

HepML: LCG Conception

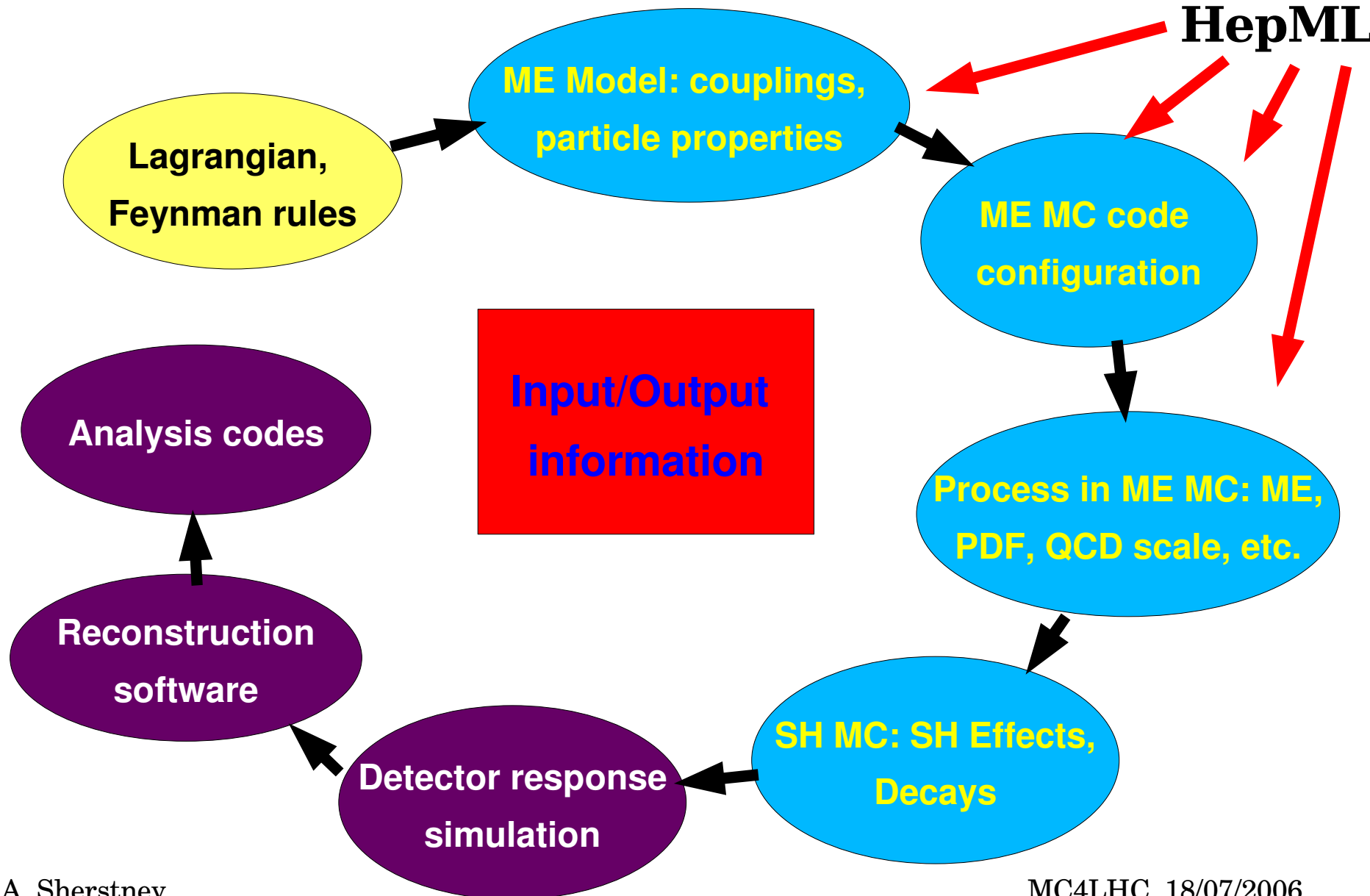
A. Sherstnev

on behalf of the HepML team

- Problem description
- General conception of HepML
- HepML Schema
- HepML software
- HepML: CEDAR and LCG
- Questions for discussion

HepML team:

Sergey Belov (JINR),
Lev Dudko (MSU),
Anton Gusev (IHEP)
A. Sh. (U. of Cambridge)



Unification of data file formats is useful on all stages of the full simulation. **The problem: information is very heterogeneous.**

Main players in the area: LHA-I, SUSY LHA, PDF LHA.

- They standardize data transferred!
- No matter: HOW the data should be stored...

Benefits of the unified formats:

- Common manipulation software (logging, histograms, etc.)
- Comparison of different codes
- Simpler mechanism to store and reuse samples

The main message of HepML: presentation of information is also important factor in complicated software projects...

- XML is a native tool to describe structures of data (format)
- XML is a simple language (syntax, core ideas)
- XML document is a ASCII text
- Possibility to keep versions (important for development)
- XML provides flexibility: no pre-defined tags and elements. User should invent tags (a dictionary of tags).
- Lots of developed/standard software (parsers, support of DOM, etc.)
- W3C standard, supported by IBM, Sun, Microsoft, Linux community.
- XML Schema technology: formal description of connections between elements

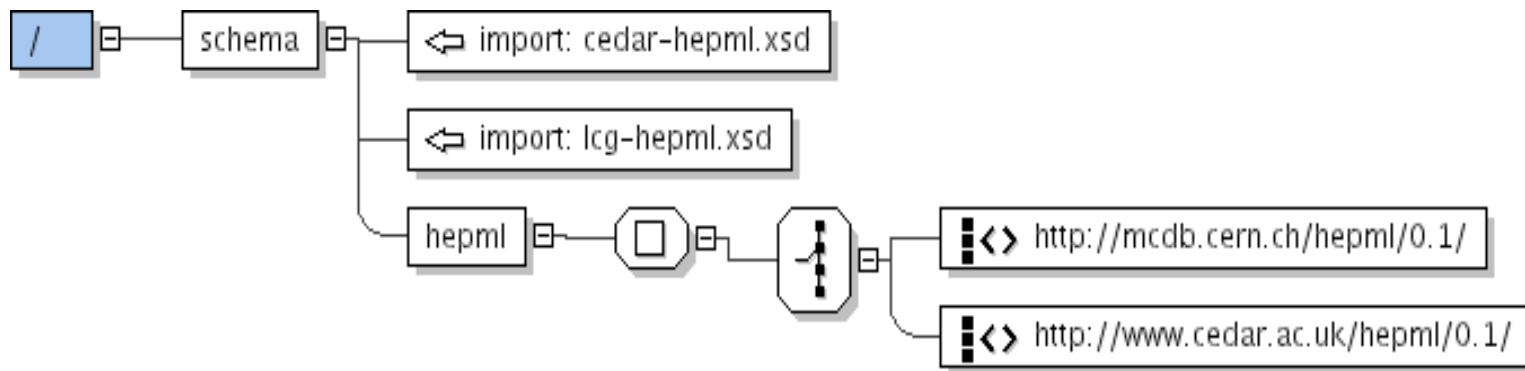
On the 1st stage we limit ourselves with ME->SH data transferring.

- A general HepML schema (see later): hepml.xsd
- Namespaces for independent groups: it gives a possibility to develop particular schemas separately
- Several types of HepML documents (see later).
- Tag dictionary: a set of tags, defined in a schema
- The first dictionaries: LHA-I schema [lha1.xsd] and MCDB schema [mcdb.xsd]: masses and widths, couplings, etc.
- Two API (see later):
 - a general parser and parsers for dictionaries (e.g. for LHA-I)
 - a library to store data to files in the HepML format

Possible HepML documents

- **Event sample meta-information:**
 - Meta-information/information: a bit artificial division. Information described a given sample as the whole.
 - Can be generated in addition to original event files
 - Can contain a description of event record (self-documented)
- **MC generator description:** store the whole info about a current version of a code (close to CEDAR generator description).
- **MC model description:**
 - A native solution for complicated model (e.g. in CompHEP)
 - Can contain num. parameters and symb. expressions (MathML)
 - Simple validation procedures possible

- The common XML Schema [hepml.xsd](#): guarantee a general framework for all groups participated in the project.
- The scheme content: the most general tags, references to common particular schemas, and definitions of namespaces for each group in the project.
- Each group will be allocated a unique namespace (e.g. cedarns, lcgns) to provide a possibility to develop particular schemas and HepML docs independently.
- If a group changes its schema version [hepml.xsd](#) should change version too.
- A developer can assign a default namespace for a given HepML document, but (s)he can use tags from other namespaces.

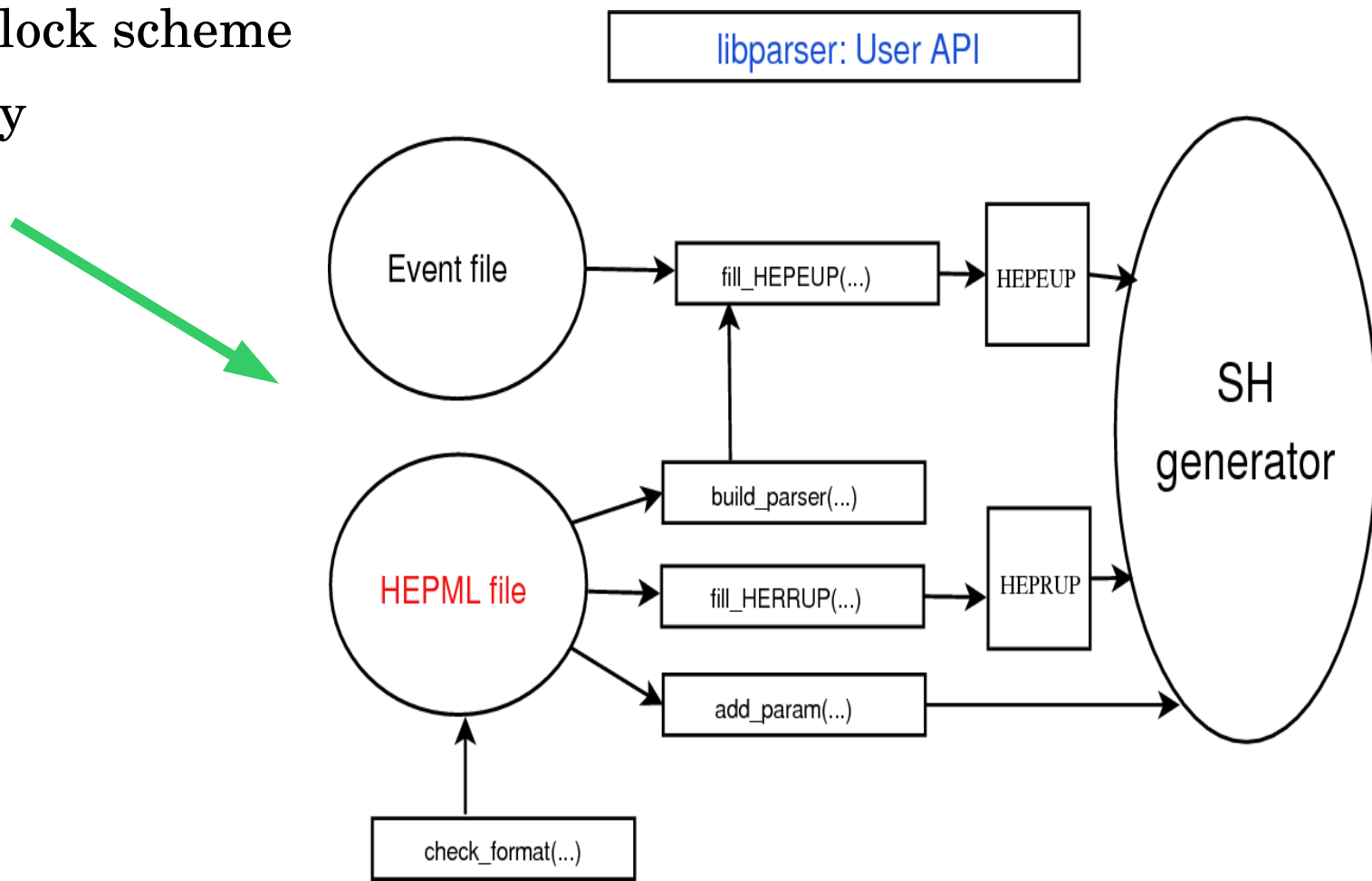


Possible problems

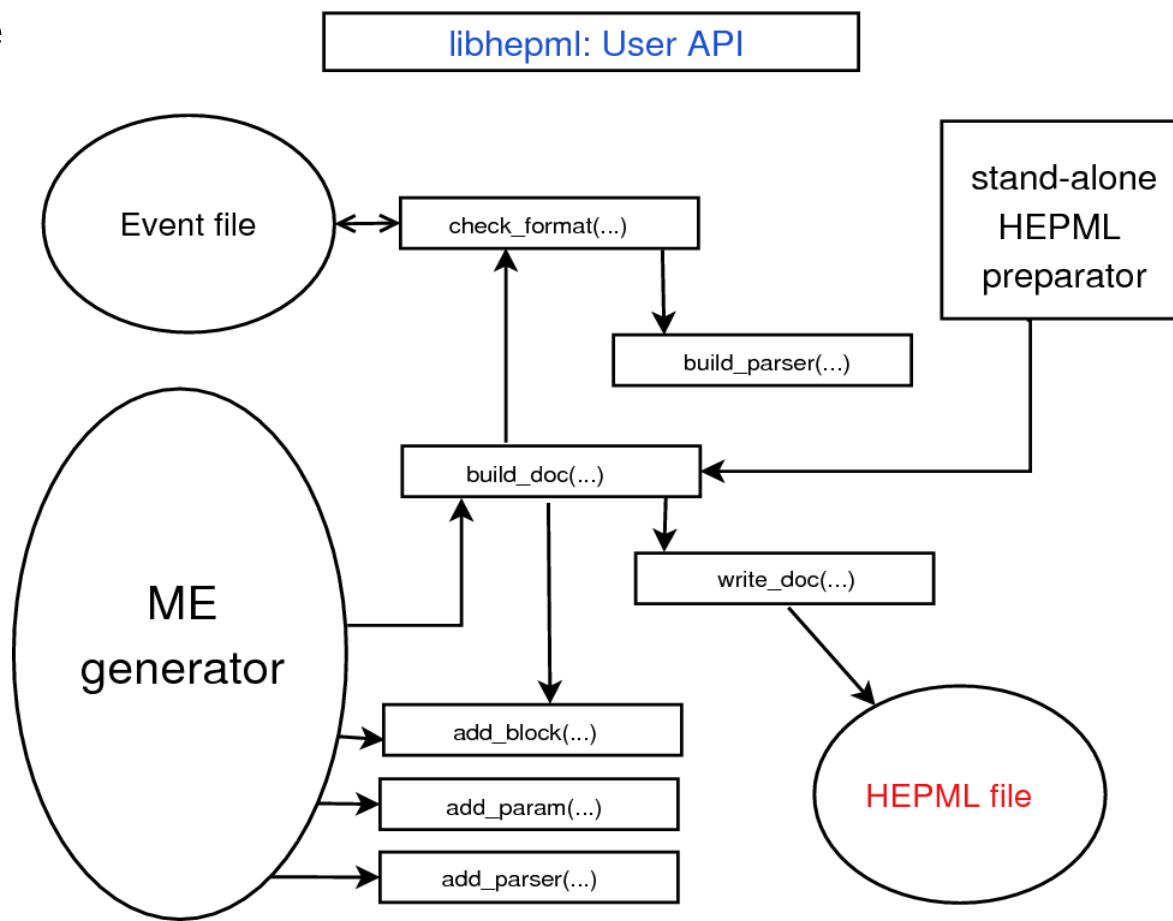
- Fast version changing: if we will change the general schema ver.
 - at the end the version will stabilize
 - periodical change (a-la GENSER)
- How to resolve tags in different namespaces with the same sense
 - :) - permanent discussions between groups, clearing of schemas
 - :(- define attributes which define all synonyms in other namespaces

- Besides HepML documents and schemas we need a special software to use the schemas, tag dictionaries, HepML documents.
- Should be provided to C/C++ and FORTRAN environments:
 - Our choice: Xerces or libxml2 (see comparison: <https://twiki.cern.ch/twiki/bin/view/Main/XMLparser>)
 - FORTRAN: wrappers to Xerces-C
- First stage: 2 API and simple standalone package
 - 1st API: libhepml – routines to store data in HepML format, maintenance of the LHA-I dictionary
 - 2nd API: libparser – general validation and parser routines +

- **Tasks:** form a HepML document, based on information taken from memory or form an event file. The library should know about the event file format!
- A possible block scheme of the library



- **Tasks:** parse a HepML document, form a format record for events (if they are not saved in the HepML format), fill LHA-I structures, transfer auxiliary parameters to a SH code
- A possible block scheme of the library



- Documents
 - General Schema: ready
 - Schema for MCDB: ready
 - LHA-I tag dictionary and schema: ready
 - MC generator description HepML documents: in progress
(could be built on CEDAR schema)
- Software:
 - Parser API: in progress
 - Writer API: in progress

Our projects are not competitors!

Different goals and application area:

- **CEDAR**: XML format for log-files, input configuration of MC generators, and HepData experimental data records
- **LCG**: store records on MC event files to MCDB; unified interface ME->SH (based on LHA-I); XML format for event data meta-information; MC model description

Common base:

- Configuration parameters for MC generators (MC models)
- Standard description of physical processes in HepML

It is very native to establish a close collaboration

1. The project is a part of the LCG MC Services group
2. Current team:
Sergey Belov, Lev Dudko, Anton Gusev, A. Sh.
3. Homepage (Wiki):
<https://twiki.cern.ch/twiki/bin/view/Main/HepML>
4. All groups interested in the project are welcome to collaborate!

1. Can CEDAR adopt the proposed general HepML schema?
2. Who will develop HepML dictionaries? HepML collaboration? MC code authors?
3. Do we need a common storage for all HepML documents?
4. Do we need an interface to HepMC?