Tracking Software Discussions at SLAC, May 7-11, 2007

Rob Kutschke,Fermilab

SiD Tracking Meeting

May 18, 2007

- Participants:
  - Norman Graf, Tony Johnson, Caroline Milstene, Jeremy McCormick, Tim Nelson, Dima Onoprienko, Rich Partridge, Nick Sinev, Hans Wenzel, Rob Kutschke
  - Matt Charles, Ron Cassel (ALCPG Tuesday meeting).
- Notes and transparencies on ILC Wiki:
  - https://confluence.slac.stanford.edu/display/ilc/Home
    - Reconstruction and Analysis
    - SiD Tracking Planning May7-11, 2007
  - Or direct url: https://confluence.slac.stanford.edu/display/ilc/SiD+Tracking+Planning+May+7+to+11%2C+2007
- My opinions are in red.

LCIO to Physicist Translation

- SimTrackerHit
  - The information in one G4 stepping point.
  - One or more per track/sensor intersection.
  - The information sent from G4 to org.lcsim
- RawTrackerHit
  - Intended to represent the information readout from the detector for a single hit strip/pixel.
  - And to retain MC truth info.
- TrackerHit
  - Intended to represent the input to the pattern recognition and fitting codes.
    - ie a cluster of strips/pixels.
  - Some important functionality missing or awkward.

Language (2)

- “Cylindrical Detector”
  - Refers to models in which layers are a single cylindrical sensor.
    - For example, sid00.
  - Distinct from polyhedral detector
- “Polyhedral Detector”
  - Layers made up of overlapping planar wafers.
  - AKA: “planar sensor detector” …

Persistency

- We will only persist the official LCIO objects
- We will work with LCIO to get small changes made to these objects, but not big changes. We will shoehorn our objects into the defined persistent objects.
- We will not support persistency for other information.
  - I think this is a mistake.
  - I agree that we must share common formats for things close to simulated physcis analyses: tracks, vertices …
  - The choice to persist something or not should be driven by our needs, not those of LCD, GLD and the 4th.
- A future evolution of the reconstructed track class (trajectory?), will allow us to persist residuals, track parameters at multiple locations etc.

Interface to Real Data

- We choose not to consider that someday we will use this framework to analyze data from a test beam or from a real experiment.
  - We chose not to define separate MCHit and DataHit classes and to design a dataflow that uses these in an appropriate way.
  - All hit and cluster classes all contain both “data” and “mc” methods.
- I don’t like this but we can live with it for a long time.

RawTrackerHit

- The class that represents a simulated single strip/pixel, as it might be read from the detector (+some MC truth).
- Agreed to make a few small changes – see wiki.
- Decided not to extend it to include:
  - The true, undigitized, pulse height deposited on a strip/pixel.
  - A breakdown of which track (or noise) contributed how much energy to the pulse height.
    - It is possible to indicate a pure noise hit but not a mostly noise hit.
- Persistency:
  - A flat container.
  - After readin, user can make structured containers from this.
  - I think we should persist structured containers that hold all elements hit on a single sensor, ready for cluster finding.

RawTrackerHit (2)

- Merging of events:
  - Option 1:
    - Merge events at SimTrackerHit level (g4 stepping point).
    - Recompute individual pulse heights.
    - Digitize
  - Option 2:
    - Merge events after digitization.
- Option 3 is explicitly excluded by the design:
  - Process “background” events to the point of undigitized pulse heights on strips/pixels.
  - Merge these in to signal events just before digitization.
- OK to exclude option 3 for now but it could become a problem if we want to do large runs of MC with a high fidelity model for simulating cluster shapes.

Clusters

- Everyone has their own cluster class.
  - Some extend TrackerHit others are completely independent.
- Need to make a public solution that works for everyone.
- Two options were presented:
  - Rob's and Rich's (stripped down version) were very similar.
  - Dima break the problem into two pieces:
    - ClusterData: A list of continguous hits.
      - Cannot return a meaningful position in any coord system.
    - TrackCluster:
      - Given (ClusterData, Track) it can return position information in either the local or global coord systems.
      - Can pass a null track as an argument.
- In all options, TrackerHits are the output of pattern recognition, not the input!!!
  - Why? Because you cannot get a final measurement until the cluster is associated with at track.

Clusters (2)

- Clusters will not be persisted in their native form but maybe we can shoehorn them into TrackerHit.
  - I prefer to persist in their native form.
- I like Dima's idea.
  - I have asked Dima to prepare a version of his code that I can use to feed cheated tracks to one of the existing fitters.
  - Should this be driven from RawTrackerHit or his own replacement class?

Track Fitters

- Many exist but none are ready for general use on both cylindrical and polyhedral detectors.
- Norman's TRF
  - Needs interface to new geometry.
- Nicks weight matrix
  - Cylindrical only.
- Caroline's Kalman filter used for muon tracking.
  - Needs to be adapted to be used for tracking.
- Colorado: KFitter
  - I don't know the status of this.
  - The main support person has left.

Fitted Track Class

- Need to report parameters + Cov matrix valid at many points along the track:
  - Innermost few hits.
    - Vertexing may indicate innermost hit is spurious.
  - Outermost hit.
  - Point of closest approach to beam line.
  - Impact point on calorimeter.
- Need access to residuals for hit rejection and diagnostics.
- Must allow option to drop information to satisfy disk space police.
- Norman: At SACLAY Ties Behnke proposed developing such a class for LCIO, he called it trajectory.
  - We should get our version of this quickly in order to get our requirements into LCIO.

TrackerHits

- We plan to persist TrackerHits.
  - Now tproduct of pattern recognition and track fitting, not input!
  - Have ceased being useful objects for anything except graphics.
- We should modify TrackerHit to point to clusters so that the full audit trail is available.
  - However we do not plan to persist the clusters so the full audit trail is only available within the job that created the tracks.
  - Therefore we cannot run clustering once and then run multiple jobs to do R&D on pattern recognition and fitting.
- Need to add a “type” field to TrackerHit.
- We plan to shoehorn clusters into TrackerHits? We plan to shoehorm TRF hits into TrackerHits.
  - Is this plan self consistent? . Probably once the “type” field exists.

Pattern Recognition Cheaters

- A variety of codes exist.
- So far they only work with cylindrical geometries, not polyhedral geometries.
  - At least one of them should be updated once the new geometry system is in place.
  - They updated ones should work with both geometries.

Configuration and Object MetaData

- We have no system for run time configuration of org.lcsim.
  - We need one.
- Should record the run time configuration information in the output file.
  - LCIO allows meta data associated with each object in the event. For much of the run time configuration information this would be a good solution: ie this collection of tracks was created by version xxx of some pattern recognition and version yyy of some fitter.

Old Style Geometries

- There are many existing LCIO files containing events simulated with the cylindrical detectors.
  - It will still be possible to read these files and instantiate the geometry representation.
  - Old code will continue to work.
  - What about pattern recognition and fitting codes developed for the planar detectors?
    - Fitters that we continue to develop work with both geometries.
    - Cheaters should work with both geometries.
    - I think that pattern recognition codes are likely to work with one style or the other and they should not be required to work with both. This is not a unanimous opinion.

Version Control

- Important in general, may be very important when dealing with codes for which old versions work on the old geometries but not the new ones.
- This is on the wish list, no more.

Original Goals

1. Define Classes for tracking infrastructure.
   - We have a plan with a few questions still open.
   - See Work Packages on Wiki:
     - https://confluence.slac.stanford.edu/display/ilc/Work+Packages
2. Define a cheater for SimTrackHit to whatever it is that is used as input for pattern recognition.
   - Rob will do this once the cluster classes are settled.
3. Re: Links back and forth between geometry and hits.
   - Important to distinguish event scope information from job and run scope information. So we should not hang hits on the geometry system.
4. Re: Forward Geometry. Can Hans present a roadmap?
   - See summary wiki page. Not presented due to time constraints.
5. How does the above affect the tracking algorithms.
   - Reduced scope of disagreement regarding code that works on both cylindrical and polyhedral geometries.
   - No real progress on version control.