A Standard Event Class

L. Garren

Fermi National Accelerator Laboratory USA

http://www-pat.fnal.gov/stdhep/c++

June14, 2001

ROOT Users Workshop

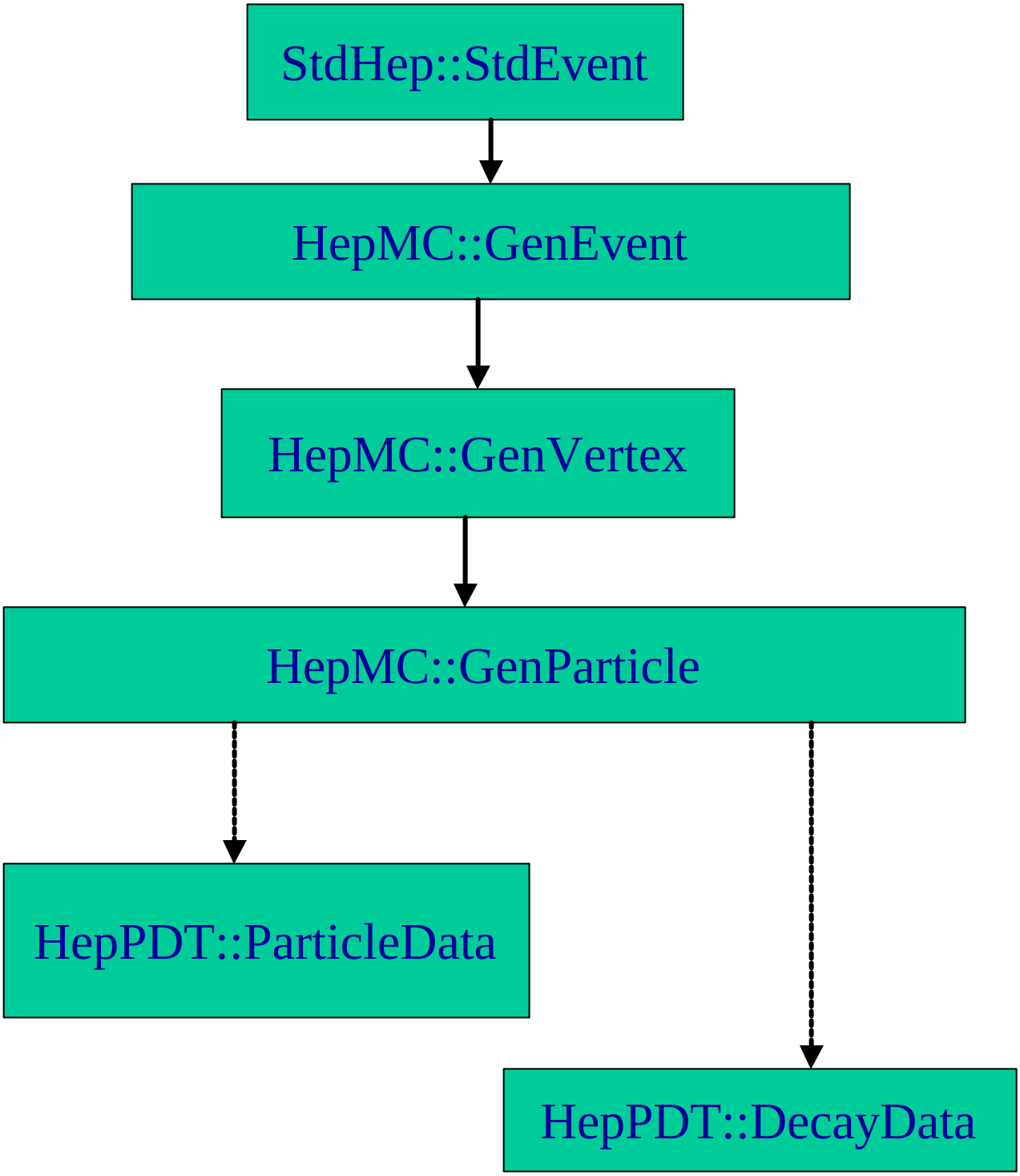
StdHepC++: the first pass

- StdHep::Run knows about Events
  - does not contain Events
  - run identifier
  - begin/end run information
- StdHep::Event is a set of Collisions
  - to allow for beam pileup
- StdHep::Collision is a set of Particles
- StdHep::Particle
  - looks a lot like the HEPEVT common block
- Translation methods
- StdHep::ParticleData (orphan class)

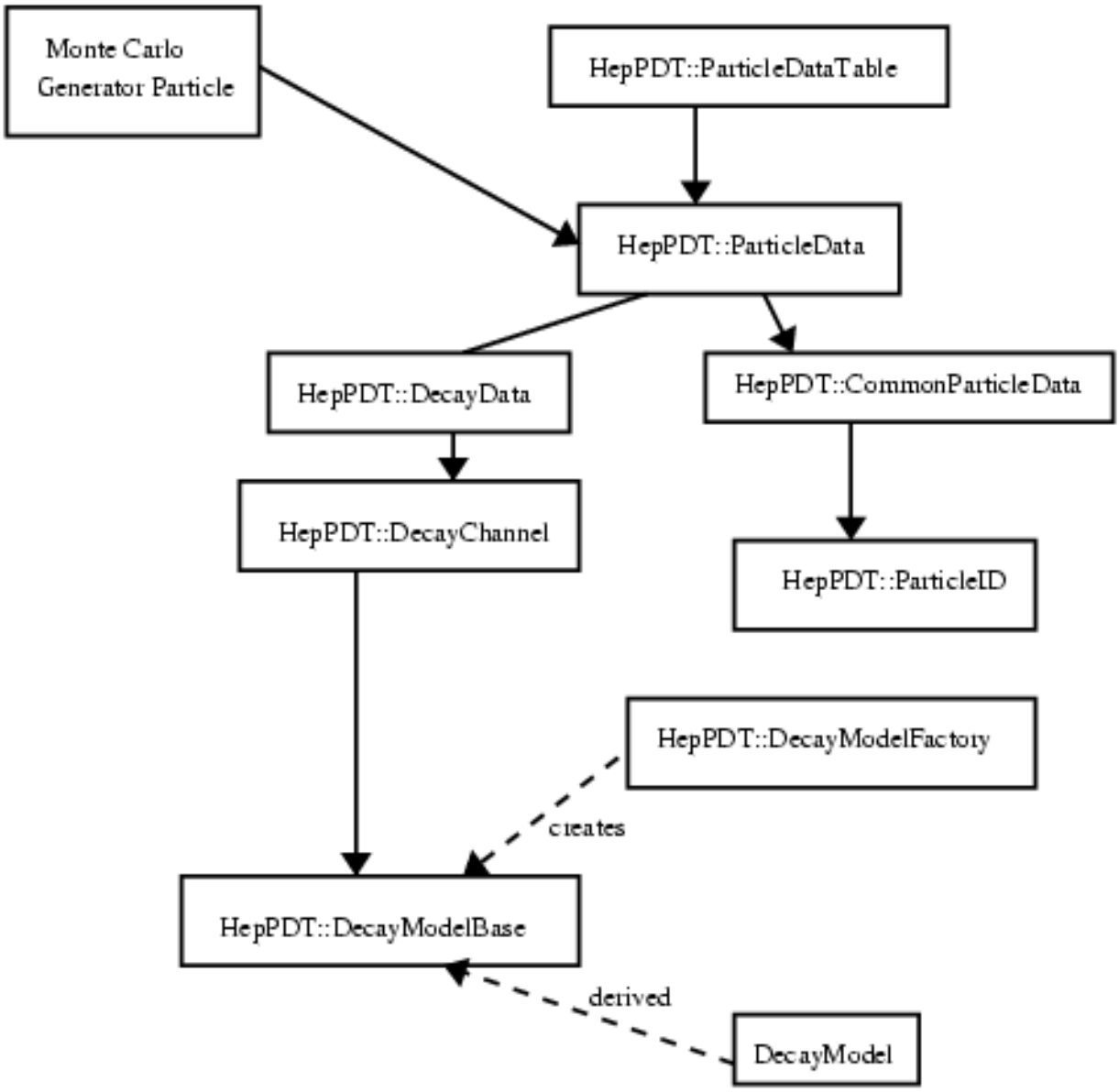
StdHepC++: what's new

- Major development to utilize power of C++
- HepPDT
  - particle data table
  - contains information from Particle Data Handbook
- a generated event is a Directed Acyclic Graph (DAG) of vertices connected by particles
  - a parent may have any number of children
  - a child may have more than 1 parent
  - each vertex is a node on the graph
  - use HepMC classes

StdHepC++


StdHep::StdEvent
HepMC::GenEvent
HepMC::GenVertex
HepMC::GenParticle
HepPDT::ParticleData
HepPDT::DecayData

HepPDT


Monte Carlo
Generator Particle
HepPDT::ParticleDataTable
HepPDT::ParticleData
HepPDT::DecayData
HepPDT::CommonParticleData
HepPDT::DecayChannel
HepPDT::ParticleID
HepPDT::DecayModelFactory
creates
HepPDT::DecayModelBase
derived
DecayModel

HepPDT::ParticleData

- All the information from PDG
  - HepPDT::ParticleID
  - std::string & name
  - double charge
  - double color
  - HepPDT::SpinState (all of them)
  - HepPDT::Measurement mass
  - HepPDT::Measurement width
  - double lowerCutoff
  - double upperCutoff
  - boolean methods (hasCharm)
  - decay data

HepPDT::DecayData

- std::vector< DecayChannel >
- decay(int)
- decay(double random)
- size()
- appendMode(DecayChannel)
- write(std::ostream)

HepMC::GenParticle

- HepLorentzVector momentum
- HepPDT::ParticleID
- int status
- Flow (color, charge, …)
- Polarization (theta, phi)
- production GenVertex*
- decay GenVertex*
- print(std::ostream)
- double generatedMass
- ParticleData*
- (DecayData*)

(HepMC::GenParticle continued)

- All the old HEPEVT information
  - mother
  - secondmother
  - index to the first daughter
  - index to the last daughter
  - maintain compatibility
- collision number
- <Particle*> parents()
- <Particle*> ancestors()
- <Particle*> children()
- <Particle*> desendants()

HepMC::GenVertex

- HepLorentzVector position
- std::set<GenParticle*> incoming particles
- std::set<GenParticle*> outgoing particles
- WeightContainer
  - vector of doubles
- iterators
- methods to access ancestors and desendants

HepMC::GenEvent

- single beam interaction
- signal process ID
- int event number
- GenVertex* signal process vertex
- std::set<GenVertex*>
- WeightContainer
- std::vector<double> random states
- iterators
- collision number
- number of particles, vertices
- parents, children, etc. methods

StdHep::StdEvent

- Each beam crossing may have several interactions. Therefore a StdEvent is a collection of GenEvents.
  - event number
  - std::vector<GenEvent*>
- translate HEPEVT to C++
- <Particle*> parents(Particle*)
- <Particle*> ancestors(Particle*)
- <Particle*> children(Particle*)
- <Particle*> descendants(Particle*)
- number of particles, vertices, collisions
- accessors to return particles and vertices
- iterators

more utility methods

- methods to convert directly from generator output
- isParent
- isDaughter
- <Particle*> quarkList(quark ID)
  - get a list of particles containing the specified quark
- <Particle*> stableDescendants
- <Particle*> chargedStableDescendants

Conclusion

- There is a strong need for C++ standard Monte Carlo generator interface.
- StdHepC++/HepMC is a natural object-oriented implementation of such an interface which could be used in ROOT
- HepPDT incorporates all static particle data information and provides a natural interface to decay methods.