LaTeX-ed on June 19, 2003.

The HepMC C++ Event Record for High Energy Physics

Matt Dobbs
Lawrence Berkeley Lab

June 20, 2003

`http://cern.ch/HepMC/`

Outline

- Introduction
- Description of the Event Record
- Package Status – Maintenance Requirements
- Conclusions

Motivation

FORTRAN standard → HEPEVT

- simple, fast, sequential, fixed size & content

Event Number Sequential list of particles • Status, PDG ID, Mothers(2) Daughters(2) • Momentum 4-vector, Vertex Position 4-vector

user responsible for internal consistency etc...

modifying an event (often) means re-creating a new event

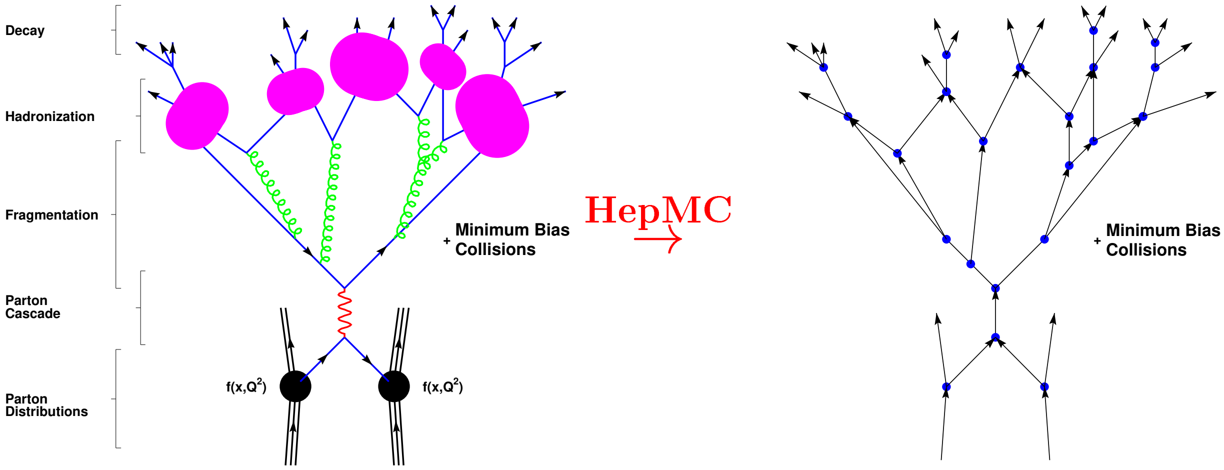
Requirements for an OO Event Record

- act as a generic interface b/t event generators and detector simulation
- support **modularization of event generators**
- store event in a physical manner - higher abstraction
- **simple** & **intuitive** for the end user
- **generic** - not specific to experiment or generator

→ *OO technology is well suited for these goals.*

HepMC Features

- store information in a **graph structure**, physically similar to a collision event



graph-like records will give more knowledge/abstraction than sequential lists

- unlimited number of Particles / Vertices
- easy to use and access information: powerful iterators
  → can iterate over particles or vertices
  specifying range (immediate parents, or all ancestors) and selection criteria (predicates)

refer to `http://cern.ch/HepMC/` for iterator description and to `http://cern.ch/HepMC/examples.html` for a set of complete examples.

HepMC Features

- modular event generation
  contains necessary info to be used as a messenger between two modular components of event generation

$$
\begin{array}{lcl}
\overbrace{\text{SubProcessGenerator}}^{\text{Specific NLO Generator}} & \overset{\text{EventRecord}}{\longrightarrow\longrightarrow} & \overbrace{\text{CascadePackage}}^{\text{ARIADNE}} \\
 & \overset{\text{EventRecord}}{\longrightarrow\longrightarrow} & \overbrace{\text{HadronizationPackage}}^{\text{Herwig}} \\
 & \overset{\text{EventRecord}}{\longrightarrow\longrightarrow} & \overbrace{\text{DecayPackage}}^{\text{Bpack+Pythia}} \\
 & \overset{\text{EventRecord}}{\longrightarrow\longrightarrow} & \overbrace{\text{DetectorSimulation}}^{\text{ATLfast}}
\end{array}
$$

Herwig++ and Pythia7 modularize their components in similar steps

The Fortran Les Houches Accord HepUP *[hep-ph/0109068 Generic User Process Interface for Event Generators, M. Dobbs and Les Houches collaborators]* (which is largely motivated by the HepMC goals/information content) has been hugely successful as an event record *interface*. HepMC is compatible with the same information content, allowing the extension of this modularity to OO.

General Package Structure

namespace

HepMC:: $\underbrace{\textbf{8 classes}}$ + $\underbrace{\textbf{few utility classes}}$

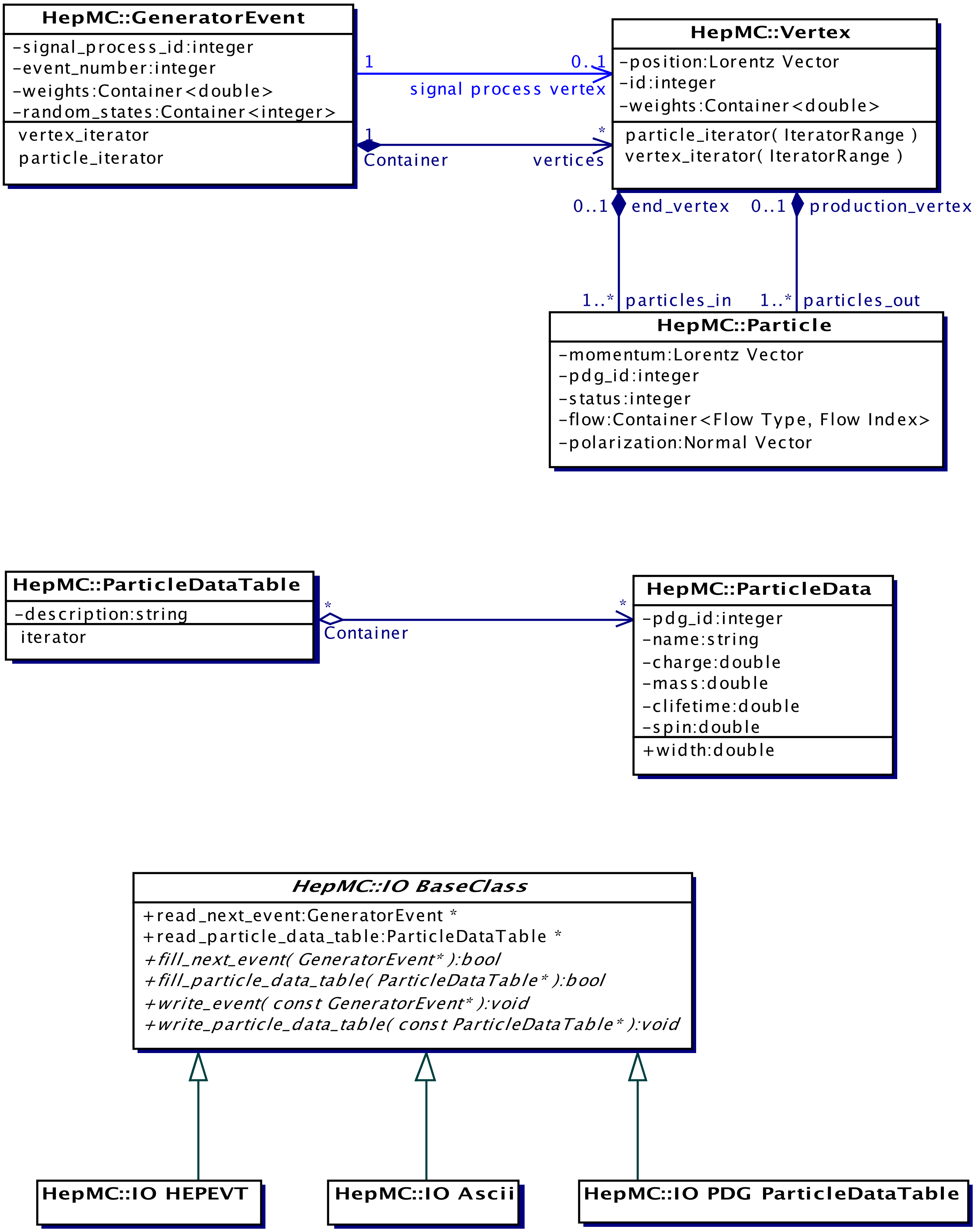
8 classes	few utility classes
GenEvent	
GenVertex	HEPEVT_Wrapper
GenParticle	IO_HEPEVT
Flow	IO_Ascii
Polarization	IO_PDG_ParticleDataTable
ParticleData	IO_Herwig
ParticleDataTable	PythiaWrapper
IO_BaseClass	HerwigWrapper

- $\approx$ 5000 **lines of code**
- development on **Linux with gcc 2.95/3.2**
  tested on HP-UX, Solaris
  ported to Windows Visual C++ by Witold Pokorski/Pere Mato (LHCb)
- syntax/container structure in **STL style**
- **dependencies**: **STL & CLHEP** vector classes

Note: ParticleData & ParticleDataTable are modular from the Event record
- provided only for completeness
- intended only as a lightweight tool
  probably something more substantial is necessary → HepPDT (?)
- interface is via integer particle ID
- allows any particle data table to be used/ attached

IO Strategies can easily be extended to other formats
OO database, ROOT, HBOOK

General Package Structure



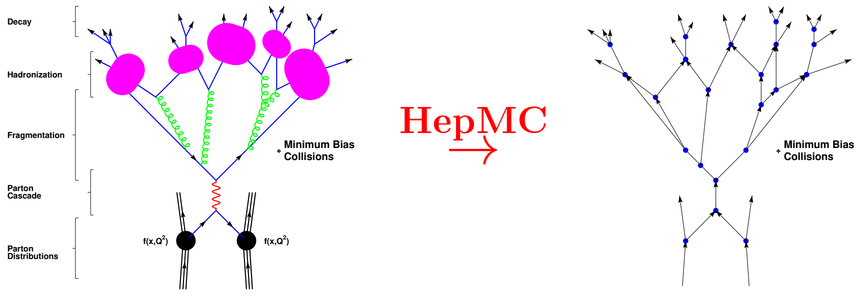
Package Status & Maintenance

- Event record is **Mature & Stable**
  used successfully in ATLAS for 2 years to interpret generically the output from Pythia, Herwig, Isajet, and even Hijing.
  - Persistified (ROOT) within ATLAS Athena structure.
- Maintenance was handed over to FNAL in 2001, who released it as part of CLHEP in 2002.
  so currently there are two versions of HepMC out there:
  (1) my version of hepmc.cern.ch (also referred to as the ATLAS version)
  (2) the CLHEP version
  - unfortunately that release (1.8.0.0) didn't compile (missing header file)
  - no new release that I am aware of.
  - no translation (F77 Generator to HepMC) routines in that release, so difficult for anyone to use it in practise.
  - so far most users of HepMC (ATLAS, LHCb, ..) get their code directly from me or the HepMC webpage.
- at least one other group (Cambridge) has expressed interest in maintaining it.
- Dobbs is not interested in maintaining it, but has been supporting it since conception.
- Where's the challenge in Maintenance??
  The core event record is easy to maintain (essentially I've touched very little in 3 years). The translation/IO routines which interpret the F77 output from Pythia/Herwig/Isajet are very tricky to get right. Requires an expert who understands each generator. Many bugs have been discovered–some were bugs inside the generators, some weren't–all require significant attention.

  Note: for the C++ generators, this maintenance problem goes away, since the Monte Carlo authors plan to supply the translation routines.
  But we will want access to the F77 generators well into the lifetime of LHC.

Conclusions

- HepMC is a object oriented event record in C++
- it is widely used within ATLAS, and is becoming the OO standard for LHC experiments.
- physics driven design with simple and intuitive usage higher abstraction - stores event in a physical manner



- supports a number of translation strategies – those strategies REQUIRE maintenance beyond what they are getting now.
  An efficient maintenance solution, with quick response to bug reports in the translation routines needs to be achieved.
- Product is the fruit of many useful suggestions/comments and open development with HEP community.

Documentation

- M.Dobbs, J.B. Hansen, Comp. Phys. Comm. 00 (2000) 1-6.
- user manual available at `http://cern.ch/HepMC/`
- online code browser documentation with examples