Muon Reconstruction Integration Status Report

David Quarrie, Niels van Eldik
LBNL, UMass

Brief Reminder of Strategy

- Demonstrate a hybrid reconstruction chain that can accommodate components based on either Moore or Muonboy
  - Road search, Segment finding, track finding, track refinement
- New track finding steering to replace existing ones
  - Able to reproduce multiple tracking finding strategies and to be highly flexible and tunable
- Goal is to have prototype hybrid in place for release 17.0.0
- Tune and validate the hybrid and components
- Selection of baseline components and deployment of hybrid chain
  - Fall 2011 reprocessing
- Re-implementation, validation and deployment of fully unified chain
  - Replacing wrapped FORTRAN components by equivalent C++ ones
  - Beginning of 2013

Work Areas

- Road Search
- Segment Finding
- **Track Finding**
- **Error Handling**
- **Validation Framework**
- Client Integration (Combined Tracking, Trigger, Calibration & Monitoring)

- Weekly meetings
  - 11:00 Mondays (brief and informal)
- Twiki at:
  - https://twiki.cern.ch/twiki/bin/viewauth/Atlas/MuonISR
- Mailing list
  - hn-atlas-muonSWReconstruction@cern.ch

People

- Good response to adding more people to the various areas
- Cast of characters (names in magenta are new to these areas)
  - Error Handling
    - Christof Pahl, Ed Moyse
  - Prototype Steering
    - Zach Marshall, Elisa Pueshel, Christof Pahl, Dinos Bachas, Niels van Eldik, David Quarrie
  - Validation
    - Liming Chen, Valerio Consorty, Tobias Rave, Biagio Di Miccio, Remi Zaidan
    - Ahmimed Ouraou, Jen-Francois Laporte, Niels van Eldik, Ed Moyse
    - Alan Poppleton, Peter Kluit, Rosemarie Aben, Tulin Varol

Road Search

- IMuonRoadSearch API defined
- Data flow during the segment finding different in Moore and Muonboy. Common solution under investigation, work postponed for the moment.
- Road EDM (MuonRoad, MuonRoadCollection and MuonRoadStationIntercept) defined
  - Low priority

Segment Finding

- IMuonSegmentMaker APIs defined
  - implementation exists on both Moore and Muonboy side.
  - initial test of running the Muonboy segment finding in Moore successful.
  - further work needed (validation)
- Implementations at lower priority to Track Finding (next Page)

Track Finding

- Internal EDM defined and implemented
  - Converters to/from public EDM implemented
- Prototype Track Finding Steering
  - Implemented and being extended and tuned
  - Based on list of strategy objects (e.g. BOS->BMS->BIS->All]
- Track Finder API defined
  - Moore-based implementation in place
  - Muonboy-based implementation due this week
- Track Ambiguity Solver API defined
  - One concrete implementation based on Moore
- Track Refinement API defined
  - One concrete implementation based on Moore
- Validation and tuning just beginning

Error Handling

- Requirements in reconstruction
  - different error tunings depending on the reconstruction phase (segment/track finding)
  - segment Errors and Track Errors need to be addressed
- Situation in 16.6.X
  - multiple ROT creators configured on the python side -> configuration nightmare
- Situation in 17.0.X
  - strategies defined and EDM updated to accommodate them
  - ROT creator supports multiple error strategies on the c++ side
  - code backwards compatible -> clients need to migrate to benefit from the new scheme
- Implementation of Segment Errors completed
- Implementation of Track Errors underway

Validation

- Goal is common framework based on Tracking D3PDs.
  - Currently on Inner Detector Hits (PrepRawData) info is in the D3PDs
- Tasks are:
  - Add muon information
    - Initially Track and TrackParticle information, then PRDs, etc.
  - Check that truth is complete and update if necessary
  - Evaluate ID validation and take identical approach if possible/appropriate
  - Identify validation quantities (histograms, summaries)
  - Implementation of comparison framework (RTT, DCube, etc.)
  - Validate against existing tools and deprecate latter
- Extension of Tracking D3PDs almost complete
  - Trying to reproduce existing tests but based on D3PD information

Integration With Clients

- Main clients:
  - trigger
    - full segment and track finding within a given ROI
    - currently call Moore segment and track finding using Moore APIs
  - combined reconstruction
    - segment finding (MuGirl)
    - track cleaning (Muid, MuGirl)
    - hit recovery (Muid, MuGirl)
- Interfaces used by the combined reconstruction will be supported in the future
- The trigger will have to adopt to the new framework BUT migration trivial
  - only few lines of code changes needed
  - can coexist with the old API -> allows for a phased migration

Not Forgetting

- Ongoing support and improvements to both Moore and Muonboy
  - bug fixes
  - CPU reduction
  - preparation for fall reprocessing
    - release 17 validation
    - retuning of the algorithmic code to adopt to new alignment/calibration

Summary

- The “cast of characters” has increased significantly
- Good progress being made
  - Although a bit slower than originally estimated
- Validation and tuning of new track finding steering about to begin
- Road and segment finding put at lower priority, but some components in place
- New validation framework is being put into place
- Initial hybrid chain should be in release 17.0.0
  - Allowing detailed comparisons and selections to be made