# Comments on Olivier's physics event model

I think it would be nicer if all reconstructed particles, simple and composite, are of the same base type, eg. ReconstructedParticle.
Why must there be 2 types, ProtoPart and Particle for Brunel and DaVinci?
Surely Brunel will produce some composite particles, eg. $V^0$, $\pi^0$.

Similarly, I think all vertices, including primary, should be of the same base class, ReconstructeVertex.

In Olivier's design the identity of a ReconstructedParticle is given by its concrete class type.
I think the identity of a ReconstructedParticle should be given by a data member of the class and not by the concrete class type.
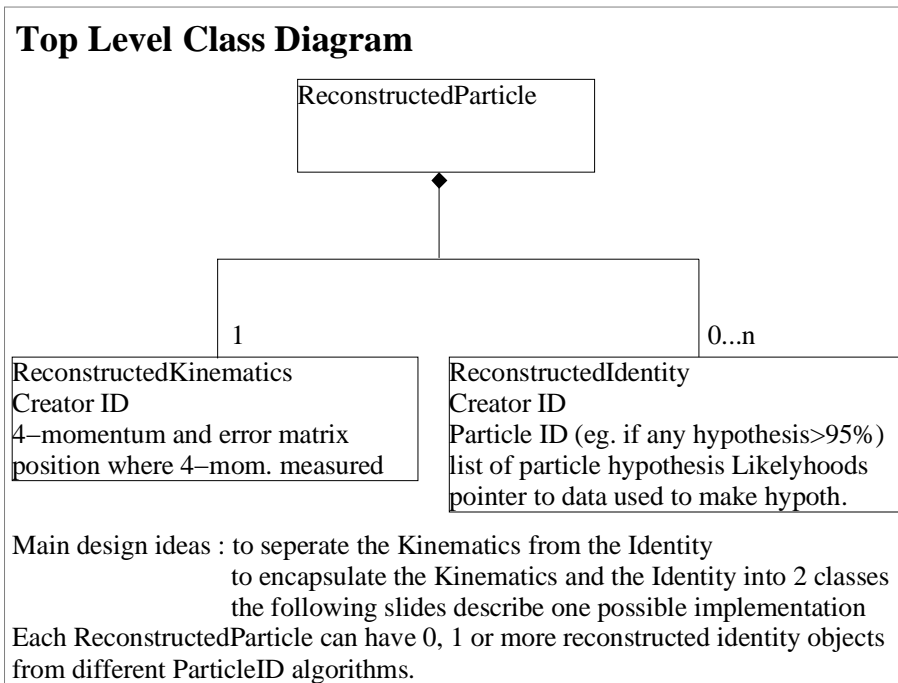The identity of a ReconstructedParticle may not be defined, and when defined it should be easy to change, eg. after application of an alternative ParticleID algorithm.

I think the kinematics and the identity of a ReconstructedParticle should be independent, and implemented as two contained objects, eg of abstract type ReconstructedKinematics and ReconstructedIdentity.

## Naming

The MC produces MCParticle and MCVertex objects.

Therefore, I think it would be nice if the Reconstruction (Brunel and DaVinci) produced ReconstructedParticle and ReconstructedVertex objects.

**Top Level Class Diagram**

```
                    ┌─────────────────────────┐
                    │ ReconstructedParticle   │
                    │                         │
                    │                         │
                    └───────────┬─────────────┘
                                ◆
                    ┌───────────┴─────────────────┐
             1                              0...n
┌──────────────────────────────┐  ┌──────────────────────────────────────┐
│ ReconstructedKinematics      │  │ ReconstructedIdentity                │
│ Creator ID                   │  │ Creator ID                           │
│ 4−momentum and error matrix  │  │ Particle ID (eg. if any hypothesis>95%)│
│ position where 4−mom. measured│  │ list of particle hypothesis Likelyhoods│
└──────────────────────────────┘  │ pointer to data used to make hypoth. │
                                   └──────────────────────────────────────┘
```

Main design ideas : to seperate the Kinematics from the Identity
                to encapsulate the Kinematics and the Identity into 2 classes
                the following slides describe one possible implementation
Each ReconstructedParticle can have 0, 1 or more reconstructed identity objects
from different ParticleID algorithms.

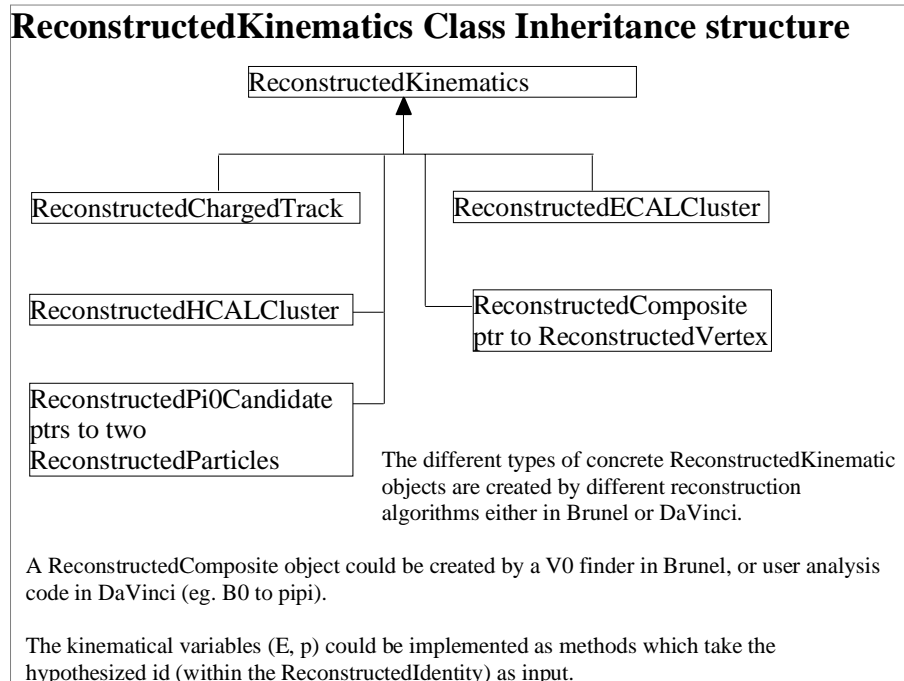A ReconstructedParticle is completely defined by its kinematics and its identity.

The kinematics and identity are orthogonal/independent and complete.

Kinematics are determined by VELO, Tracking and Calorimetry.

Identity is determined by RICH, Calorimetry and Muon.

The ReconstructedKinematics are created when the ReconstructedParticle is created.

In general, the ReconstructedIdentity is created afterwards by particleID algorithms.
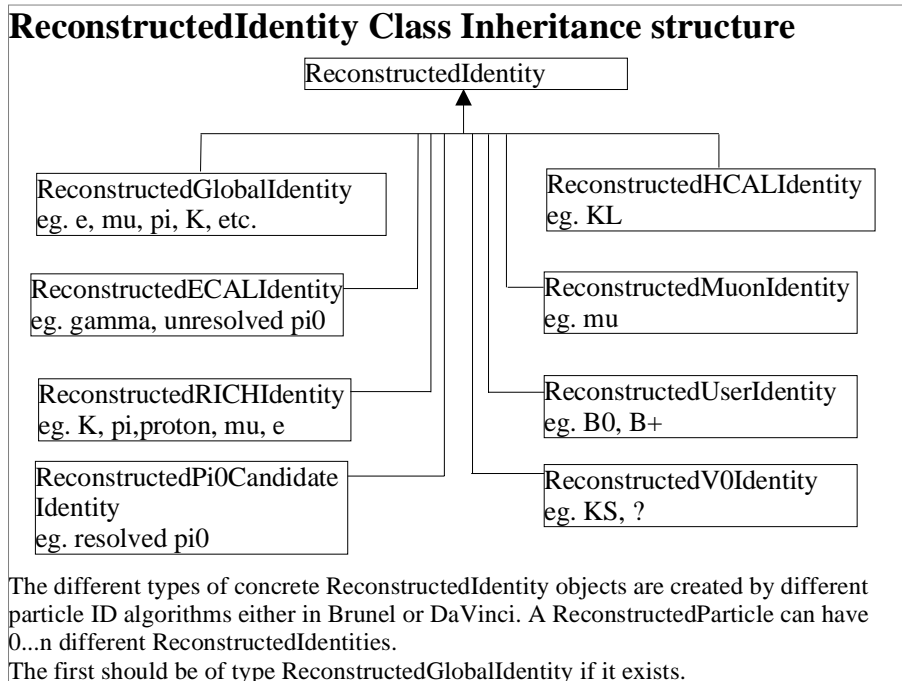
**ReconstructedKinematics Class Inheritance structure**

```
                    ReconstructedKinematics
                              ▲
         ┌────────────────────┴───────────┐
  ReconstructedChargedTrack      ReconstructedECALCluster


  ReconstructedHCALCluster ─┐      ReconstructedComposite
                            │      ptr to ReconstructedVertex
                            │
  ReconstructedPi0Candidate─┘
  ptrs to two
  ReconstructedParticles
```

The different types of concrete ReconstructedKinematic
objects are created by different reconstruction
algorithms either in Brunel or DaVinci.

A ReconstructedComposite object could be created by a V0 finder in Brunel, or user analysis
code in DaVinci (eg. B0 to pipi).

The kinematical variables (E, p) could be implemented as methods which take the
hypothesized id (within the ReconstructedIdentity) as input.

Energy and 3–momentum can be implemented as
methods of the ReconstructedKinematics class
which takes the hypothesized identity
(from ReconstructedIdentity) as input from
the containing ReconstructedParticle.
eg. E is a function of p and pid for charged tracks.
        P is a function of E and pid for neutrals.
The Kinematics are defined by the fitted track
parameters or calorimeter cluster info, or by the decay
ReconstructedVertex if a ReconstructedComposite.
The mother–daughter relationship is independent of
the particle ids. It is defined purely by the decay
kinematics.

**ReconstructedIdentity Class Inheritance structure**

ReconstructedIdentity

ReconstructedGlobalIdentity
eg. e, mu, pi, K, etc.

ReconstructedECALIdentity
eg. gamma, unresolved pi0

ReconstructedRICHIdentity
eg. K, pi,proton, mu, e

ReconstructedPi0Candidate
Identity
eg. resolved pi0

ReconstructedHCALIdentity
eg. KL

ReconstructedMuonIdentity
eg. mu

ReconstructedUserIdentity
eg. B0, B+

ReconstructedV0Identity
eg. KS, ?

The different types of concrete ReconstructedIdentity objects are created by different particle ID algorithms either in Brunel or DaVinci. A ReconstructedParticle can have 0...n different ReconstructedIdentities.
The first should be of type ReconstructedGlobalIdentity if it exists.

The ReconstructedIdentity is created by a ParticleID algorithm when given an unidentified RecoParticle.
Exceptions : eg. pi0 ReconstructedParticle objects are created by a Brunel or DaVinci pi0 finder. The pi0 finder creates both the ReconstructedKinematics and and the ReconstructedIdentity based on default cuts. The hypothesis should be easy to change in case one wants to tighten pi0 cuts for example. For this reason the ReconstructedIdentity object of a pi0 candidate should have concrete type ReconstructedPi0**Candidate**Identity. Its actual identity, pi0 or not, is given by the value of the ParticleID data member.
A ReconstructedParticle can contain more than 1 ReconstructedIdentity object, corresponding to the results of different ParticleID algorithms.
For a composite particle, the ReconstructedIdentity could be used to store a hypotheses, eg B0.

**eg. Object Diagram for B to pi+ KL** (only concrete instantiated objects)

ReconstructedParticle

ReconstructedUserIdentity
particleID = B0

ReconstructedComposite

ReconstructedVertex
NumberDaughters=2

Pointer

Pointer

ReconstructedParticle

ReconstructedParticle

ReconstructedChargedTrack

ReconstructedHCALCluster

ReconstructedHCALIdentity
particleID=KL

ReconstructedGlobalIdentity
particleID=pi0

ReconstructedRICHIdentity
particleID=pi0