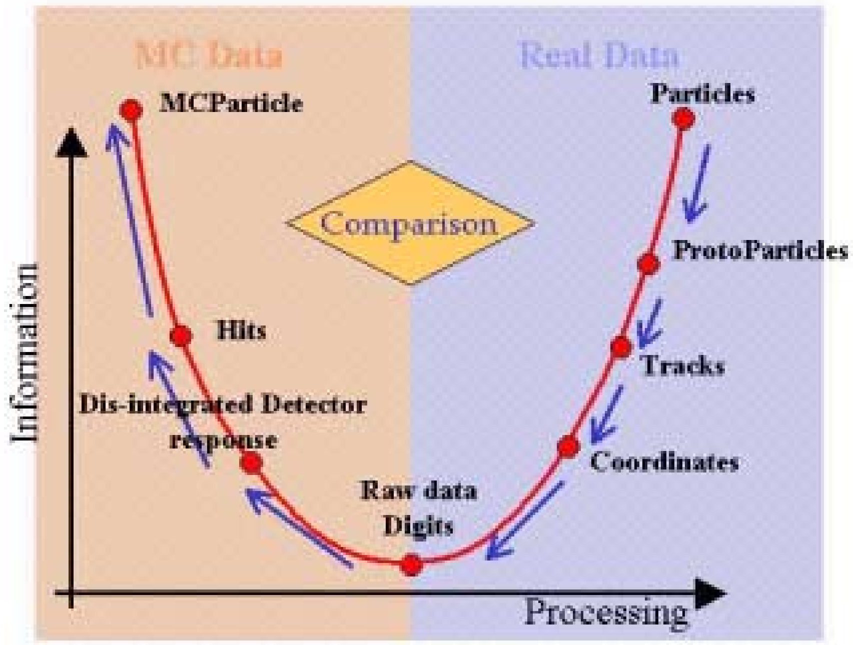
**Discussion on Physics Event Model - November 6, 2001**

Present:
O.Callot, M.Cattaneo, Ph.Charpentier, G.Corti, M.Frank, E.deOliveira, O.Schneider

- It would be necessary, as for any other contained data object to have a unique label. Association between objects would be done by mapping pointers to this label and maps should be checked for validity.

- Sorting of objects should be done only on local copies, but via this unique label it would be possible to map to the original container.

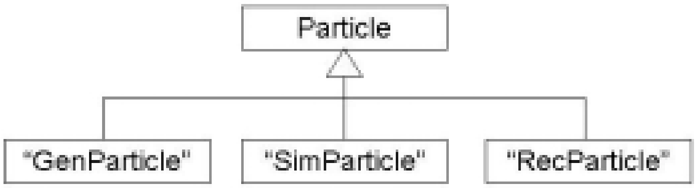
- It is felt necessary to have a **clear split between MC & Reco**, full navigation back to the original MC counterpart is available by stepping through the links or by accessing short cuts tables where previously performed navigations are stored. Tools (associators) should provide the connection as pointers in both cases.



Association between Particles and MCParticle is possible following link from previous stages + algorithmic operation or short cuts stored in a "table". The navigation should be done via an associator (the same one with the two options?)

- ➡ We would like to **run the same decay reconstruction algorithms** on generator and simulation classes as well as on physics analysis classes this implies that we should have a Particle Base class from which the generator particles, the simulation particles and the reconstruction particles derive. Tools should then be able to work on the common part. Only when creating objects it is necessary to know of which concrete type they are.

This implies that we could not use HepMC as it is, since it is very specific to the generator world. A converter would have to be provided and user should only do their analysis via the LHCb particle class.



- ➨ **Particles** on which to run the physics reconstruction/analysis **are different objects from ProtoParticles**. ProtoParticles have all the links to reconstruction informations: tracking, calorimeters, individual and global particle ID results. NO particle ID choice is done in Protoparticles: all valid hypothesis are listed. Particles instead have a CHOSEN particleID and hence a mass: in case of composite particles this is the measured mass. Particles should NOT know about detectors directly: if such info is necessary it should be accessed via the ProtoParticle.
  **Particles and ProtoParticles are also different classes**.

- Protoparticles are the end products of the reconstructions and as such cannot be modified by a physics algorithm. The first thing a physics algorithms will do is (via tools) to request list of particles from the ProtoParticles according to some particleID criteria. Particles objects will have a reference to the ProtoParticle they have been created from. Tools to create standard list of particles should be provided.

- Particles can be modified as long as they are private copies to the algorithms or passed to the tools but when they are put in the transient store they can no longer be modified.

- Kinematic fitting that will modify particle parameters will always work on copies of the objects (it should anyway be possible to compare the parameter of a particle before and after the fit).

- Particles will be part of decay trees. It should be possible to store whole tree or partial trees: B/D/$K_S$ trees. Particles could be part of different trees. In fact the tree represent the relationship between particles (and vertices). In general Gaudi containers “own” the contained objects. There will be a flat container that analysis will start with (for ex. List of pions) and then a decay tree “container” will be built. This is a problem with ownership of container when wanting to save the tree.

- Should the relationship between particles and vertices be in the Particle/Vertex classes (as in Babar) or in the container (as in Delphi) ? The idea of having it separate from the class (many tree possible with the same particle) is attractive, on the other end the other way conforms more to the present containers structure. Further investigation in necessary. How is the tree/graph structure implemented in Delphi? How could it fit in the Gaudi Data Model?

**Main points for next discussion:**

- **Relationship/history**
  a. in the class or in the container
  b. is the tree a container or a navigator
  c. which if the container that owns new particles when in transient store
- **Particle and Vertex Base Class design**
  Keep it simple: it is easier to add than to remove
  a. Methods: fourMomentum, threeMomentum (at which point?), error matrix?
  b. Reference to ProtoParticle
  c. Which methods are data?