February 14th, 2023

Detector and MDI meeting

Small workflow improvements for the next software release

Geometry + LCIO

N. Bartosik [a, b]

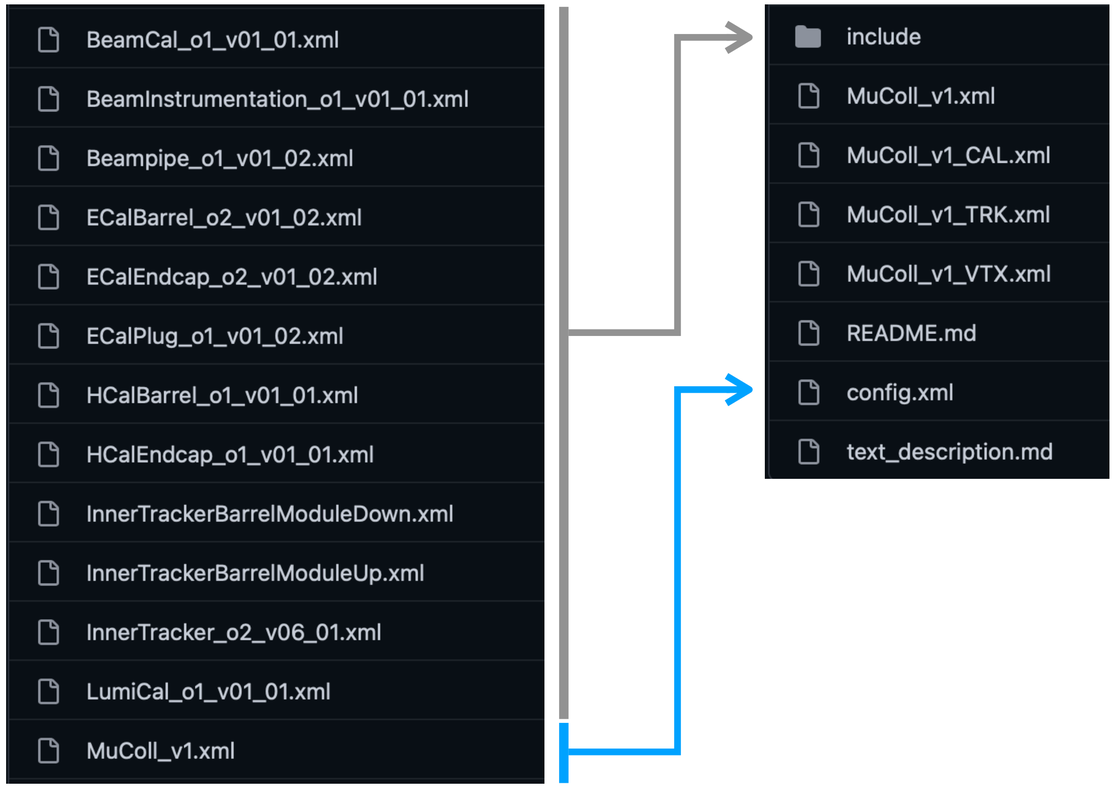
for the **Muon Collider Physics and Detector Group**

[a] INFN Torino *(Italy)* [b] CERN *(Switzerland)*

Cleaned-up geometry: MuColl_v1.0.1

Cleaned-up and restructured geometry prepared: **MuColl_v1.0.1** PR #4

- removed unused subdetector definitions and constants
- resolved 68 overlaps between senstitive surfaces and support structures
- optimised file structure with all common definitions moved to a single file: `config.xml`




```xml
<lccdd>

    <info name="MuColl_v1.0.1"
        title="Muon Collider geometry for sqrt(s) = 1.5 TeV"
        author="Nazar Bartosik"
        url="https://confluence.infn.it/display/muoncollider/Muon+Collider+Home"
        status="development"
        version="$Id$">
        <comment>Cleaned-up 'v1' version with a few bug fixes</comment>
    </info>

    <define>
        <!-- World dimensions -->
        <constant name="world_side"          value="30000*mm"/>
        <constant name="world_x"             value="world_side"/>
        <constant name="world_y"             value="world_side"/>
        <constant name="world_z"             value="world_side"/>
```

Light geometry versions

Default geometry file is very concise now [35 lines of code] ▶
just including other XML files

Several lighter versions of the geometry prepared for convenience
only including everything up to a certain subdetector

- `MuColl_v1_VTX.xml` → Beampipe + Nozzles + Vertex detector
- `MuColl_v1_TRK.xml` → … + Inner Tracker + Outer Tracker
- `MuColl_v1_CAL.xml` → … + ECAL + HCAL
- **`MuColl_v1.xml`** → … + Muon chambers [default]

	FULL	CAL	TRK	VTX
Load time	30 s	30 s	28 s	1 s
RAM	1.2 GB	1.2 GB	800 MB	280 MB

```xml
<lccdd>

    <!-- Loading constants and geometry configuration -->
    <includes>
        <file ref="include/elements.xml"/>
        <file ref="include/materials.xml"/>
        <file ref="config.xml"/>
    </includes>

    <!-- Including subdetector definitions -->
    <include ref="include/Beampipe_o1_v01_02.xml"/>
    <include ref="include/Nozzle_10deg_v0.xml"/>

    <include ref="include/Vertex_o2_v06_01.xml"/>

    <include ref="include/InnerTracker_o2_v06_01.xml"/>
    <include ref="include/OuterTracker_o2_v06_01.xml"/>

    <include ref="include/ECalBarrel_o2_v01_02.xml"/>
    <include ref="include/ECalEndcap_o2_v01_02.xml"/>

    <include ref="include/HCalBarrel_o1_v01_01.xml"/>
    <include ref="include/HCalEndcap_o1_v01_01.xml"/>

    <include ref="include/Solenoid_o1_v01_01.xml"/>

    <include ref="include/YokeBarrel_o1_v01_01.xml"/>
    <include ref="include/YokeEndcap_o1_v01_01.xml"/>

    <!-- Including plugins -->
    <plugins>
        <include ref="include/plugins.xml"/>
    </plugins>

</lccdd>
```

No need to comment lines in the default geometry. Just use a lighter version when appropriate

Geoemtry versioning

I've decided to use the following naming convention for the Muon Collider geometry versions

- `MuColl_v1` global version of the geometry, relevant for the outside world
  e.g. geometry designed for $\sqrt{s}$ = 1.5 TeV, frozen for Snowmass studies
- `MuColl_v1.0.1` same global version, with very minor changes: code imrovements, small fixes
  newer versions can be gradually adopted by users
  v1.0.1 inside our group; v1 for the outside world
- `MuColl_v1.1.1` same as v1.0.1, but with some systematic change in the geometry
  e.g. increased thickness of passive material in VXD to evaluate potential impact of dedicated cooling on occupancy due to increased # of secondary BIB particles

... after many iterations it could be something like this:

- `MuColl_v1.11.5` **the latest $\sqrt{s}$ = 1.5 geometry with added ECAL shielding, optimised magnetic field, changed double-layer spacing in VXD, etc.**
  ↳ **defined as the current baseline geometry for our studies**
  presented as MuColl_v1 for the outer world

Utility improvement: anajob

`anajob` utility is routinely used for checking contents of LCIO files → written in C++

```
>  anajob mumi_t25ns_n150MeV_j1.slcio
```

ANNOYING

```
anajob:  will open and read from files:

     mumi_t25ns_n150MeV_j1.slcio     [ number of runs: 1, number of events: 374 ]

  Run : 0 - MuColl_v1:
 parameter CommandLine [string]: /muoncdata/users/bartosik/clic/rel/ILCSoftware/v02-01/v02-01-pre/DD4hep/v01-11/bin/ddsim --physics.list
QGSP_BERT --steeringFile clic_steer_v0.py --skipNEvents 0 --numberOfEvents 374 --inputFile bib_mcp/mumi_t25ns_n150MeV.slcio --outputFile
bib_simhit/mumi_t25ns_n150MeV_MuColl_v1.slcio_j1.slcio,
 parameter DD4HEPVersion [string]: v01-11,
 parameter DateUTC [string]: 2022-06-17 10:13:21.727360 UTC,
 parameter GEANT4Version [string]: $Name: geant4-10-06-patch-01 $,
 parameter ILCSoft_location [string]: /muoncdata/users/bartosik/clic/rel/ILCSoftware/v02-01/v02-01-pre,
 parameter LCIOFileName [string]: mumi_t25ns_n150MeV_j1.slcio,parameter CommandLine [string]: /muoncdata/users/bartosik/clic/rel/
ILCSoftware/v02-01/v02-01-pre/DD4hep/v01-11/bin/ddsim --physics.list QGSP_BERT --steeringFile clic_steer_v0.py --skipNEvents 0 --
numberOfEvents 374 --inputFile bib_mcp/mumi_t25ns_n150MeV.slcio --outputFile bib_simhit/mumi_t25ns_n150MeV_j1.slcio,
 parameter DD4HEPVersion [string]: v01-11,
 parameter DateUTC [string]: 2022-06-17 10:13:21.727360 UTC,
 parameter GEANT4Version [string]: $Name: geant4-10-06-patch-01 $,
 parameter ILCSoft_location [string]: /muoncdata/users/bartosik/clic/rel/ILCSoftware/v02-01/v02-01-pre,
 parameter LCIOFileName [string]: mumi_t25ns_n150MeV_MuColl_v1_j1.slcio,
 parameter SteeringFileContent [string]: import os\n\nfrom DDSim.DD4hepSimulation import DD4hepSimulation\nfrom g4units import mm, GeV,
MeV, m, deg\nSIM = DD4hepSimulation()\n\n## The compact XML file\n#SIM.compactFile = "/muoncdata/users
. . . . . .
few hundred lines of metadata

--------------------------------------------------------------------------
COLLECTION NAME                 COLLECTION TYPE          NUMBER OF ELEMENTS
==========================================================================
ECalBarrelCollection            SimCalorimeterHit              3209
ECalEndcapCollection            SimCalorimeterHit               507
HCalBarrelCollection            SimCalorimeterHit              2728
HCalEndcapCollection            SimCalorimeterHit              3860
HCalRingCollection              SimCalorimeterHit               520
InnerTrackerBarrelCollection    SimTrackerHit                   817
```

USEFUL

Utility improvement: anajob

Replaced C++ implementation by one in Python → flexible treatment of command-line parameters

```
> anajob -h
```

```
usage: anajob [-h] [-e NUMBER] [-r NUMBER] [-m N] [-v] FILE [FILE ...]

Prints summary of the LCIO file contents

positional arguments:
  FILE                  Input file(s)

optional arguments:
  -h, --help            show this help message and exit
  -e NUMBER, --event NUMBER
                        Event number to print
  -r NUMBER, --run NUMBER
                        Run number to print
  -m N, --max_events N  Maximum number of events to print
  -v, --verbose         Print run headers
```

No run headers printed by default → much less cluttering of the terminal window

You can print just the 1st event → no need to Ctrl + C for stopping the event loop

↳ `anajob -m 1 FILE.slcio`

Tracker digitization: to do

Currently we have 3 versions of tracker digitization code

DDPlanarDigiProcessor

- simple SimHit smearing
  1 particle → 1 hit
- integrated in ILCSoft

MuonCVXDDigitiser master

- pixelated clustering
  1 particle → 1 cluster
- not integrated in ILCSoft
  ↳ seems to be functional

MuonCVXDDigitiser exp.

- pixelated clustering
  ≥1 particles → 1 cluster
- not integrated in ILCSoft
  ↳ needs additional work

Tracker digitization: realistic timing

Currently we have 3 versions of tracker digitization code

DDPlanarDigiProcessor

- simple SimHit smearing
  1 particle → 1 hit
- integrated in ILCSoft

MuonCVXDDigitiser master

- pixelated clustering
  1 particle → 1 cluster
- not integrated in ILCSoft
  ↳ seems to be functional

MuonCVXDDigitiser exp.

- pixelated clustering
  ≥1 particles → 1 cluster
- not integrated in ILCSoft
  ↳ needs additional work

Cluster shapes are a powerful handle on BIB → should integrate MuonCVXDDigitiser to ILCSoft

master **branch is technically usable in our chain:** converting cluster → LCIO::TrackerHitPlane

The only missing feature → proper treatment of timing (we don't know exact ROC functionality)

↳ **we have to be generic and take into account what we know for sure**

- fired pixel becomes blind for a few ns >> σ_t (~10 ps) → configurable dead time
- TDC available at much smaller granularity → configurable TDC matrix size: dX, dY
  ↳ loop over all the pixel blocks → assign t of the earliest pixel to all the dX × dY pixels of the block

Tracker digitization: realistic clustering

Currently we have 3 versions of tracker digitization code

DDPlanarDigiProcessor

- simple SimHit smearing
  1 particle → 1 hit
- integrated in ILCSoft

MuonCVXDDigitiser master

- pixelated clustering
  1 particle → 1 cluster
- not integrated in ILCSoft
  ↳ seems to be functional

MuonCVXDDigitiser exp.

- pixelated clustering
  ≥1 particles → 1 cluster
- not integrated in ILCSoft
  ↳ needs additional work

experimental branch is more tricky to integrate in our simulation chain → worth leaving for later

- more complex cluster shapes in case of overlapping clusters
  ↳ cluster-shape analysis has to be more sophisticated
- more complex retrieval of the truth information
  Track → *RecHit* → *Cluster* → *SimHit* → *MCParticle*
- might require an adjusted data model
  1-to-many relation for RecHit → SimHit