What is in my contribution area

Nick Sinev, University of Oregon

Event

- Event/util/SmearTrackerHits
  - Driver to combine SimTrackerHit s in event into clusters if they have same MC parent and in the same layer. There is setings of maximum cluster span on the surface and in the depth. This is to exclude combining multiple track crossing of the layer in one cluster. Coordinates of the center of gravity of such cluster are randomly smeared with specified sigmas and used to generate SmearedTrackerHit s put back into event.
- Event/base/SmearedTrackerHit
  - Extends BaseTrackerHitMC by adding method to smear hit by specified displacements in RPhi, R and Z (all linear)

Ztracking

- Here I had modifications of Mike’s Ronan TrackingCheater. It is called TrackingCheaterSmHits and uses event’s list of SmearedTrackerHit, not SimTrackerHit s as in Ronan’s original code. It generates CheatedBaseTrack s and put them in the event.
- CheatedBaseTrack extends BaseTrack by adding reference to MCParticle which generated all the hits.
- Many of original Mike’s classes also are modified here to provide fully qualified Track interface (Mike had dummy implementations for some methods). Also I am using LinkedHashMap instead of HashMap in original code, to provide repeatability of Track list in the event in repeated analysis runs.

Tracking/util

- CMTransform is the utility to transform covariance matrix between Cartesian and cylindrical or disk surface representations
- ThreePntTrkPar returns track parameters for track traced through 3 space points, given as arguments to getParams access function. It uses ThreePntCircle class also provided in this package.
- WMTrackPropogator is one of the major part of track fitter. It traces Track through all elements of the Detector inside tracking volume, determines multiple scatter angles for each layer and generates Error matrix for all tracker active layers, including correlated terms. It also provides coordinates of the track crossing points for each layer and teack momentum at each such point. It takes into account air in the detector and energy loss along track as track passes through materials.

Tracking/wmfitter

- WeightMatrix calculates weight matrix for hit position measurements for all affected layers based on the information from WMTrackPropagator and assigns hits (only one in each layer) to be used by fitter. It uses hit list provided by Track interface, and assigns hits closest to track and consistent with expected errors in position (though pretty liberally, allowing 10 sigmas deviation). Weight matrix calculated by this code includes only elements for layers with hits. Dimension of the matrix is 2xNumberOfLayersWithHits, as each layer provides 2 independent measurement.

Tracking/wmfitter

- SLDFitTrack – track fitter using weight matrix algorithm (similar to what was used in SLD tracking). It solves problem of fitting track to measured points by solving set of the minimialization equations for chi2. This is set of 5 equations with 5 unknowns (number of track parameters), so solving of this equations is not time consuming. But the conversion of error matrix into weight matrix (by inversion), which is performed by WeightMatrix class may be time consuming, if detector has many layers.

Additional utilities of SLDFitTrack

- As the fitting of the track provides track parameter errors covariance matrix, this matrix can be used to study tracking resolution for different detector configuration. This is exactly the same utility as Bruce's covariance matrix calculation code. It can be considered as crosscheck. But it is backed by the fact, that one can check that fitted parameters residuals are consistent with covariance matrix. And I did check it for this fitter, results are pretty consistent, except for curvature residuals in endcap tracker – fitted residuals has about 15% narrower distributions than predicted by covariance matrix.
- To utilize covariance matrix calculations of the fitter, one does not even need to have simulated events. Fitter has the function "SimulateHits" which allows it to generate hits according to expected helix crossings of active layers for any detector given by compact.xml description. It does not generate displacement due to multiple scattering, so it is not the replacement of Geant4, but for calculation of covariance matrix hit positions are not needed, only track crossing positions. I generate this hits to let fitter perform it's fitting, because covariance matrix calculations are done in the fitting process.