

GBL in hps-java

Norman Graf (SLAC)

HPS Software Meeting, 09/18/14

Java GBL porting

- At a previous meeting, reported on first round of commits to svn of:

- Java Matrix and Vector classes to provide the functionality of the three C++ matrix/vector class packages used by GBL

`org.hps.recon.tracking.gbl.matrix`

- Three main GBL classes

- `GblData.java`
- `GblPoint.java`
- `GblTrajectory.java`

`org.hps.recon.tracking.gbl`

- Only translated methods necessary for HPS

Testing/Validating the Java GBL port

- To facilitate testing/debugging of the translated GBL classes, developed a Driver equivalent to the hps-dst HpsGblFitter class
- Read in the same Icio file containing:
 - ❑ MatchedTracks (Data, Track Objects)
 - ❑ GBLTrackData (Data, GenericObject)
 - ❑ GBLStripClusterData (Data, GenericObject)
 - ❑ GBLTrackToStripData (Relations)
 - ❑ TrackToGBLTrack (Relations)
- Can run both C++ and Java code simultaneously and compare outputs

Code Changes

- New constructors to convert from GenericObject
 - GBLStripClusterData
 - GBLTrackData
- For example:

```
List<LCRelation> track2GblTrackRelations =  
    event.get(LCRelation.class, "TrackToGBLTrack");
```

```
// loop over the relations  
for (LCRelation relation : track2GblTrackRelations) {  
    Track t = (Track) relation.getFrom();  
    GBLTrackData gblTrack = new GBLTrackData((GenericObject) relation.getTo());  
    ...  
} //end of loop over tracks
```

Code Changes II

- Lots of changes to GblTrajectory, mostly having to do with the Matrix classes
 - Lots of initialization issues
 - pass-by-value vs pass-by-reference issues
 - other mind-numbingly mundane C++/Java differences
- Some ugly implementations to maintain rough line-by-line parity with the C++ code
 - Java code will be refactored later
 - resisted urge to rewrite the C++ code

Status

- Java package produces same output as C++ hps-dst when compared on few-event samples
- Work ongoing to implement the binary millepede II output.
 - Will then compare high statistics versions of the output files created by the C++ and Java versions.
- Work ongoing to pack results of GBL re-fit into new Track collection.
- Once a validated release is created will go back to refactor and worry about speed performance.
- Driver `org.hps.recon.tracking.gbl.HpsGblRefitter` ready for others to test.