

# First release of LibHepML

S. Belov, JINR, Dubna  
on behalf of MCDB group

LCG MCDB group:  
S. Belov, JINR  
L. Dudko, SINP MSU  
A. Ribon, CERN  
A. Sherstnev, Univ. of  
Cambridge

LCG Generator Services monthly meeting,  
CERN, 9 April 2008

# Outline

- What is libhepml
- Why libhepml
- Main features
- Examples
- libhepml in real projects
- Further developments
- Conclusion

# What is libhepml

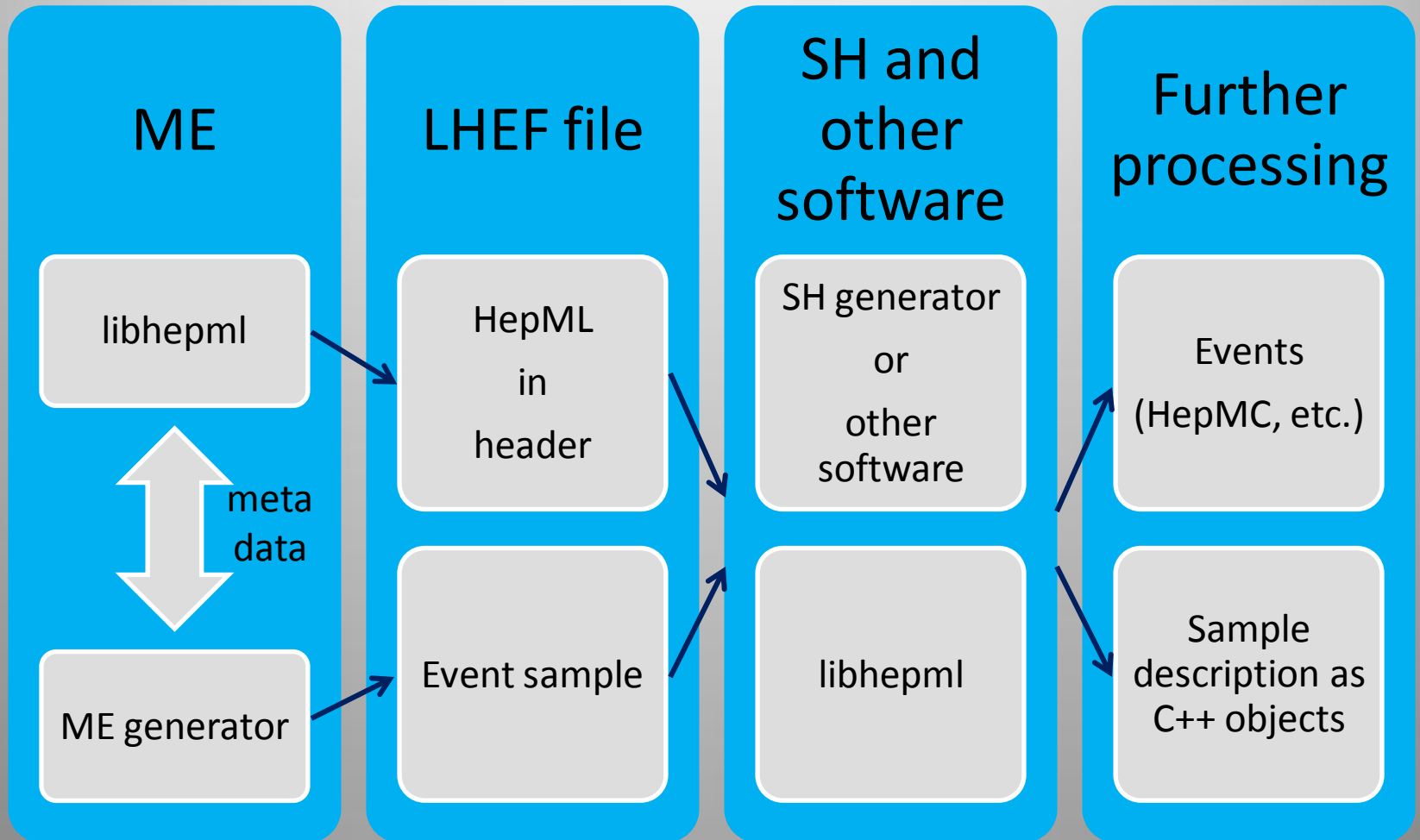
- Library to read/write/modify HepML descriptions of MC event files
- Simple and reliable way to deal with ME generator metadata
- Intended to automatize sample description in the MC simulation chain

# HepML markup language

- Unified XML format of MC event files metadata
  - to store all possible information from MC generators in XML format
  - to store generator input parameters and setup
  - an effort to fix a unified extensible way of MC events description
  - is an allowed part of LHEF standard event file header
- Main purposes
  - to unify MC event files description
  - to facilitate information transfer from Matrix Element generators to Shower generators
  - usage in MC generators tuning and testing
- Contributors
  - CEDAR <http://www.cedar.ac.uk>
  - LCG MCDB <http://mcdb.cern.ch>
- Homepage <https://twiki.cern.ch/twiki/bin/view/Main/HepML>

Work is supported by RFBR (grant 07-07-00365-a)

# libhepml in MC simulation



# HepML in LHEF

- J. Alwall et al., A standard format for Les Houches Event Files (2006) [hep-ph/0609017]
- structure:

```
<LesHouchesEvents version="1.0">  
  <header>  
    <hepml>  
      <!-- HepML sample description is here -->  
    </hepml>  
    .....  
  </header>  
  <init> ... </init>  
  <event> ... </event>  
  <event> ... </event>  
  .....  
</LesHouchesEvents>
```

# Features of libhepml

- Creating standard HepML headers from C++, C and Fortran programs in a very simple way
- Reading and parsing HepML
  - result is C++ objects
  - obtained data could be easily converted to other formats (HTML, plain text, ROOT trees, etc.)
- Mixing HepML headers of event files
- Modification and saving HepML headers

# Current state of developments

- Full support of HepML schema release 0.1
  - read/write/parse support
- Partial support for new HepML schema release 0.2
  - header writing support
  - HepML parsing (in progress)
  - mixing (in progress)
- Release for public use is LibHepML 0.2



# HepML Header Writing example - C (I)

```
void form_sample_article(void * doc)
{
    lhaef_general_description * sample;
    sample = calloc(1, sizeof(lhaef_general_description));
    strcpy(sample->title, "p,p->Wbbj->l,nu,b,b,j process from CompHEP");
    strcpy(sample->abstract, " There are about 1.1 MEvents of CompHEP p,p->Wbbj with leptonic decays of W. Spin correlations and contribution from the initial sea quarks were taken into account.");
    sample->experiment = calloc(1, sizeof(lhapdf_experiment) );
    strcpy(sample->experiment->experiment, "CMS");
    strcpy(sample->experiment->group, "RDMS");

    set_general_description(doc, sample);
}
```

# HepML Header Writing example - C (II)

```
lhaef_model * model;
  model = calloc(1, sizeof(lhaef_model) );
  strcpy(model->name, "SM, Feynman gauge");
  set_model(doc, model);
  ...
  lhaef_parameter * param;
  lhaef_math_notation * notation;
  param = calloc(1, sizeof(lhaef_parameter) );
  notation = calloc(1, sizeof(lhaef_math_notation) );
  param->notation = notation;

  strcpy(param->name, "Ms");
  strcpy(param->value, "0.117");
  strcpy(notation->plain, "Ms");
  strcpy(notation->html, "m<sub>s</sub>");
  add_model_parameter(doc, param);
```

.....

....

# HepML Header Writing example – C (III)

```
int main()
{
    void * doc;
    char * hepml_doc;

    doc = init_lhaef_document();
    form_sample_article( doc );
    form_hepml_document( doc, &hepml_doc, 0, 1 );

    printf( "%s\n", hepml_doc );

    release_hepml_document( hepml_doc );
    release_lhaef_document( doc );

    return 0;
}
```

# HepML Header Writing example – C++ (I)

```
#include "hepml.hpp"  
...  
mcdb::Article article;  
article.title() = "article title";  
article.abstract() = "abstract is here";  
....  
article.generator().name() = "gen name";  
article.generator().version() = "1.2.3";  
....  
mcdb::Subprocess sub;  
sub.crossSection().unit = mcdb::CrossSection::mub;  
article.process().subprocesses().push_back( sub );  
article.process().subprocesses().push_back( mcdb::Subprocess() );  
....
```

# HepML Header Writing example– C++ (II)

```
mcdb::Model::Parameter mp;  
mp.name() = "s12";  
mp.value() = "s12 value";  
article.model().parameters().push_back( mp );  
....  
mcdb::Author author1, author2;  
au1.firstName() = "Name1";  
au1.lastName() = "Surname1";  
...  
a.authors().push_back( author1 );  
a.authors().push_back( author2 )  
.....  
cout << hepml::articleToXml( article);
```

# libhepml in real projects

- Packages
  - CompHep 4.5 (Header of LHEF output file)
  - MCDB (sample description in Interfaces)
  - CMSSW (Sample description in MCDB Interfaces)
- Libraries
  - libmcdb: LCG AA external library  
*/afs/cern.ch/sw/lcg/external/mcdb*

# Conclusion

- libhepml is an easy way to store and get metadata from standard HepML descriptions
- First public release (0.2) of libhepml is ready
  - HepML header create functions are supported
  - Mixing and parsing will be ported to the new public version very soon
- libhepml is already in use in several projects
- We propose libhepml for the standard way to keep ME generator metadata in LHEF
- Download is available here:  
<http://mcdb.cern.ch/distribution>