (and only) functionality necessary for our goals. Therefore, we need to customize the toolkits (in many cases very deeply).

Our originally developed code for WMS will consist of separate blocks, which will perform a specific role in WMS. Certainly these separate blocks can be adopted from the existing software of the CMS MCDB or from other WMS (WebGUI, Metadot, etc.).

Now we mark out nine isolated and independent blocks which can be realized as a set of subroutines (Perl scripts) with known input and output.

1. Authorization. This block sets permissions for each user according to the user login and password.
   numb_mode = general_ACTH(login, password)
   where **num_mode** could be:
   0 (administrator),
   1 (moderator),
   2 (author, with login name confirming the permission of the particular author),
   3 (end-user, after the "License Agreement").
   As in CMS MCDB this block will be based on the CERN AFS passport base.
2. Authors management. This block manages the information about the MCDB authors and can consist of three requests (with interfaces via Web forms):
   author_create(id,'login_name','Last name','First Name','e-mail',...)
   author_modify(id,'new_login_name','new_name','new_e-mail',...)
   author_delete(id)
3. Meta-data management. Authors should save meta-data about event files in SQL tables of LCG MCDB. After that post management block creates an article from the data. Basically the meta-data management system has to allow to store, modify and remove the meta-data in the SQL database. Since the data are very closely related with articles we will give prefix "article" to the scripts in the block. These scripts will be realized as Web forms. In the future we plan to develop a procedure of automatic parsing of event files to fill some fields in the Web forms. Certainly, this operation will apply some requirements on format of the event files:
   article_create(article_id,'category_id1:category_id2:...','author1:author2:...',...)
   article_modify(article_id,'category_id1:category_id2:...','author1:author2:...',...)
   article_delete(article_id)
   article_parser(article_id)
4. Category management. This block manages categories and sub-categories. According to the goals of LCG MCDB, we intend to organize the categories in several hierarchical levels
   category_create(id,up_category_id,'category_name')
   category_modify(id,up_category_id,'category_name')
   category_delete(id,up_category_id,'category_name')
5. Comments management. This set of subroutines creates and manages users comments to articles.
   comment_create(article_id,'comment_body')
   comment_answer(article_id,comment_id,answer_author_id,'answer')
   comment_modify(article_id,comment_id,answer_author_id,'answer')
   comment_delete(article_id,comment_id)
6. Templates management block. Each article requires a set of information about event sample. Articles in different physics categories can have a different form and the different minimal set of information. This block manages templates which make a pattern to articles in a particular category (or a set of categories) and defines a form how articles will be looked on the Web. There are two subparts in each template. The first part defines the Web form for the submission of an article to a particular category. The second part defines an HTML format for publication of articles from a particular category.
   Functions to work with templates:

template_new(templ_id,'category_id1:category_id2:...','fields')
template_delete(templ_id,'category_id1:category_id2:...')
template_modify(templ_id,'category_id1:category_id2:...','fields')
template_html_modify(templ_id,'category_id1:category_id2:...','html_cod')

7. Post management. An author submits and edits information about event samples to the SQL database by the articles management block. Finally, these information should be re-organized to an article (statically or dynamically, depending on templates) and published on the LCG MCDB Web site. The main purpose of this block is to compile an article from information kept in Event Meta-Data SQL DB and post it to the LCG Web site. This block will allow to remove articles from the site if the author decides to do it.
post_article(article_id,'category_id1:category_id2:...',post/delete)
8. Uploading block. This block manages the files attached to article. There are several methods to upload event samples by using different protocols (HTTP, ftp, ftpgrid). This block also will realize requests to (re)move files according to permissions and storage system optimization procedures.
9. Log system. For security and debug reasons WMS needs to keep different levels of information about transactions in LCG MCDB. The main function of this block is to collect log information (to log files) from all other WMS blocks according to a debug level in a WMS configuration file.

4.2 Choice of SQL DBMS for LCG MCDB

After the comparison of different SQL databases we decided to apply MySQL [9] as SQL DBMS for LCG MCDB. MySQL offers many advantages for our project in comparison with other SQL database systems (Oracle, PostgreSQL, etc.):

- MySQL is supported as standard software in the CERN LCG collaboration, in contrast to PostgreSQL.
- Open-source and free software: MySQL can be used, modified, and distributed by everyone free of charge for any purpose, be it private, commercial, or academic.
- Better support: in addition to strong support offerings, MySQL have a vibrant community of MySQL professionals and enthusiasts whose advice can be used during developing.
- Reliability and stability: unlike many proprietary databases, it is extremely common for MySQL users to report that MySQL has never crashed for them in several years of high activity operation.
- Cross platform: MySQL is available for almost all Unix flavours and for Windows.
- GUI database design and administration tools: several high quality GUI tools exist to both design and administer the database.

We plan to organize communication between WMS and SQL by means of separate library of PERL subroutines. Each block in WMS will correspond to a set of subroutines in this library which will construct the SQL queries from WMS. The library of WMS-SQL interface for the LCG MCDB project will be described comprehensively in future technical documentation for the developers of the project.

5. Unified event meta-data format

The proposed conception of LCG MCDB demands no specific requirements on a format for the event files (it can be ntuples, plain text files, etc.), since this project does not intend to

manipulate with events themselves.

It would be highly useful to write the meta-data information in a unified format. This would allow to treat the event files automatically in LCG MCDB. For example, such format would allow to fill fields in SQL tables of Event Meta-Data DB automatically. It would simplify writing of articles in LCG MCDB and decrease a probability of errors in MCDB and the articles. Unification of event file formats would allow to develop a reliable and simple interface to the experimental environments.

Initial requirements for the format:

- Platform independence;
- An uniform syntax for any types of meta-data stored in the event files.
- The syntax should be extensible (allow to add some new information to files with minimal changes in files).
- Simple I/O and parsing by already existed and well-maintained software tools (XML parsers, etc.).

One of the variants for the format has been proposed in [10].

The only requirements on meta-data contents in event file is "self-description". The meta-data, kept in an event file, should provide understanding what the events are and how users can use them.

6. Meta-data and the query interface model

If an end-user wants to take data from MCDB, at first, (s)he has to compose a query to Event Meta-Data SQL DB to get a list of articles with the requested parameters. There are some different types of interfaces to build this query. We plan to organize a combination of two methods in LCG MCDB. The articles in MCDB will be sorted out to a tree structure of categories. In the ideal situation each category is related with a class of physical processes (as it has been done in CMS MCDB). The end-user will browse the tree and will find the necessary class. The main task here is to choose interesting (for the end-user) category of the processes. Articles in LCG MCDB will not be connected with the categories (i.e. one article can fall into several categories).

When the end-user has chosen a category, (s)he may build a SQL query by simple "language" interface. This search query will be processed just in the frames of the category. Since the information which describes these events is very heterogeneous, the proposed scheme allows to simplify syntax of the interface in the separate categories.

Preliminary list of information which will be available to search in LCG MCDB:

1. **Initial state**: proton-proton, ion-ion, maybe other beam information for some specific samples (machine and beam related backgrounds).
2. **Type of the final state**: partonic level, parton level plus showers, particle level (after hadronization).
3. **Generator specific parameters**: showering and hadronization models and their parameters, initial value of random generator in PYTHIA (or other MC generator), other parameters specific for a MC generator, etc.
4. **Physics which is related to the sample**: for instance, Higgs in MSSM, extra dimensions, gauge bosons production, top quark physics, etc.

5. **Physics parameters for the process**: values of couplings, masses, widths, and other properties of particles (W- or Z-bosons, etc.), CKM matrix elements, and so on.
6. **Formal information**: who, when, and how (by which generator) was created this sample, how many events in the sample, which format was been used, etc.
7. **Applied cuts, the process cross section and cross section errors**.
8. **Direct links to event files**.

The first two points will be encoded in the main tree of the LCG MCDB site. Other information will be collected for SQL queries by the "language" interface.

Also authors may provide extra information in the samples description using the special Web form (some author's free comments about the samples). This information will not be used by the SQL queries of end-users for article searching. It will only get to an article related with the sample as an extra comment.

7. How it works

User interface to LCG MCDB will be organized as a Web site. To search for some article on the Web site, a user browses through the tree of (sub)categories (process classes) and fills out the search Web form. The result of the search will be a list of articles satisfied the queried conditions. After that (s)he can read the selected articles. The article contains direct references to event samples. If the end-user already registered (and accepted the MCDB License), (s)he can download the event samples to a local machine and add comments to the articles. Unregistered users may register directly on the LCG MCDB Web site.

Any author must register by filling out the author registration Web form. By the registration the author also accepts conditions of the MCDB License. The registration of new author should be approved by a moderator. Authorized author can work with his/her own articles:

- To create an article that describes the uploaded event file. First of all, the author uploads a new event sample. The article consists of two part. The first part is filled automatically by parsing event meta-data from the uploaded file. The second part is a comment supplied by the author. If the event sample is written in a not recognized format (by a WMS script), author fills all fields in the form by hands.
- To modify an existing article. The author chooses it from the list of his/her articles. The author can modify, add, or remove comments to the article and upload/remove event samples.
- To make the articles accessible to the Web. The author publishes the articles in the LCG MCDB Web site. After that, the article will be available for the end-users. Also the author can disable any own article in this Web site (but it is still kept in MCDB itself and author can finish or modify it later and publish it again).

Moderator manages author profiles via a special Web interface. Also (s)he can make disable or even remove articles. For example, if the sample or/and an article has bugs the moderator may disable the article temporarily.

Application software have an access to the event data via API directly (e.g. by ftp anonymous session or by a GRID protocol in the future). Incoming data for the API are direct references to the event samples. These references can be obtained from the Web site. By default, API download full event data. if an event sample is kept in a standard and accepted by LCG MCDB format the application software can request a particular number of events.

8. Milestones and requested resources

The duration of proposal is two years. A plan of the first year (by the quarter) is the following.

First quarter:

1. Evaluation of software for LCG MCDB (WMS, SQL DBMS, etc.).
2. Design the LCG MCDB Web site.
3. Design of structure and connections of SQL tables for LCG MCDB.

Second quarter:

1. Development of Meta-data SQL DB (all necessary tables, subroutines to communicate with MySQL).
2. Design and development of scripts for the meta-data management and authorization blocks.
3. Design and development of the author management, comments management, and post management (in a reduced form) blocks.
4. Deployment an customization of all necessary software at the CERN. Deployment of the test version of LCG MCDB.

The result of the first six months will be a test prototype of LCG MCDB. The software will be installed at the CERN and will include a first version of Event Meta-Data DB (all SQL tables) and all parts of WMS (all blocks will provide necessary options for testing only, full planned functionality will realize later on).

Third quarter:

1. Testing of the first LCG MCDB release.
2. Writing of LCG MCDB documentation, including HOWTO's for end-users and authors.
3. Development of a full version of the post management system.
4. Development of a full set of libraries.

Forth quarter:

1. Development of the log system and templates management block.
2. Development of tools for large file uploading – Uploading block.

Tasks for the second year:

1. Creation of event Meta-data parsing mechanism.
2. Development of API for application software.
3. Adaptation of LCG MCDB for LCG persistence system (POOL) and distributed storage systems.
4. Development of HEPML: specification (XML Schema) and parsing software including XSLT.

Manpower: It is difficult to give precise estimation of manpower for the project. Our estimation is 1.5-2 FTE per year. Nevertheless, we propose to start with 0.5 FTE from Russian side plus 0.5 FTE from CERN (1 FTE per year in total) with possible increasing of the manpower after six months.

Hardware and software requirements: Web server with large storage capacity (about 0.5-1.0 TB of local space) under Linux OS. We need installed Apache and MySQL on the machine.

If the server will be under central administration, all MCDB developers have to included to /etc/sudoers with all necessary rights.

Periodicity of reports: Internal review should appear every six months.

References

[1] V. A. Ilyin, PEVLIB: `/afs/cern.ch/cms/physics/PEVLIB/`

[2] E. Boos *et al.*, "CompHEP 4.4: Automatic computations from Lagrangians to events," arXiv:hep-ph/0403113, to be published in the proceedings of IX International Workshop on Advanced Computing and Analysis Techniques in Physics Research December 1-5, 2003. KEK, Japan. 10 pages, 2 figures.

[3] FNAL ENSTORE tape system:
http://www.fnal.gov/docs/products/enstore/enstoreTOC.html

[4] L. Dudko, S. Mrenna, FNAL MCDB: `http://www-d0.fnal.gov/~dudko/mcdb`

[5] M. Dobbs *et al.*, "The QCD/SM working group: Summary report", Proceedings of Les Houches "Physics at TEV Colliders 2003". arXiv:hep-ph/0403100.

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

[7] `http://lcgapp.cern.ch/project/simu/generator/`

[8] CMS MCDB: direct AFS link
`/afs/cern.ch/cms/generators/{mcdb,mcdb2,mcdb3,...}`

[9] MySQL 4.0.18 `http://www.mysql.com`

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