



Olivier Callot

12 June 2012



IN2P3

INSTITUT NATIONAL DE PHYSIQUE NUCLEAIRE
ET DE PHYSIQUE DES PARTICULES



Velo Lite Critique

- Naming
- Removal of dead leaves
- Packaging
- Miscellanea

Naming

◆ The detector name is too long

- And confuses with existing classes
 - This means that some pieces of code use Velo Strip...

◆ I would strongly advocate to use a two letter name

- Like ST, IT, TT, OT, FT
- I would propose VL for Velo Light, and VP for the Pixel option.
- It is important to do it now, or we will keep unnecessary long names for the future!

- But this is just my opinion!

Removal of dead leaves

◆ This code is clearly a cut-and-paste-and-edit of the Velo code

- But is it a good idea to keep all the legacy without a strong clean-up?
- There are clear dead bodies
 - A method of the ChannelID that is commented as obsolete and kept only for backward compatibility!

◆ Many methods with no clear use

- All the various flavours of iterators in DeVeloLite
 - As clean to have `m_vlDet->sensors().begin()` than `m_vlDet->sensorBegin()` !
 - And this means several dozen of method less...
- Who is using `idealPhi`, `halfBoxPhi`, `globalPhi` and similarly offsets?
 - The pattern doesn't use half box nor local for phi...

◆ Several uses of ST base code

- Is it needed? Useful ? It gives constraints that are probably not worth the increase in complexity / decrease in readability due to inheritance.

◆ Handling of routing lines

- The methods are just a place holder now
- Should it be cleaned-up?

◆ Handling of readout order / strip order

- Current phi sensors are readout in a non-natural order by the hardware, re-sorted by the TELL1
- This part of the code is obsolete

Packaging

◆ The location of packages should be made consistent

- The DAQ packages are in LHCb project for some detectors (OT), in Lbcom for other (ST)
 - FT follows OT, VL follows ST...
- How can the user select in which project, when ambiguous like LHCb and Lbcom ?

◆ What should we do for test algorithms

- In the package, in the same src area ?
- Do we need them ?
- Should we have nightly test suites?

Miscellanea

◆ Use of float versus double

- Double are no problem for internal computation
- Any object with double is larger than with float, and uses more cache / memory access
 - Why should we use double to store simple numbers
 - The precision of float is ~ 24 bits, 10^{-7} , this is enough for everything we use in LHCb, except during computation where difference of large numbers can give trouble.
- Use float for all caching array of the geometry ?

◆ Truth computation

- Could be done at the same time as we build clusters
- Just need to add some information in the intermediate MC classes
 - The current implementation with linker between MCClusters and MCHits could be part of the event class MCCluster

◆ Use the method getOrCreate

- Instead of a test if it exist, and a create if not.
 - This is done internally by this method.

◆ The monitoring package in Boole produces plots

- Are they looked at ? Are they useful ?
- Wouldn't it be better to use a counter, with mean / sigma, to monitor 'automatically' that values are OK for this or that production?

◆ What are the classes VeloLiteThreshold and VeloLineInternalCluster for ?

◆ Bug in the DAQ package

- The strip number needs one more bit
- There is no space for it in the current format
- Some information should be removed
 - High Threshold bit?
 - Less precision for fractional part?

- And the decoding should be made available, so that the encoding/decoding could be tested.

Summary

◆ I haven't looked at all details of the code

- This is a copy of the current code, with many, many features...
- I would have liked to have a cleaned version.

◆ There is a (small) problem with the raw bank

- Need one more bit, what should be suppressed ?

◆ This code should work

- As correct as the current Velo digitisation

◆ But the main test will be track reconstruction

- Efficiency and speed critical for HLT1 speed and thus farm size