BES III common analysis tools

— CLHEP, Histogramming and BParticle

Ji Xiaobin

March 1, 2006

Contents

- CLHEP Vector Package
- Histogram and Ntuple
- BParticle

CLHEP Vector Package

The CLHEP Vector package implements 3-vectors, 4-vectors, rotations, Lorentz transformations and related concepts.

Avaliable Classes

- Hep3Vector – Vector of real quantities in 3-space
- SpaceVector – Derived from Hep3Vector
- UnitVector
- HepLorentzVector – Vector of real quantities in 4-space
- LorentzVector – Typedefed from HepLorentzVector
- Hep2Vector
- HepRotation
- Rotation – Derived from HepRotation
- HepLorentzRotation
- LorentzTransformation – Derived from HepLorentzRotation

Hep3Vector

Hep3Vector v1(v1x, v1y, v1z), v2(v2x,v2y,v2z)

- dot product: HepDouble v12 = v1.dot(v2);
- cross product: Hep3Vector v12c = v1.cross(v2);
- magnitude: HepDouble Magv1 = v1.mag();
- angle: HepDouble Angv12 = v1.angle(v2);
- perpendicular component: HepDouble perpv = v1.perp();
- polar angle: HepDouble pang = v1.polarAngle(v2);
- azimuth angle: HepDouble aang = v1.azimAngle(v2);

HepLorentzVector

HepLorentzVector p1(p1x,p1y,p1z,e1), p2(p2x,p2y,p2z,e2)

- mass: HepDouble p1m = p1.m();
- invariant mass: HepDouble p12m = (p1+p2).m();
- Get 3-vector direction of the Lorentz vector:
  Hep3Vector p1v3 = p1.vect();
- boost: HepLorentzVector p2star = p2.boost(-p1.boostVector());

Histograms & N-tuples

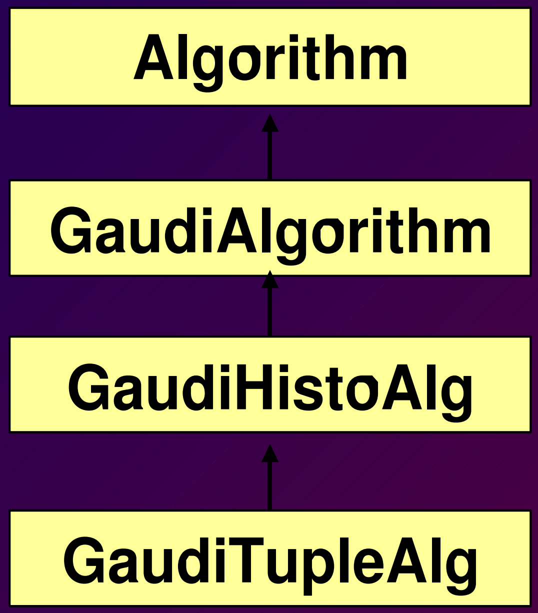
- One of the key tools in HEP
- HBOOK was one of the best packages in CERNLIB
- Usage and function is obvious
- In Gaudi it's the same concept
  - First book then fill, requires explicit use of histogram pointer (c.f. HFF1)
  - Simplification in GaudiHistoAlg, combine in a single call, and hide pointer handling in base class

Histograms - Good To Know...

- Histograms are kept in memory
- If not saved - they are lost
- Like all other data - they reside in a Data Store
  - Same access mechanism
- Persistency is configurable
  - HBOOK, ROOT

GaudiHistoAlg and GaudiTupleAlg

- Specialisations of GaudiAlgorithm



Algorithm

GaudiAlgorithm

GaudiHistoAlg

GaudiTupleAlg

- Simplify handling of histograms and N-tuples

Book and Fill Histograms

```
#include "AIDA/IHistogram1D.h"
...
// Book 1D histogram in the histogram data store
IHistogram1D* m_hTrackCount= histoSvc()->
      book( "simple", 1, "TrackCount", 100, 0., 3000. );
SmartDataPtr<MCParticleVector> particles( eventSvc(), "/Event/MCParticles" );
if ( 0 != particles ) { // Filling the track count histogram
  m_hTrackCount->fill(particles->size(), 1.);}
```

Histogram Persistency(ROOT)

CMT requirements file

```
use RootHistCnv v*
```

Job options file

```
ApplicationMgr.DLLs += {“RootHistCnv” };
ApplicationMgr.HistogramPersistency = “ROOT”;
RootHistSvc.OutputFile = ”histo.root”;
```

N-tuples - Good To Know...

- Cannot be kept in memory

  - Grow and grow and grow...
- Like all other data - reside in a Data Store
  - Same access mechanism
  - Usage simplified by GaudiTupleAlg

Book and fill an N-tuple

- Defining N-tuple tags

```
NTuple::Item<long>                m_ntrk;
NTuple::Item<float>               m_energy;
// Items for the column wise n-tuple
NTuple::Array<long>               m_iNumbers;
NTuple::Array<float>              m_fNumbers;
NTuple::Item<long>                m_n;
```

- Booking and Declaring Tags to the N-tuple

```
NTuplePtr nt1(ntupleSvc(), "MyTuples/1");
if ( nt1 ) m_tuple1 = nt1;
else {
  m_tuple1 = ntupleSvc()->book ("MyTuples/1", CLID_RowWiseTuple, "example");
  if ( m_tuple1 )    {
    status = m_tuple1->addItem ("Ntrack",   m_ntrk);
    status = m_tuple1->addItem ("Energy",   m_energy);
    status = m_tuple1->addItem ("N", m_n, 0, 100);
    status = m_tuple1->addIndexItem ("FNumbers", m_n, m_fNumbers);
    status = m_tuple1->addIndexItem ("INumbers", m_n, m_iNumbers);
  }
  else    {   // did not manage to book the N tuple....
    log << MSG::ERROR << "    Cannot book N-tuple:" << long(m_tuple1) << endmsg;
    return StatusCode::FAILURE;
  }
```

Book and fill an N-tuple(cont.)

```
  static int n = 0;
  m_ntrk   = long(sin(double(n)) * 52. + 50.);
  m_energy = sin(double(n)) * 52. + 50.;
  m_n      = abs((1234567*(n+1))%100);
  for( int i = 0; i < m_n; i++ ) {
    m_fNumbers[i] = cos(double(2*n)) * 52. + 50.;
    m_iNumbers[i] = long(cos(double(2*n)) * 52. + 50.);
  }
  n++;
  status = m_tuple1->write();
  if( status.isFailure() ){
    log << MSG::ERROR << "    Cannot fill N-tuple:" << long(m_tuple1) << endmsg;
    return StatusCode::FAILURE;
}
```

It is up to the user to ensure the arrays do not overflow

N-tuple Persistency

Job options

```
NTupleSvc.Output = {”MyTuples DATAFILE=’ntuple.root’ OPT=’NEW’ TYP=’ROOT’”}
```

B(es)Partilce Project

- Write Particle (Charged tracks, neutrals, Vertice and composed particle) information into TDS. Analysis Algorithm can access these info.
- May develop to a common standard platform, make Analysis job easily and friendly
- Allow Physics groups to design Physics Data model easily

Contents

BParticle Collection Event Analysis Event Data Model

BParticleID Collection Particle identification data

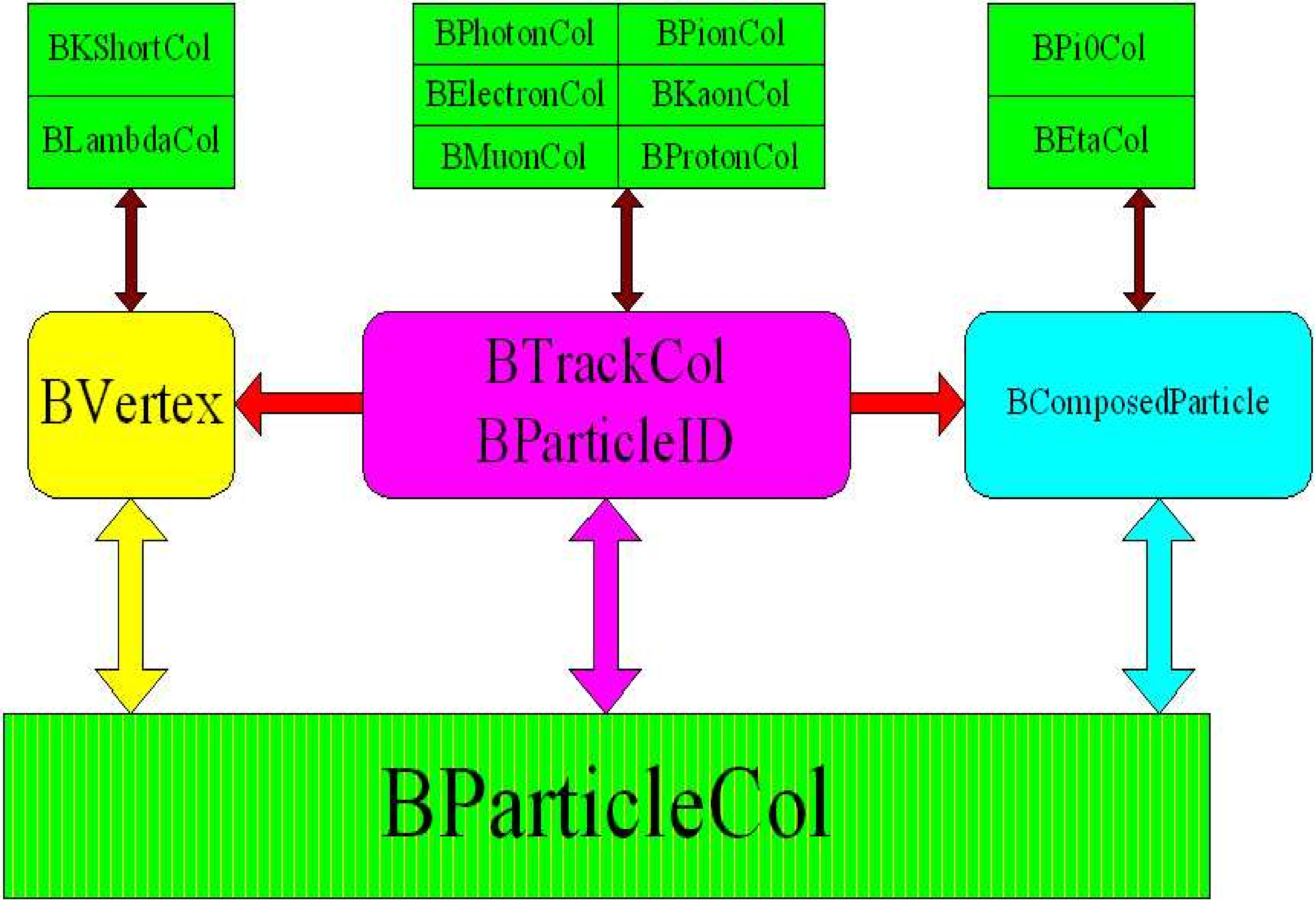
BTrack Collection Charged and Neutral Track infomation data

BVertex Collection (2nd)Vertex Reconstruction data

BStableParticle Collection Photon, Electron, Muon, Pion, Kaon, Proton

BComposedParticle Collection π^0, η, K_S, Λ etc.

BParticle


BKShortCol
BLambdaCol
BPhotonCol
BPionCol
BElectronCol
BKaonCol
BMuonCol
BProtonCol
BPi0Col
BEtaCol
BVertex
BTrackCol
BParticleID
BComposedParticle
BParticleCol

How to use BParticle(in BOSS 5.0.0)

1. cmt co Event/EventModel
2. cmt co Analysis/VertexFit
3. cmt co Analysis/BParticle
4. cmt co Analysis/Physics/ParticleSelectionAlg

BParticle

```
// track List ID
int m_trackID;
// PDG ID
int m_particleID;
// reference point(0, 0, 0)
HepPoint3D m_refpoint;
// WTrackparameters(charge, px, py, pz, e, x, y, z)
WTrackParameter m_wtrk;
SmartRef<DstTrkList>  m_dstTrk; // Dst TrackList
SmartRef<BParticleID>  m_pid; // ParticleID info
SmartRef<BVertex>   m_vertex; // Vertex info
SmartRef<BComposeParticle> m_bcompart; // ComposeParticle
SmartRef<BParticle> m_mother; // mother
SmartRefVector<BParticle>  m_daughters; //decay daughters
```

BParticleID

```
int m_trackID;  // TrackList ID
int m_type; // 1: electron  2: muon  3: pi/K/p
int m_ndof;
double m_chiDedx[5];
double m_chiTof1[5];
double m_chiTof2[5];
double m_chiTofE[5];
double m_chiTofQ[5];
double m_chiEmc[5];
double m_prob[5];
double m_chisq[5];
```

BVertex

```
int m_vertexID;
VertexType m_vertexType;
VertexParameter m_vpar;
double m_chisq;
int m_ndof;
WTrackParameter m_wtrk;
double m_lxyz;
double m_lxyz_err;
SmartRefVector<BParticle> m_outgo;
```

BParticle Frame work

- Tracklist duplication
- VeeVtxReconstruction
  - KShortReconstruction
  - LambdaReconstruction
- GoodTrackSelection
  - GoodPhotonSelection
  - GoodElectronSelection
  - GoodMuonSelection
  - GoodHadronSelection
- BuildPi0List
- BuildCharmMesonList
- Analysis Cut are set by job option file

BParticle in Analysis

```
#include "BParticle/BParticle.h"
#include "BParticle/BParticleID.h"
#include "BParticle/BVertex.h"
#include "BParticle/StableParticle.h"
#include "BParticle/BComposedParticle.h"
// get pion Lists
SmartDataPtr<BPionCol> pionCol(eventSvc(), EventModel::Analysis::BPionCol);
// get Kaon Lists
SmartDataPtr<BKaonCol> kaonCol(eventSvc(), EventModel::Analysis::BKaonCol);
// get Photon Lists
SmartDataPtr<BPhotonCol> photonCol(eventSvc(), EventModel::Analysis::BPhotonCol);
//get KShort Lists
SmartDataPtr<BKShortCol> ksCol(eventSvc(),EventModel::Analysis::BKShortCol);
//get pi0 Lists
SmartDataPtr<BPi0Col> pi0Col(eventSvc(), EventModel::Analysis::BPi0Col);
```

DO not forget to add following line in your job option file

```
#include "$PARTICLESELECTIONALGROOT/share/jobOPtions_ParticleSelection.txt"
```

BTrack in Analysis

```
#include "BParticle/BParticle.h"
#include "BParticle/BParticleID.h"
#include "BParticle/BVertex.h"
#include "BParticle/StableParticle.h"
#include "BParticle/BComposedParticle.h"
SmartDataPtr<BTrackCol> trkCol(eventSvc(), EventModel::Analysis::BTrackCol);
BTrackCol::iterator itTrk = trkCol->begin();
for(; itTrk < trkCol->end(); itTrk++) {
   // if((*itTrk)->isCharged()) continue;
    if(!(*itTrk)->isGoodTrack()) continue;
    if(!(*itTrk)->isPhoton()) continue;
    if((*itTrk)->isDaughter()) continue;
    BParticle* photon = (*itTrk)->photonTrack();
    double energy = (photon->p()).e();
    // fill ntuple
}
```

More Information

- CLHEP manunal
- Gaudi manunal
- Example: Analysis/Physics/BParticleTestAlg

Thank You!