Towards a New Track Class

Rob Kutschke, Hans Wenzel

Fermilab

June 15, 2007

Motivation

- Persistable class that reports the complete status of what we know from the vertexer+tracker and provides input to:
  - Vertex fitting.
  - Matching to clusters in calorimeter.
  - Muon ID.
  - Constrained fitting: beamline, mass constraints … .
  - Physics analysis.
  - Quality control and selections in general.
- What it is not:
  - This is not a class that can refit tracks; just the output of the fit.
  - Not sufficient to be the internal representation in a track finder/fitter.
  - Does not include info from ECAL, muon system.
  - Neutrals (K0s, Lambda, converted gamma) are objects that are produced by the constrained fitting system, not instances of this class.
    - They have links back to their daughters, info about fit quality ….

Method Overview

```
Class CompleteTrack {
   int getCreatorID();
   // … more information valid everywhere on this track for all mass hypotheses.

   boolean   hasMassHypothesis( MassHypothesis m);              // ?Is this needed?
   GlobalTrack getGlobalTrack( MassHypothesis m );              // Track information valid for one mass hypothesis
}

Class GlobalTrack{
   int          getChisquared();
   int          getNDF();
   // … more information valid everywhere on the track for  this mass hypothesis.

   LocalTrack getLocalTrack( Location loc );              // Track information valid locally for this mass hypothesis.
}

Class LocalTrack {
  TrackParameters  getTrackParameters();
  Symmetric Matrix  getErrorMatrix();
  // … more information valid only locally.
}
```

Example Locations: PCAO, PCA beam line, innermost hit, outermost hit, impact on ECAL (any hit, any surface?)
MassHypothesis: e, mu, pi, K, p, (others?)

For Details: http://home.fnal.gov/~kutschke/Track_doc/

Open Questions?

- Should these be interfaces, not classes?
- How do we link to MC Truth?
  - Method: List<MCParticle> getMCParticles();
  - Cross-reference objects in the event?
- What about 2D tracks? Should they be instances of this class?
- How do we link to hits, residuals, individual chi2 contributions?
  - Method: List<Cluster> getClusters();
  - Cross-reference objects in the event?
- How do we describe muons, electrons …
  - Cross-reference objects in the event?
  - Derived class with additional attributes?
- How should the class behave if the requested location and mass hypothesis are not available?
  - Best effort? Throw? User configurable?

Next Steps

- Is the overall structure OK?
- Add details to the classes.
  - We are looking at all ~20 classes in org.lcsim that implement the Track interface and will decide which functions from these we should include in these classes.
  - Open to suggestions for additional methods.
  - Work in progress: http://home.fnal.gov/~kutschke/Track_doc/
- Will speak with tracking and vertexing people outside of SiD and see what they have to say.
- When there is broad agreement, propose a new persistent class to LCIO (LCIO 2?).
  - Tony/Norman: any idea of the timescale of LCIO 2?

Backup Slides

Use Cases for Mass Hypotheses

- We would like to be highly efficient for finding converted photons and exclude them from our leptons in jets sample that is used for b and c tagging. Do we need proper energy loss and scattering corrections for the electron and positron to find their vertices optimally?
- We will be interested in finding Lambda to p pi decays so that we do not confuse flavor tagging. Do we need to do proper energy loss for the proton or are all Lambdas fast enough that this is not an issue?
- In the case that we have a fully reconstructed D from a B, is it important to fit the Kaon correctly in order to get optimal flavor tagging?
- Can the slow pion from D* decay be used for c tagging?
  - Probably not an issue since pion would be the default.