RPROM

CERN 22 October 2001

CommonDet Use in Muon

Stefano Lacaprara,*INFN, Padova.*

OUTLINE:

- Detector description;
- Persistency;
- TrackReconstruction and Navigation;
- Mis-Alignment;
- Problem and future;

Most of these functionality was implemented by Anna and I just inherit: so I'm not a real expert on everything

Detector description

- CommonDet functionality and interface implemented for the 3 sub-detectors;
- DT: MuBarChamber *is a* DetUnit;
- CSC: either MuEndChamber and MuEndLayer *are* DetUnits;
- RPC: MRpcDetUnit (==MRpcChamber) *is a* DetUnit (Giacomo B.);
- Also DetLayer implemented in the 3 system, but for Navigation (see after).

Persistency

- SimHit persistency fully based on CommonDet PSimHit;
- Digi directly from CARF;
- Tracks: Not working at the moment!!

  In principle should work out of the box, but does not: some investigation by Vincenzo.

  High priority, but we are still working on improvement-debugging of tracking, persistency comes after.

Track reconstruction:

CommonDet schema used:

- SeedGenerator

  The L1 seed is a FTS, we use a trick to associate a RecHit to it.

- TrajectoryBuilding (now split in several classes, new)

  We start from the FTS taken from the seed

  The navigation is not *a la CommonDet*

  The trajectory updating is not trivial

- TrajectorySelector (trivial one)

- TrajectorySmoother (not used)

Navigation

- It's where Muon differs most from CommonDet implementation;
- Original implementation is "fake" CommonDet:
- MuBarDetLayer *has a* MuBarNavigableLayer which works only within the barrel system and can't be easily extended to include CSC and RPC.
- ME/MRpcDetLayer has none but the dummy one provided by the base class (so nextLayers() gives always null vector)
- Navigation inside MuonTrajectoryBuilder is "one shoot": given a FTS and a propagationDirection a class MuonNavigableLayer MuonNavigableLayer::nextLayers(FTS,dir) returns **all** DetLayers compatible with the FTS, not only the nearest ones.

- MuonNavigableLayer *is a* NavigableLayer, but does not belong to any DetLayer: It's used *a la Singleton* without being implemented as a Singleton, just instantiated once!
- Same CommonDet interface but different functionality: very dangerous!!
- Now we have a "CommonDet" navigation implemented via a MuonNavigationSchool (no RPC yet) not used for the Track finding, even if should be easy to plug in the new MuonTrajectoryBuilder. Studies needed to test performances!

Trajectory updating

- For CSC we get MuEndSegment (which *is a* RecHit) from MuEndChamber, but we use MuEndRecHit (RecHits as well) to update the trajectory.

- complication in the TrajectoryBuilder: we must get the predictedState at each MuEndLayer and update there the state: not as simple as Updator.update();

- Now we have a dedicated class to do that, a class which update the state w/o being an Updator...

Propagator

Common problem in CMS: critical for Muon

- Now Gtf (tracker) and Geane (Muon);
- Geane is precise but slow!!: most of the time spent in the MuonTrajectory Building is spent in propagating state around.
- most propagation is done to find out which is the “right” DetUnit inside a DetLayer: Optimisation needed!!
- e.g. have a fast (and dirty) propagator to find out the right DetUnit, and then use Geane just for that predicetedState.
- a micro workshop on propagators held at CERN some time ago: some idea to extend Gtf also in a varying magnetic field.

Mis-alignment

- Francisco Matorras is starting to use new CommonDet functionality to study Mis-alignment in the MuonSystem: feedback expected soon.

Conclusion and future

- CommonDet heavily used by Muon package.
- Muon need some more work to be fully CommonDet compliant but we are on the way.
- In general good interaction between us.
- Lately some problem in communication!! In ORCA 510 we saw big macroscopic effect in muon tracking (always worst performances, efficiencies, resolution) due to changes in CommonDet we were not aware of (until too late).
- Lot of time spent (wasted) to find out what has changed: solution easy, once found the problem, but prior communication would be better!

- More general comment: Often there is not enough time during the release sequences to carefully test the effects of modification in CommonDet on muon system (Muon is expected to tag $1 \div 2$ days after CommonDet). Sometime we did "blind tag" and then found bug immediately after the release (forcing a new one...)

  It would be nice to have more time to test the new stuff before and outside the release sequence.

- Future requirements:
  - Track persistency (why it does not works??)
  - Optimisation in finding DetUnits
  - Propagator