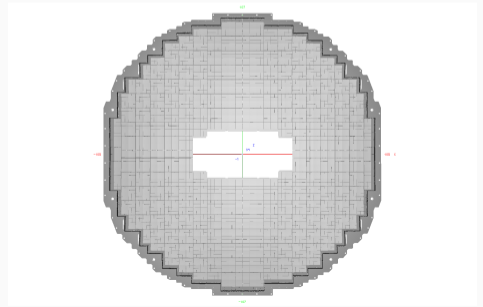


Progress on the Emc-code restructuring and on the new Forward Endcap geometry

Ben William Salisbury

November 5, 2019

HISKP University Bonn



The 1. issue: A single processing chain

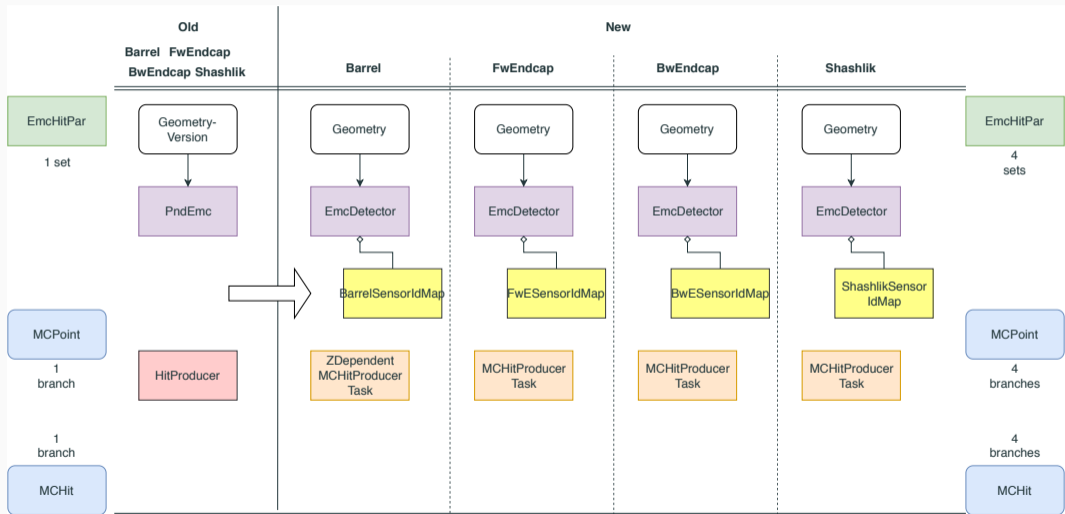
Treating the Emc as one detector, even though there are four, sometimes differently behaving, subdetectors.

Altering behavior is mostly implemented via if-statements into a baseclass.

The effects:

- Larger, hard to maintain classes
- Hard to include new special functionality for a subdetector

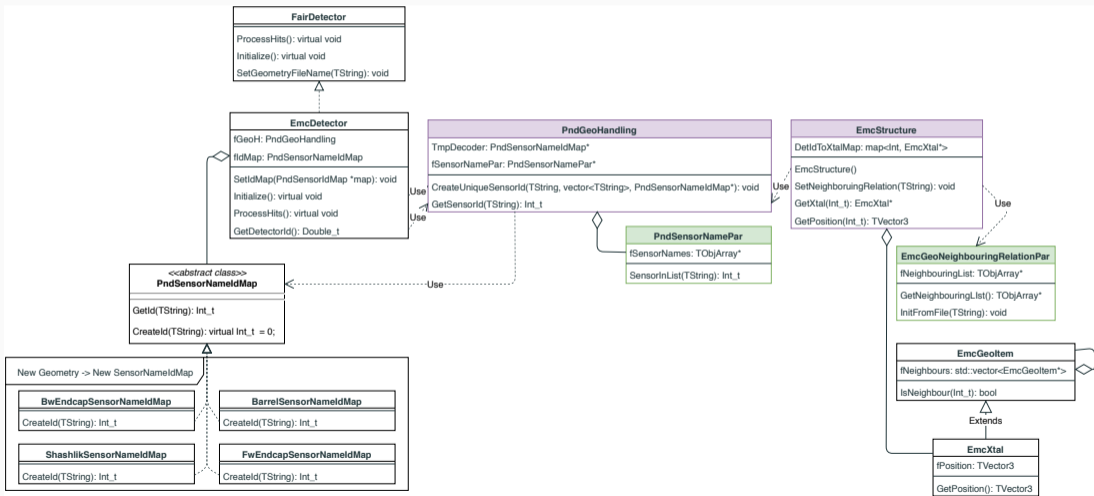
1. Restructuring: Our proposal - Separate the subdetectors



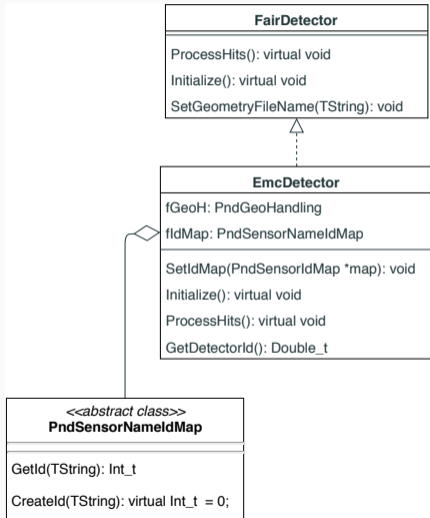
1. Restructuring: Our proposal

- Splitting up the processing while deriving sensibly from common-baseclasses solves our issues
- Hardest part was restructuring the PndEmc as it was overburdened with responsibility:
 1. Creation of MCPoints (actual main responsibility)
 2. Handled multiple geometries (solved by splitting)
 3. Decoding of detector id (cause of code bloat)

Separation of the geometry handling

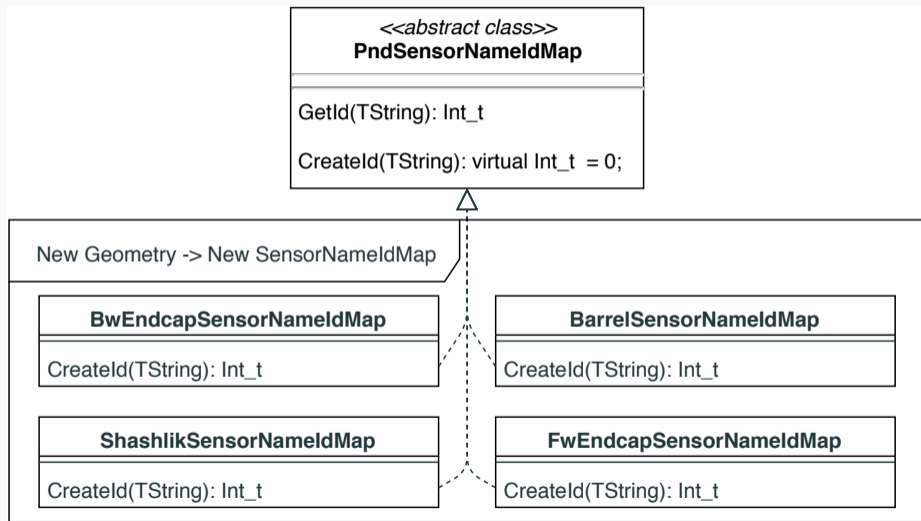


Separation of the geometry handling



1. MCPoint creation is equal for all subdetectors, so it can be implemented in **EmcDetector**
2. Multiple geometry handling gone, as subdetectors-processing split and each **EmcDetector** instance handles just one geometry
3. Detector decoding is specific not just to subdetector, but geometry → add decoding through composition with specific **SensorNameIdMap**

Separation of the geometry handling

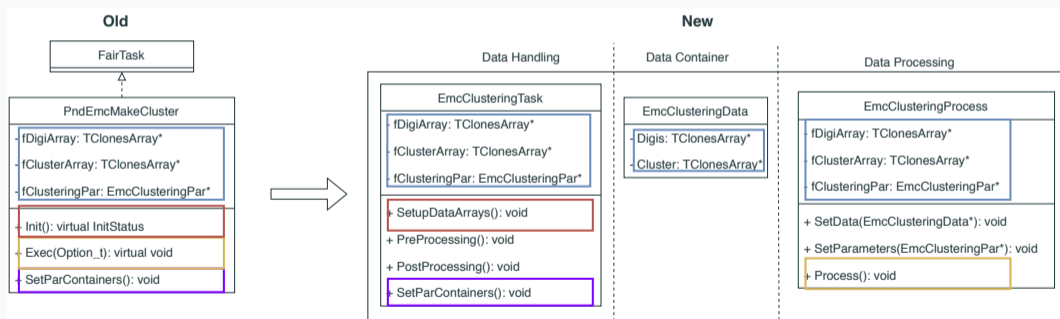


1. Restructuring: Results

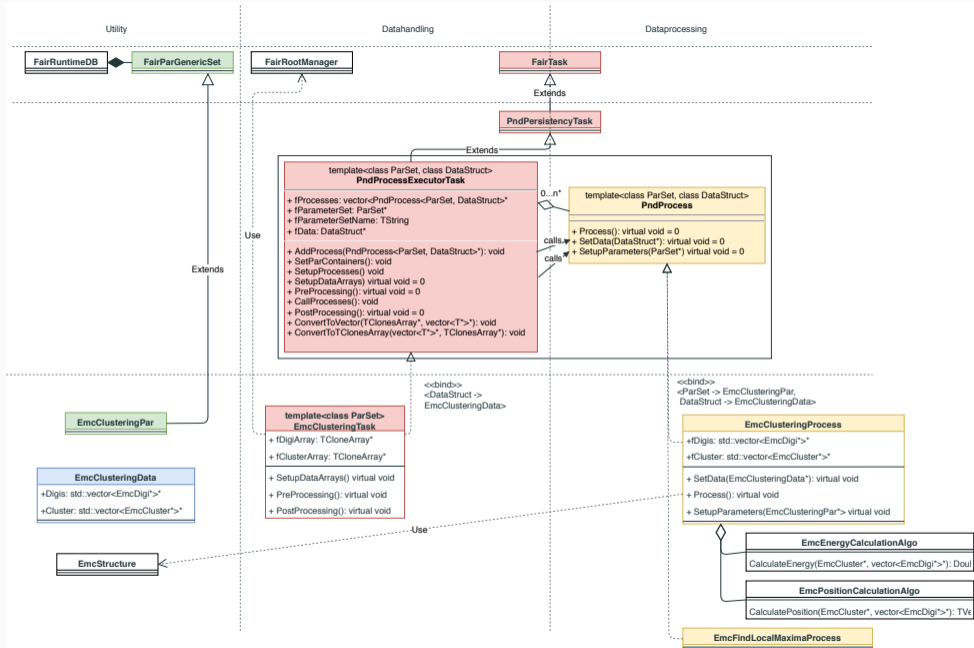
- Smaller classes through usage of inheritance and separation of concerns
- No touching of running code required to include new geometry

The 2. issue: Mixing data-handling code and algorithms

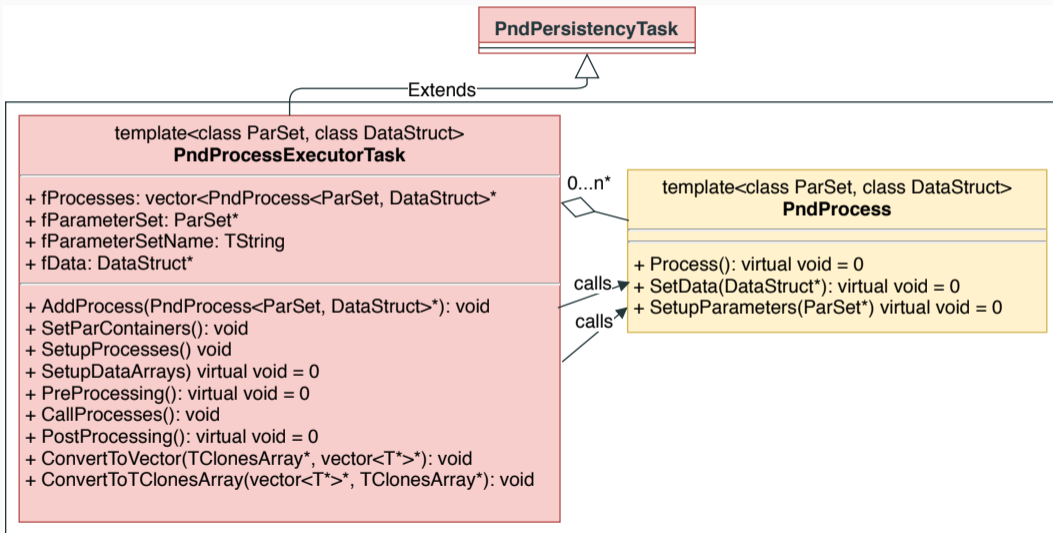
We want to separate the actual processing code/algorithm from the FairROOT's way of handling data.



Else we can not easily reuse our algorithms.



2. Restructuring: Split Task into Process-and-Task



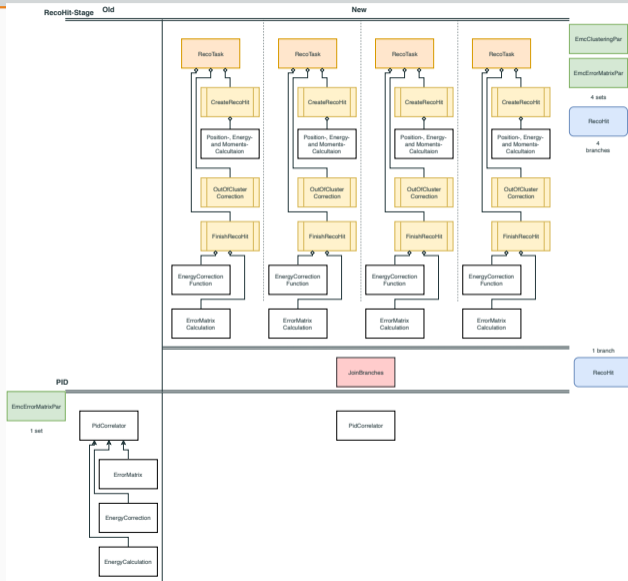
2. Restructuring: Results

- Algorithm/Process separate from Task/Data-handling
- Reduced code repetition through reuse of Tasks
- First step taken to reuse an algorithm while using a different way of data-handling

Emc related processing on Bumps/Cluster in PID

- If we change Emc-reconstruction we have to adjust the PID
- Interesting effects on intermediate Emc-data class such as a Cluster or Bump as they have to contain PID relevant data

3. Restructuring: Adding an additional “RecoHit”-layer

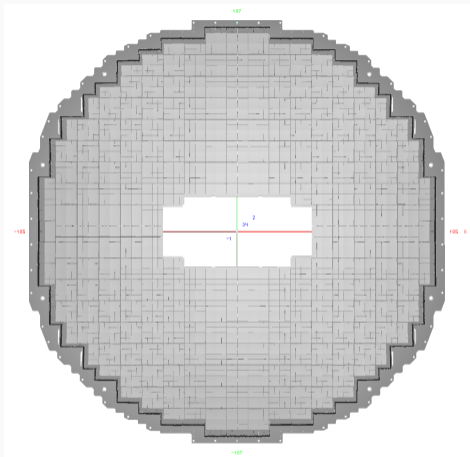


- Include an additional Emc-final-“layer”: the RecoHit
- The RecoHit links to the Bump/SubCluster and contains all of data-members that are required by the PID
- Here we do all final computations: Energy- and Position-Corrections, Error-Calculations
- The PID should just work with the RecoHit

3. Restructuring: Results

- Emc-processing cleanly separated from the later processing-stage (PID)
 - We can change intermediate data-classes (e.g. Cluster, Bumps) and clean them up
 - ...

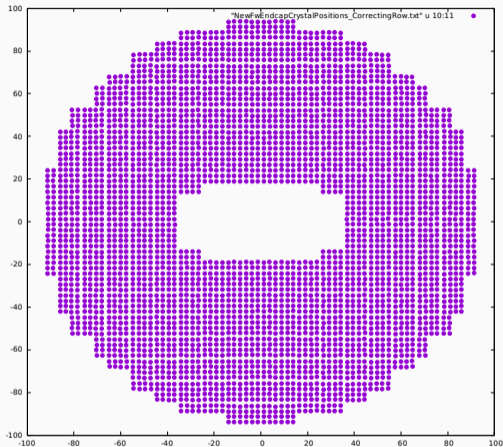
My current Forward Endcap Geometry



Removed the carbon fiber alveole and the insulation for the conversion.

- Since last time I adjusted and changed the CAD drawing
- Now the conversion with T.Stockmann's CADToROOTConverter possible
- Geometry includes passive materials such as the Backplate, Inserts, Mountingplates and Interfaces, as well as the sensitive crystals

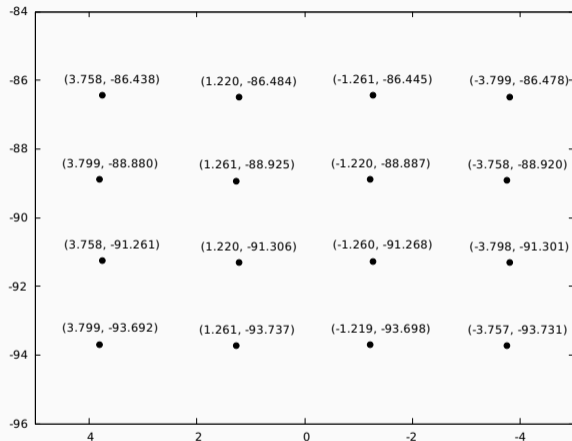
My current Forward Endcap Geometry



Positions of the front face center of each crystal.

- Decoding of the new geometry is working nicely. We can get the same unique id for the same crystal → no new neighbouring list required

My current Forward Endcap Geometry



Positions of the front face center of the X0Y-9 crystals.

- Decoding of the new geometry is working nicely. We can get the same unique id for the same crystal → no new neighbouring list required
- There is some form of systematic noticeable when looking at an alveole closely → so back to CAD

- Each subdetector gets its own processing chain
- Tasks are split up into Task-Process
- Emc-processing cleanly separated from PID
- Reduced CAD-Model now easily convertible, including more material should be easy